

ANÁLISE DO ALGORITMO CMPN EM REGIME PERMANENTE

Eduardo Santos de Oliveira

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Elétrica , do Centro Federal de Educação Tecnológica Celso Suckow da Fonseca, CEFET/RJ, como parte dos requisitos necessários à obtenção do título de mestre.

Orientador(a): Diego Barreto Haddad

Rio de Janeiro

Junho 2023

ANÁLISE DO ALGORITMO CMPN EM REGIME PERMANENTE

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Elétrica , do Centro Federal de Educação Tecnológica Celso Suckow da Fonseca, CEFET/RJ, como parte dos requisitos necessários à obtenção do título de mestre.

Eduardo Santos de Oliveira

Banca Examinadora:

Documento assinado digitalmente
gov.br
DIEGO BARRETO HADDAD
Data: 17/06/2023 06:49:54-0300
Verifique em https://validar.iti.gov.br

Presidente, Professor D.Sc. Diego Barreto Haddad (CEFET/RJ) (Orientador(a))

Documento assinado digitalmente
gov.br
GABRIEL MATOS ARAUJO
Data: 16/06/2023 13:55:52-0300
Verifique em https://validar.iti.gov.br

Professor D.Sc. Gabriel Matos Araujo (CEFET/RJ)

Documento assinado digitalmente
gov.br
KENJI NOSE FILHO
Data: 16/06/2023 14:14:09-0300
Verifique em https://validar.iti.gov.br

Professor D.Sc. Kenji Nose Filho (UFABC)

Documento assinado digitalmente
gov.br
LEONARDO CORREIA RESENDE
Data: 16/06/2023 16:06:29-0300
Verifique em https://validar.iti.gov.br

Professor D.Sc. Leonardo Correia Resende (IFRJ)

Rio de Janeiro

Junho 2023

Ficha catalográfica elaborada pela Biblioteca Central do CEFET/RJ

O48 Oliveira, Eduardo Santos de
Analise do algoritmo CMPN em regime permanente / Eduardo Santos de Oliveira. — 2023.
77f. : il. color. , enc.

Dissertação (Mestrado) Centro Federal de Educação Tecnológica Celso Suckow da Fonseca, 2023.
Bibliografia : f. 73-77
Orientador: Diego Barreto Haddad

1. Filtros adaptativos. 2. Processamento de sinais. 3. Algoritmos. 4. Engenharia elétrica. I. Haddad, Diego Barreto (Orient.). II. Título.

CDD 621.3822

Elaborada pela bibliotecária Tania Mello – CRB/7 nº 5507/04

DEDICATÓRIA

Este trabalho é dedicado a minha mãe Angela, por ter dado tudo de si para que os filhos tivessem o mínimo de dignidade na vida, e condições de seguirem seus próprios caminhos.

AGRADECIMENTOS

Agradeço ao Prof.Diego Barreto Haddad pela paciência, compreensão e apoio nessa difícil jornada acadêmica.

RESUMO

Análise do Algoritmo CMPN (*Continuous mixed p-norm*) em Regime Permanente

Resumo: O algoritmo de filtragem adaptativa conhecido como CMPN (*Continuous Mixed p-Norm Adaptive*) tem atraído atenção por parte da comunidade científica. Devido à grande não linearidade de sua equação de atualização, inexiste na literatura um modelo estocástico capaz de prever seu desempenho. Nesta dissertação, a não linearidade da sua equação de atualização é relaxada, sendo aproximada por uma relação linear por partes. Tal aproximação é conjugada com a técnica de análise via balanceamento de energia, o que dá origem a uma solução analítica/numérica que prevê com acurácia o desempenho assintótico do algoritmo. Embora o modelo assuma gaussianidade do ruído de medição, ele é capaz de prever o desempenho do algoritmo sob condições de ruído impulsivo, quando o passo de aprendizagem assume valores pequenos. A qualidade das predições teóricas é aquilatada por meio de simulações computacionais.

Palavras-chave: Filtragem Adaptativa; Análise em Regime permanente; Algoritmo CMPN

ABSTRACT

Steady-state Analysis of CMPN (*Continuous mixed p-norm*) Algorithm

Abstract The adaptive filtering algorithm known as CMPN (Continuous Mixed p-Norm Adaptive) has attracted attention from the scientific community. Due to the high nonlinearity of its update equation, there is no stochastic model in the literature capable of predicting its performance. In this dissertation, the nonlinearity of its update equation is relaxed and approximated by a piecewise linear relationship. This approximation is combined with the energy balance analysis technique, which results in an analytical/-numerical solution that accurately predicts the asymptotic performance of the algorithm. Although the model assumes Gaussian measurement noise, it is capable of predicting the algorithm's performance under impulsive noise conditions when the learning step assumes small values. The quality of theoretical predictions is assessed through computational simulations.

Keywords: Adaptive Filtering; Steady-state Analysis; CMPN Algorithm

LISTA DE ILUSTRAÇÕES

Figura 1 – Modelo esquemático de Filtragem adaptativa 20

Figura 2 – Diagrama de bloco de identificação de sistemas com filtro adaptativo 20

Figura 3 – Evolução do MSD ao longo das iterações. 22

Figura 4 – Evolução do MSE ao longo das iterações. 23

Figura 5 – Superfície da função custo do *Steepest Descent*. 24

Figura 6 – Função erro $f[e(k)]$ conforme dado pela Equação (41) (em vermelho). Em verde: aproximação linear por partes conforme Equação (112) . 36

Figura 7 – Evolução do MSD ao longo das iterações dos algoritmos CMPN (em vermelho) e LMS (em azul). Os fatores de aprendizado adotados são $\beta_{\text{LMS}} = 1,7 \times 10^{-2}$ e $\beta_{\text{CMPN}} = 5 \times 10^{-2}$, escolhidos de modo a equalizar o desempenho assintótico. 37

Figura 8 – Evolução do MSD ao longo das iterações. Em vermelho: curva teórica; em azul: curva simulada. 49

Figura 9 – Evolução do MSE ao longo das iterações. Em vermelho: curva teórica; em azul: curva simulada. 49

Figura 10 – Função erro $f[e(k)]$ conforme dado pela Equação (41) (em vermelho). Em verde: aproximação linear por partes conforme Equação (112) . 54

Figura 11 – Comparação do MSE assintótico entre a estimativa teórica (em vermelho) e o resultado de simulações (em azul), referente ao Cenário 1. 62

Figura 12 – Comparação do MSE assintótico entre a estimativa teórica (em vermelho) e o resultado de simulações (em azul), referente ao Cenário 2. 63

Figura 13 – Comparação do MSE assintótico entre a estimativa teórica (em vermelho) e o resultado de simulações (em azul), referente ao Cenário 4. 64
Figura 14 – Comparação do MSE assintótico entre a estimativa teórica (em vermelho) e o resultado de simulações (em azul), referente ao Cenário 5. 67

LISTA DE TABELAS

Tabela 1 – Exemplos de funções do erro adotadas nas equações de atualização de diferentes algoritmos, com α, α_0 e α_1 denotando hiperparâmetros adicionais. 31

Tabela 2 – Valores das constantes adotadas na aproximação linear por partes (112). 59

Tabela 3 – MSE simulado e teórico em regime permanente no Cenário 3. 65

Tabela 4 – MSE (em dB) em regime permanente do algoritmo CMPN num cenário com presença de ruído impulsivo, 69

LISTA DE ABREVIATURAS E SIGLAS

CMPN	Continuous Mixed p-Norm
EMSE	Excess Mean-Square Error
FIR	Finite Impulse Response
GMN	General Mixed-Norm
IIR	Infinite Impulse Response
LAD	Least Absolute Difference
LEAKY-LMS	Leaky-LMS
LMF	Least Mean Fourth
LMMN	Least Mean Mixed Norm
LMS	Least Mean Square
LS	Least Squares
LSE	Least Sum of Exponentials
MSD	Mean Square Deviation
MSE	Mean Squared Error
NLMS	Normalized Least Mean Square
NRMN	Normalized Robust Mixed Norm
RMN	Robust Mixed Norm
SSLMS	Step-Size Saler LMS
ZA-LMS	ZA Least Mean Square

SUMÁRIO

1	**Introdução**	**14**
1.1	Objetivos	15
1.2	Metodologia	16
1.3	Notação Matemática	17
1.4	Publicação	17
1.5	Estrutura deste Trabalho	18
2	**Filtragem Adaptativa**	**19**
2.1	Algoritmo *Steepest Descent*	22
2.2	Filtro de *Wiener*	25
2.3	O algoritmo LMS - *Least Mean Square* - Fundamentos	26
2.3.1	LMS	28
2.3.2	*Excess Mean-Squared Error* - MSE	29
2.3.3	LMS Normalizado	30
2.3.4	Funções custo em filtragem adaptativa	31
2.4	Hipóteses adotadas na filtragem adaptativa	32
2.5	Comentários Finais	32
3	**Algoritmos de Norma Mista**	**33**
3.1	Introdução	33
3.1.1	Algoritmo de Norma Mista Contínua	35
4	**Técnicas de Análise de Algoritmos de Filtragem Adaptativa**	**39**
4.1	Análise de Primeira Ordem do algoritmo LMS	39
4.2	Análise de Segunda Ordem do algoritmo LMS	42
4.3	Fluxo de Energia	44
4.4	Comentários Finais	50

5 Análise de Regime Permanente do Algoritmo Adaptativo de Norma Mista 51
5.1 Análise de Regime Permanente do Algoritmo CMPN 51
5.1.1 Cálculo do termo Termo I da Equação (111) 55
5.1.2 Cálculo do Termo II da Equação (111) 57
5.1.3 Cálculo das integrais 58
5.2 Comentários Finais 59

6 Resultados 60
6.1 Cenários das simulações 60
6.1.1 Cenário 1 61
6.1.2 Cenário 2 62
6.1.3 Cenário 3 64
6.1.4 Cenário 4 66
6.1.5 Cenário 5 67
6.1.6 Cenário 6 68
6.2 Comentários Finais 70

7 Conclusões 71

Referências 72

1- Introdução

Diversas aplicações em processamento digital de sinais apresentam uma quantidade limitada de informações *a priori* acerca dos sinais envolvidos [1]. Neste contexto, técnicas de filtragem adaptativa são não raro vantajosas e flexíveis, possuindo como potenciais aplicações o cancelamento de eco, eliminação de ruído, identificação de sistemas, equalização de canal, controle, predição de sinais, entre outros. Este processamento é realizado através de estruturas (em geral com resposta ao impulso finita), que permitem tanto a extração quanto a manipulação de informações de interesse. Grosso modo, tal processamento mapeia o sinal de entrada para o sinal de saída, extraindo ao final do procedimento a informação de interesse [2].

Quando há informações *a priori* suficientes para o projeto de um filtro digital, tal projeto se torna um problema de otimização que almeja encontrar um filtro que satisfaça as especificações desejadas [3]. Diversas técnicas maduras podem ser mobilizadas para tal fim; por exemplo, um algoritmo iterativo de otimização que utiliza o método *Remez exchange* (também chamado de algoritmo de Parks-McClellan) é amplamente usado para o projetos de filtros com resposta ao impulso finita [3], seja de filtros de comprimento ímpar [4], seja de filtros cujo comprimento é par [5]. Infelizmente, o projeto de um filtro fixo requer especificações muito bem determinadas (tais como uma determinada perda mínima na faixa de rejeição), uma demanda não factível na maioria das aplicações [2]. Neste contexto, o emprego de um algoritmo de filtragem adaptativa é frequentemente uma alternativa factível, já que tal algoritmo busca se adaptar às estatísticas conjuntas dos sinais presentes na aplicação através da atualização de seus parâmetros a cada iteração, em tempo real.

O desempenho de algoritmos de filtragem adaptativa é negativamente influenciado pela presença ubíqua de sinais de ruído, cujas propriedades estatísticas são derivadas dos fenômenos físicos responsáveis por sua geração. Entre tais fenômenos, pode-se destacar o ruído *shot* (possivelmente induzido por tunelamento quântico da corrente [6]) e o ruído térmico [7]. Entre tais tipos de ruído, os de natureza impulsiva são particularmente danosos ao desempenho de diversos algoritmos de filtragem adaptativa clássicos, fundamentados em funções custo quadráticas, o que tem demandado da comunidade científica

o projeto de diversos algoritmos adaptativos *robustos ao ruído impulsivo* [8]. Tais técnica têm se tornado crescentemente importantes em muitas aplicações distintas [9].

Neste contexto, a referência [10] introduziu um novo algoritmo, conhecido como algoritmo de norma mista contínua (CMPN), do inglês *Continuous Mixed p-Norm Adaptive Algorithm*. A supracitada referência já conta com mais de 69 citações[1], o que demonstra sua relevância na comunidade científica. O cômputo da norma ℓ_p mista contínua é modulado pela adoção de uma função densidade de probabilidade ao arbítrio do projetista.

Infelizmente, tal algoritmo apresenta uma equação de atualização demasiado complexa, o que dificulta o desenvolvimento de modelos matemáticos capazes de prever o comportamento do algoritmo. Tais modelos são de grande relevância prática, já que fornecem diretrizes e *insights* para os projetistas destes algoritmos, estabelecendo por vezes relações de causa e efeito interpretáveis, bem como garantias de desempenho (como desempenho assintótico).

Em geral, a qualidade do desempenho dos algoritmos adaptativos é aferida por métricas de desempenho, tais como o erro quadrático médio (MSE), do inglês *Mean Squared Error*, as quais comumente apresentam um valor numérico cada vez menor à medida que o desempenho do algoritmo é melhorado (já que, na maioria das vezes, tais funções aferem alguma medida de desvio com relação a valores ideais). Ao longo das iterações, tais métricas tipicamente vão decrescendo monotonamente (na região conhecida como transiente), até que o desempenho se estabiliza num determinado patamar assintótico (também conhecido como *regime permanente*). Os modelos estocásticos desenvolvidos para os diversos algoritmos adaptativos podem abordar diversas propriedades de seu aprendizado, tal como região de estabilidade, taxa de convergência, evolução do desempenho ao longo do transiente ou o desempenho assintótico.

1.1- Objetivos

Este trabalho apresenta dois objetivos principais:

- Aproximar a função não linear do algoritmo CMPN por uma função que seja linear por partes. Embora tal aproximação, em termos globais, ainda seja não linear, ela

[1] Em março de 2023. Consulta feita na base IEEEXplore.

permite a obtenção da fórmula analítica de interesse.

- A partir da aproximação proposta, obter uma equação analítica de ponto fixo cuja solução coincida com o desempenho assintótico do algoritmo CMPN, a qual servirá de estimativa do desempenho assintótico do algoritmo original CMPN.

Cabe ressaltar que, devido à extrema não linearidade envolvida no algoritmo CMPN, inexistem na literatura análises estocásticas deste algoritmo até o presente momento. Neste sentido, este trabalho é pioneiro na derivação de uma fórmula analítica capaz de prever o desempenho em regime permanente do algoritmo CMPN.

1.2- Metodologia

Devido à natureza estocástica dos algoritmos adaptativos, a cada execução a curva de aprendizagem apresenta uma evolução bastante ruidosa (pouco suave). Tal característica dificulta a predição do desempenho destes, além do fato de em muitas situações relevantes não existirem maneiras de calcular de forma exata a curva de aprendizagem [11]. Na literatura que trata destes modelos, normalmente o foco é a previsão do comportamento *médio* dos algoritmos.

Assim, se uma determinada métrica ϕ apresenta um valor $\phi_i(k)$ na k-ésima iteração da i-ésima execução do algoritmo, o desempenho *assintótico médio* $\overline{\phi}(k)$ pode ser estimado por meio de simulações como [12]

$$\lim_{k\to\infty} \overline{\phi}(k) \approx \frac{1}{J}\sum_{j=1}^{J} \phi_j(K_0), \tag{1}$$

partindo sempre do mesmo ponto inicial e das mesmas condições estatísticas dos sinais de entrada e saída [11], onde K_0 é uma constante suficientemente grande para garantir que o algoritmo tenha alcançado o regime permanente em todas as J execuções, com J tipicamente se encontrando entre 100 e 1000. Assim, a aferição da qualidade da estimativa teórica proposta será efetuada por meio da estimativa simulada (ou empírica) do valor para o qual $\overline{\phi}(k)$ tende assintoticamente. No caso particular do MSE, por exemplo, a Equação (1) assume a seguinte forma

$$\lim_{k\to\infty} \text{MSE}(k) \approx \frac{1}{J}\sum_{j=1}^{J} e_j^2(K_0), \tag{2}$$

onde a escolha específica de K_0 não apresenta grande impacto na estimativa, desde que K_0 seja suficientemente elevado.

1.3- Notação Matemática

Ao longo deste trabalho, $\mathbb{E}[\cdot]$ denota o operador de valor esperado. Matrizes são denotadas por letras maiúsculas em negrito (exemplo, $\boldsymbol{A}$), vetores são denotados por letras minúsculas em negrito (por exemplo, $\boldsymbol{b}$) e escalares por letras minúsculas em itálico (por exemplo, c). Excepcionalmente, algumas constantes escalares arbitráveis são expressas em maiúsculas, sem negrito e em itálico (como o caso de K_0 na Equação (2)). A menos que se explicite o contrário, todas as variáveis assumem valores reais. O operador $()^T$ sinaliza transposição matricial, enquanto que Tr$[\boldsymbol{A}]$ significa o traço da matriz $\boldsymbol{A}$ (isto é, a soma dos elementos que compõem sua diagonal principal). Já $\nabla_{\boldsymbol{c}} h$ denota o vetor gradiente da função escalar h em termos do vetor $\boldsymbol{c}$, enquanto que $\boldsymbol{I}$ significa a matriz identidade com as dimensões compatíveis com o contexto no qual ela se encontra. A expressão det$(\boldsymbol{A})$ denota o determinante da matriz $\boldsymbol{A}$.

1.4- Publicação

Como resultado da pesquisa detalhada nesta dissertação, o seguinte artigo foi publicado:

1. **Oliveira, Eduardo Santos de**; Haddad, Diego Barreto. Análise do Algoritmo CMPN em regime permanente. *CNMAC 2022 XLI Congresso Nacional de Matemática Aplicada e Computacional*, p. 1-4, vol. 1, 2022.

1.5- Estrutura deste Trabalho

Este trabalho encontra-se estruturado do seguinte modo: o Capítulo 2 apresenta uma descrição em nível geral dos algoritmos de filtragem adaptativa. O Capítulo 3 trata da família de algoritmos de norma mista. Enquanto o Capítulo 4 aborda as diferentes técnicas de análise dos algoritmos de filtragem adaptativa, o Capítulo 5 contém a contribuição deste trabalho, referente à análise do algoritmo CMPN. As simulações e resultados são apresentados no Capítulo 6, enquanto que as conclusões e trabalhos futuros localizam-se no Capítulo 7.

2- Filtragem Adaptativa

Filtros adaptativos vêm se tornando uma solução padrão em diversas aplicações, nas últimas décadas, em particular, as que envolvem sinais e/ou sistemas variantes no tempo [13; 14]. Esquematicamente, a configuração geral de um processo de filtragem adaptativa pode ser representada conforme a Figura 1, na qual o filtro adaptativo recebe o sinal de entrada[1] $x(k) \in \mathbb{R}$, no instante de tempo k, que produzirá um sinal de saída, denotado por $y(k) \in \mathbb{R}$.

Considerando um cenário supervisionado, o sinal de entrada $\boldsymbol{x}(k)$ é comparado com o sinal desejado $d(k)$ e a diferença entre eles produz um erro $e(k)$. A partir do erro, o algoritmo adaptativo modifica os valores dos coeficientes do filtro. Diferentes algoritmos adaptativos utilizam regras distintas para tal modificação, as quais são expressas nas suas respectivas equações de atualização. Por conseguinte, a sucessiva atualização dos coeficientes $\boldsymbol{w}(k)$ ocorre através de diversas iterações deste processo [12].

Em um processo de filtragem adaptativa o projetista busca através do algoritmo empregado, além da minimização da função custo, otimizar duas características essenciais que são a taxa de convergência e o custo computacional. Entende-se a taxa de convergência como sendo o número de ciclos, durante o processo adaptativo, para o algoritmo convergir para a solução ótima. Já o custo computacional está relacionado com a quantidade de operações computacionais que o algoritmo precisará executar, além de recursos de *hardware* como processamento, armazenamento, memória. Além destas questões, de acordo com [12], a escolha de um algoritmo é determinada por outras dimensões, tais como robustez, probabilidade de divergência e capacidade de rastreamento.

Os fatores citados podem ser afetados em função da escolha da estrutura do filtro, que pode ser, no caso dos filtros lineares (FIR), do inglês *Finite Impulse Response* ou (IIR), do inglês *Infinite Impulse Response*. Ou seja, para um modelo ou protótipo de um filtro adaptativo eficiente, faz-se necessário balancear todos estes aspectos (robustez, divergência, rastreamento e etc), tendo em vista o resultado que se espera alcançar.

Na situação onde se pretende identificar um sistema (foco desta dissertação),

[1] Neste trabalho, todos os vetores são do tipo coluna.

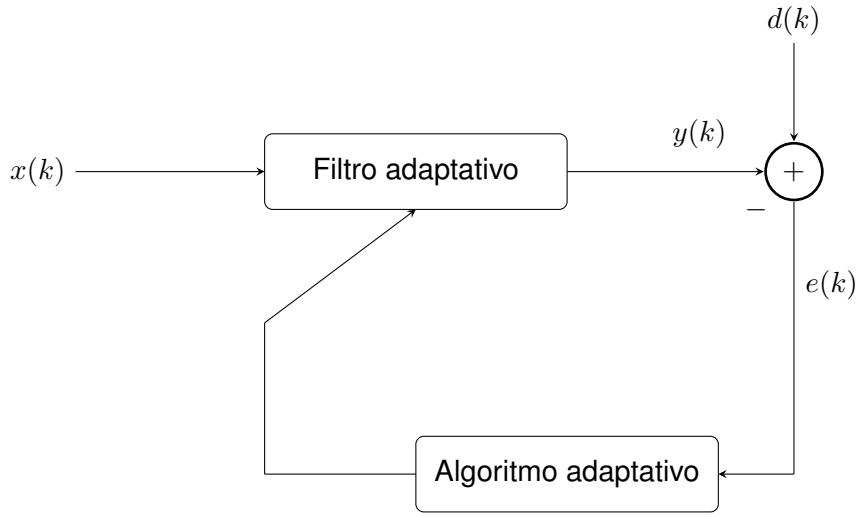


Figura 1 – Modelo esquemático de Filtragem adaptativa

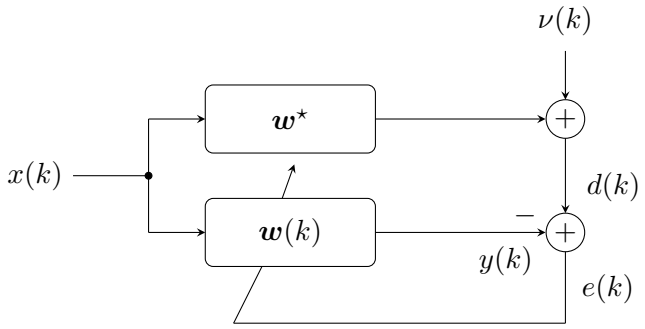


Figura 2 – Diagrama de bloco de identificação de sistemas com filtro adaptativo

através do processo de filtragem adaptativa, temos, de acordo com a Figura 2, um sinal de entrada submetido, simultaneamente ao filtro ótimo (e desconhecido) $\boldsymbol{w}^*$ e ao filtro $\boldsymbol{w}(k)$, o qual pretende-se identificar seus coeficientes.

O sinal filtrado por $\boldsymbol{w}^*$, na presença de ruído $\nu(k)$, gera o sinal desejado $d(k)$, referente ao processo de identificação. No mesmo instante k, o filtro $\boldsymbol{w}(k)$, com os coeficientes identificados até então, tem sua saída representada por $y(k)$, que ao ser confrontado com $d(k)$ produz o erro $e(k)$:

$$e(k) = d(k) - y(k), \tag{3}$$

erro o qual, como vimos, é utilizado, nas diferentes vertentes dos algoritmos de filtragem adaptativa supervisionada, para atualizar os coeficientes de $\boldsymbol{w}(k)$.

Para que um filtro adaptativo tenha êxito em sua tarefa, que é aproximar cada vez mais sua saída ao sinal desejado, cumpre minimizar uma função custo, que é uma função

que relaciona os sinais de saída, erro e de referência. A partir da discussão pregressa, a função custo[2] é dependente do erro $e(k)$.

Assim, o processo de filtragem adaptativa, no qual o algoritmo busca encontrar o conjunto ótimo dos coeficientes do filtro, apresenta conexões estreitas com a teoria de otimização [2]. A escolha entre um determinado algoritmo ou outro pelo projetista deve tentar atender tanto na acurácia ao identificar o sistema em questão, quanto ter um bom desempenho em outros fatores, como o custo computacional, taxa de convergência e o comportamento do algoritmo quando alcançar o regime permanente, que ocorre basicamente a partir do momento em que o erro quadrático médio se estabiliza num patamar suficientemente baixo. Já a taxa de convergência, grosso modo, é a velocidade com que o algoritmo alcança o regime permanente.

Naturalmente há um *trade-off* entre taxa ou velocidade de convergência e desempenho em regime permanente, sendo que a melhora em um aspecto acarreta piora em outro. Geralmente, o parâmetro que controla esses dois aspectos é o passo de aprendizagem do algoritmo (representado na literatura geralmente por μ ou β), que é uma variável ajustável pelo projetista, ou seja, um valor maior de β proporcionará uma aceleração no processo de convergência, porém terá o custo de comprometer o desempenho do algoritmo em regime permanente, já que os coeficientes do filtro estarão sendo ajustados por um parâmetro de maior magnitude. Por outro lado, um valor menor de β tornará a convergência mais lenta, mas com o chamado *ruído de gradiente* com menor impacto sobre a estimativa, apresentando uma variabilidade menor no estimador em regime permanente, com uma consequente melhora nas métricas de desempenho, tais como o MSE ou o (MSD) do inglês *Mean Square Deviation*.

Dependendo da aplicação, pode-se trabalhar com uma taxa de convergência menor porém com um regime permanente com melhor desempenho. Porém em outros casos, a convergência do algorimo pode ser o fator crítico. Tal compromisso pode ser contornado mediante a adoção de esquemas nos quais o fator de aprendizagem varia ao longo das iterações [15].

O desempenho de um algoritmo pode ser aquilatado mediante apoio dos gráficos que relacionam o MSD ou o MSE com o número de iterações. Afim de viabilizar as análises gráficas, tanto o MSD quanto o MSE têm normalmente seus valores convertidos

[2]Esta discussão é restrita aos algoritmos de filtragem adaptativa conhecidos como *supervisionados*.

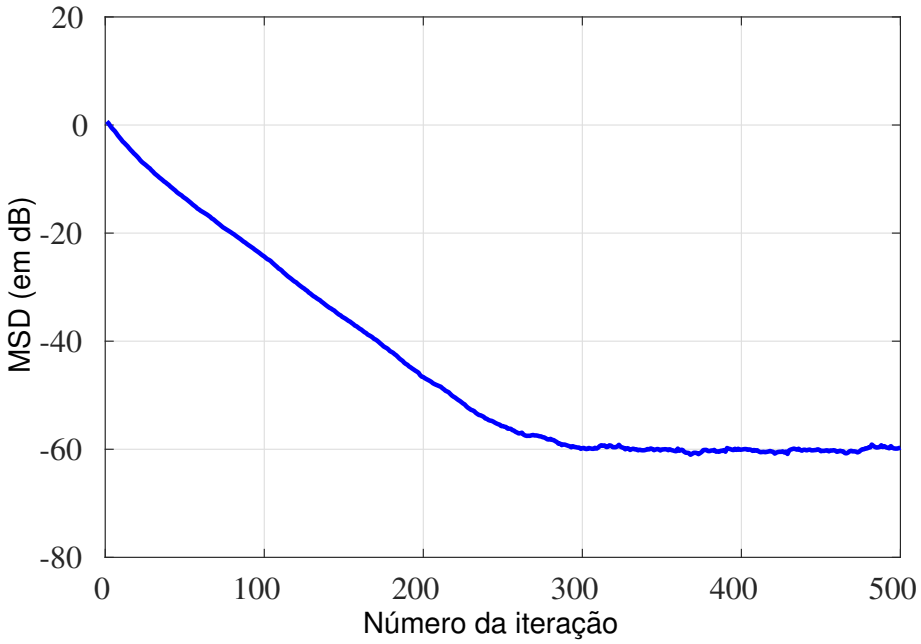


Figura 3 – Evolução do MSD ao longo das iterações.

para decibéis (dB). Como exemplo seja o caso em que o filtro ótimo seja dado por

$$\boldsymbol{w}^{\star} = \begin{bmatrix} 1 & 0,5 & -0,1 \end{bmatrix}, \tag{4}$$

o sinal de entrada é obtido pela filtragem de um sinal branco, gaussiano e de variância unitária pelo filtro $H(z) = 1 - 0,8z^{-1}$. A variância do ruído de medição branco e gaussiano é 10^{-5}. Com um número de ensaios independentes de Monte Carlo igual a 100 e com um passo de aprendizagem $\beta = 5 \times 10^{-2}$, temos a evolução do MSD (em dB) e do MSE (em dB) nas Figuras 3 e 4.

2.1- Algoritmo *Steepest Descent*

O algoritmo *Steepest Descent* é um método utilizado para ajustar os pesos ou coeficientes de um filtro com base no gradiente, porém na direção contrária deste [16]. Consiste em um método de pesquisa de um valor ótimo baseado no gradiente real da função custo, e não numa mera estimativa deste [12]. Levando em conta o contexto da filtragem adaptativa, o *Steepest Descent* trabalha de forma recursiva, à medida que

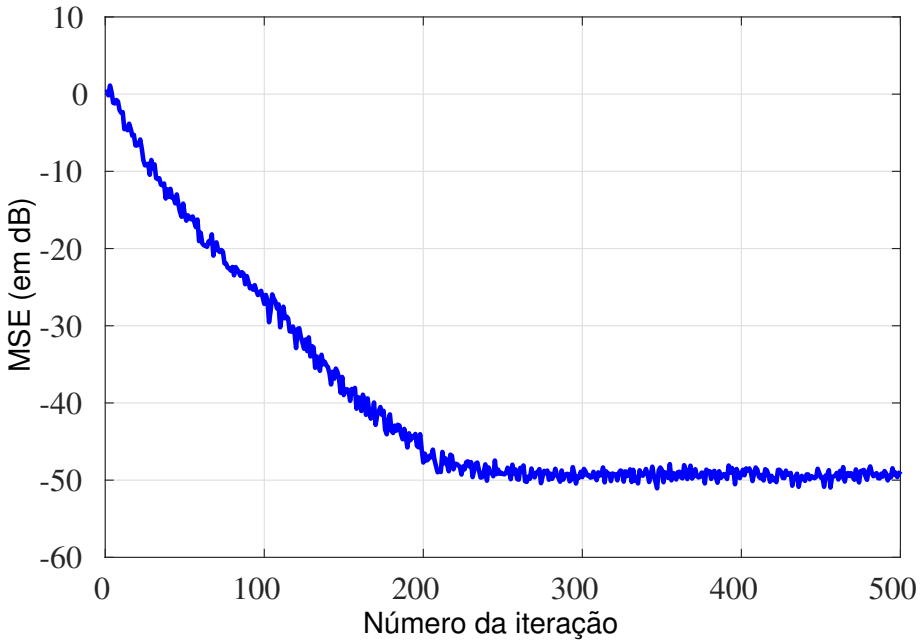


Figura 4 – Evolução do MSE ao longo das iterações.

realiza a atualização do vetor $\boldsymbol{w}$ de pesos na superfície do MSE[3], o que pode ocasionar instabilidade [12].

A ideia por trás do algoritmo *Steepest Descent* é simples. Basicamente, ao minimizar a função custo, representado por uma curva quadrática convexa, busca-se, a partir de um ponto inicial para os coeficientes do filtro, $\boldsymbol{w}$, a ser aproximado, encontrar através de várias iterações do algoritmo o mínimo global da função custo, e consequentemente a convergência para o vetor $\boldsymbol{w}^{\star}$ [17].

O algoritmo *Steepest Descent* atualiza os coeficientes do filtro em sua forma geral dado pela equação:

$$\boldsymbol{w}(k+1) = \boldsymbol{w}(k) - \beta \nabla_{\boldsymbol{w}(k)} \left\{ G[e(k)] \right\}, \tag{5}$$

Onde $\{G[e(k)]\}$ é a função custo, neste caso dada através do MSE.

O fator de convergência β do *Steepest Descent* deve estar dentro do seguinte intervalo [2]

$$0 < \beta < \frac{1}{\lambda_{\text{max}}} \tag{6}$$

onde λ_{max} é o maior autovalor da matriz de autocorrelação $\boldsymbol{R}$. Esta é uma condição

[3]Embora nossa discussão do algoritmo *Steepest Descent* se concentre no MSE, este algoritmo pode ser aplicado em outras funções custo.

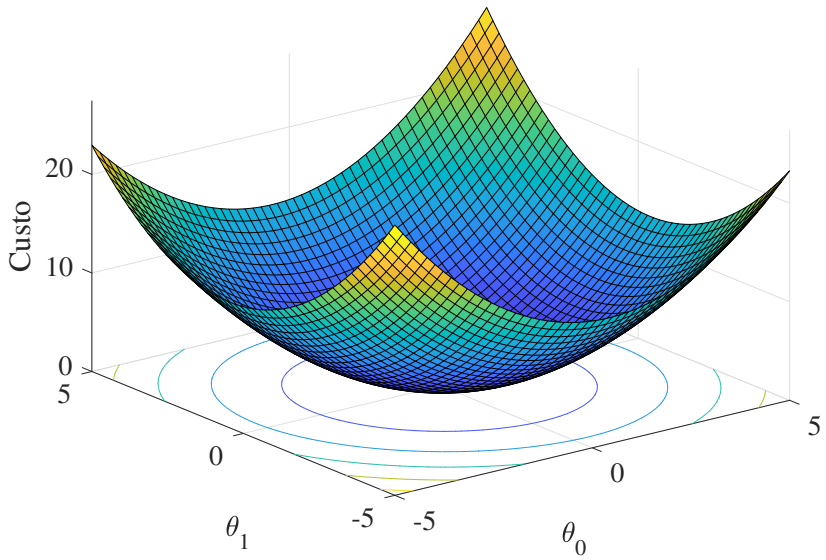


Figura 5 – Superfície da função custo do *Steepest Descent*.

necessária e suficiente para a convergência do algoritmo *Steepest Descent* sobre a superfície quadrática dada pela função custo, ver Figura 5, ou seja, o MSE [16], o que garante também na maioria dos casos a estabilidade do método.

A solucão $\boldsymbol{w}^\star$ do algoritmo *Steepest Descent* possui uma forma fechada, dada pela Equação (14), porém a solução na prática é obtida por um processo iterativo buscando aproximar o vetor de coeficientes do filtro, $\boldsymbol{w}(k)$ ao vetor ótimo $\boldsymbol{w}^\star$ [18].

Podemos assim, descrever o algoritmo *Steepest Descent* como sendo um processo iterativo que buscar minimizar o erro quadrático médio com relação a $\boldsymbol{w}$, sendo $d(k)$ uma variável aleatória com variância σ_d^2 e $\boldsymbol{x}(k)$ também, assumidamente, um vetor aleatório de média zero e variância unitária. A escolha da taxa de aprendizado β deve levar em conta a faixa de valores que garante estabilidade e por fim sua equação de atualização dos coeficientes é dada por:

$$\boldsymbol{w}(k+1) = \boldsymbol{w}(k) + \beta[\boldsymbol{p} - \boldsymbol{R}\boldsymbol{w}(k)], \tag{7}$$

à qual dará origem ao algoritmo LMS, conforme veremos mais adiante, quando $\boldsymbol{p}$ e $\boldsymbol{R}$ serão substituídos por seus equivalentes instantâneos (ou estocásticos) $\boldsymbol{x}(k)d(k)$ e $\boldsymbol{x}(k)\boldsymbol{x}^T(k)$, respectivamente.

2.2- Filtro de *Wiener*

O filtro de *Wiener* é um sistema linear, que tem como objetivo estimar um sinal de saída com base em um sinal desejado, atendendo algumas premissas, como por exemplo que os processos estocásticos envolvidos sejam estacionários e invariantes no tempo [2]. Além disso há duas condições para este tipo de filtro [12], que são:

- O filtro tem que ser linear, para torná-lo tratável matematicamente
- O filtro opera em tempo discreto, para ser possível a simulação em softwares ou DSPs (*Digital Signal Processors*).

Considerando o MSE como a função custo, temos que

$$\mathbb{E}[e^2(k)] = \mathbb{E}[d^2(k) - 2d(k)y(k) + y^2(k)], \tag{8}$$

ou seja:

$$\begin{aligned} \mathbb{E}[e^2(k)] &= \xi(k) && (9)\\ &= \mathbb{E}[d^2(k) - 2d(k)\boldsymbol{w}^T(k)\boldsymbol{w}^T(k)\boldsymbol{x}(k)\boldsymbol{x}^T(k)\boldsymbol{w}(k)] && (10)\\ &= \mathbb{E}[d^2(k)] - 2\mathbb{E}[d(k)\boldsymbol{w}^T(k)\boldsymbol{x}(k)] + \mathbb{E}[\boldsymbol{w}^T(k)\boldsymbol{x}(k)\boldsymbol{x}^T(k)\boldsymbol{w}(k)] && (11) \end{aligned}$$

Podemos observar que na Equação (11) o trecho $\mathbb{E}[d(k)\boldsymbol{x}(k)]$ é o vetor de correlação cruzada entre os sinais $d(k)$ e $x(k)$. Observamos também que o trecho $\mathbb{E} = [\boldsymbol{x}(k)\boldsymbol{x}^T(k)]$ é a matriz de correlação do sinal de entrada $\boldsymbol{x}(k)$. Chamando o vetor de correlação cruzada de $\boldsymbol{p}$, podemos reescrever a Equação (8) da seguinte forma:

$$\begin{aligned} \mathbb{E}\left[e^2(k)\right] &= \mathbb{E}[d^2(k)] - 2\boldsymbol{w}^T\mathbb{E}[d(k)\boldsymbol{x}(k)] + \boldsymbol{w}^T\mathbb{E}[\boldsymbol{x}(k)\boldsymbol{x}^T(k)]\boldsymbol{w}\\ &= \mathbb{E}[d^2(k)] - 2\boldsymbol{w}^T\boldsymbol{p} + \boldsymbol{w}^T\boldsymbol{R}\boldsymbol{w}, && (12) \end{aligned}$$

cujo gradiente com relação a $\boldsymbol{w}$ pode ser descrito por:

$$\nabla_{\boldsymbol{w}}\mathbb{E}[e^2(k)] = -2\boldsymbol{p} + 2\boldsymbol{R}\boldsymbol{w}. \tag{13}$$

Ao igualar-se o gradiente $\nabla_{\boldsymbol{w}}\mathbb{E}[e^2(k)]$ ao vetor nulo, encontramos a solução ótima da equação, a qual representa os coeficientes que minimizam a função custo, ou seja:

$$\boldsymbol{w}^{\star} = \boldsymbol{R}^{-1}\boldsymbol{p}, \tag{14}$$

a qual é a solução de *Wiener*.

Embora a solução de *Wiener* tenha uma fundamentação teórica interessante, sua implementação na prática tem limitações, visto que exige conhecimento prévio das estatísticas de segunda ordem conjunta dos sinais envolvidos, que na prática não estão costumeiramente disponíveis, além da suposição de que os sinais são conjuntamente estacionários, suposição frequentemente violada em situações práticas. Assim, mesmo não havendo aplicação prática para o filtro de *Wiener* no contexto da filtragem adaptativa, ele tem sua importância como aferição de desempenho dos diferentes algoritmos, como por exemplo o (LMS).

Sendo estimadores, os filtros adaptativos não raro apresentam algumas características que facilitam seu estudo analítico. Uma propriedade muito comum é conhecida como ortogonalidade, que simplifica, como veremos adiante, a derivação do algoritmo *Steepest Descent*, uma versão não estocástica do LMS que utiliza o gradiente do MSE. Basicamente, o princípio diz que o valor ótimo de um filtro *Wiener* e a estimativa do erro do estimador ótimo são ortogonais [17].

Assim, quando os coeficientes de um filtro estão nos seus valores ótimos $\boldsymbol{w}^{\star}$ isto é, coincidem com o filtro de *Wiener*, configurados para os seus valores ótimos, temos que as derivadas parciais da função custo quadrática em relação aos coeficientes do filtro são iguais a zero, o que demonstra que o erro é descorrelacionado do sinal de entrada do filtro [17]; isto é:

$$\mathbb{E}[e(k)\boldsymbol{x}(k-i)] = 0, \text{ para } i = 0, 1, ..., N-1. \tag{15}$$

2.3- O algoritmo LMS - *Least Mean Square* - Fundamentos

O algoritmo LMS é ainda o algoritmo mais utilizado na área de filtragem adaptativa devido à sua simplicidade e baixo custo computacional. Deste algoritmo derivam-se

vários outros com características específicas e voltadas para a solução de diversas aplicações, como por exemplo os algoritmos (ZA-LMS) [19] e (Leaky-LMS) [20], voltados para sistemas esparsos. Além disso, de acordo com [2], um outro fator que também torna atrativo o uso do LMS é a convergência em ambientes estacionários, onde convergir significa o algoritmo caminhar na média em direção à solução ótima, ou seja, aquela solução que fornece os melhores coeficientes do filtro em relação à função custo adotada.

O LMS teve sua origem no trabalho de Widrow e Hoff [16] em 1960, e desde então tem sido a base para derivação de tantos outros algoritmos. Ademais, existem ainda diversos artigos recentes que tentam elucidar alguns aspectos de seu sofisticado processo de aprendizagem [21; 22].

O desempenho do algoritmo é muito dependente da matriz de autocorrelação do sinal de entrada $\boldsymbol{x}(k)$, o qual, assumindo uma estrutura transversal, coleta N amostras consecutivas do sinal de entrada $x(k)$:

$$\boldsymbol{x}(k) \triangleq \begin{bmatrix} x(k) & x(k-1) & x(k-2) & \dots & x(k-N+1) \end{bmatrix}^T \tag{16}$$

A matriz de autocorrelação do sinal de entrada $\boldsymbol{R}$ é definida por:

$$\boldsymbol{R}(k) \triangleq \mathbb{E}\left[\boldsymbol{x}(k)\boldsymbol{x}^T(k)\right], \tag{17}$$

ou, mais explicitamente:

$$\boldsymbol{R} = \begin{bmatrix} \mathbb{E}[x^2(k)] & \mathbb{E}[x(k)x(k-1)] & \dots & \mathbb{E}[x(k)x(k-N+1)] \\ \mathbb{E}[x(k-1)x(k)] & \mathbb{E}[x^2(k-1)] & \dots & \mathbb{E}[x(k-1)x(k-N+1)] \\ \vdots & \ddots & \ddots & \vdots \\ \mathbb{E}[x(k-N+1)x(k)] & \dots & \dots & \mathbb{E}\left[x^2(k-N+1)\right] \end{bmatrix}. \tag{18}$$

A derivação clássica do LMS faz uso do gradiente, o vetor que indica a variação de uma função escalar f em um determinado instante de tempo k, ou seja, a derivada direcional da função. Mais especificamente, o LMS utiliza uma técnica de otimização conhecida como *gradiente estocástico*, na qual a função custo $G[e(k)]$, é avaliada apenas na amostra mais recente do sinal de erro para efetuar a atualização dos coeficientes, de acordo com a seguinte regra

$$\boldsymbol{w}(k+1) = \boldsymbol{w}(k) - \beta \nabla_{\boldsymbol{w}(k)} \left\{G[e(k)]\right\}, \tag{19}$$

onde $\beta \in \mathbb{R}_+$ é o passo de aprendizagem ou fator de aprendizagem (*step size*, em inglês), arbitrável pelo projetista do algoritmo e responsável por conciliar requisitos por vezes conflitantes, tais como a taxa de convergência e o desempenho assintótico (ou desempenho em regime permanente). Cabe observar que o sinal negativo em (19) deriva do fato de a otimização almejar uma minimização, a qual é empreendida seguindo-se a direção contrária à do vetor gradiente.

No caso do LMS, a escolha particular de $G[e(k)]$ é

$$G_{\text{LMS}}[e(k)] \triangleq \frac{1}{2}e^2(k), \tag{20}$$

de modo que a inserção de (20) em (19) dá origem à equação de atualização do LMS:

$$\begin{aligned} \boldsymbol{w}(k+1) &= \boldsymbol{w}(k) - \beta\nabla_{\boldsymbol{w}(k)}\{G[e(k)]\} \\ \boldsymbol{w}(k+1) &= \boldsymbol{w}(k) - \beta\nabla_{\boldsymbol{w}(k)}\left\{\frac{1}{2}e^2(k)\right\} \\ \boldsymbol{w}(k+1) &= \boldsymbol{w}(k) + \beta\boldsymbol{x}(k)e(k). \end{aligned} \tag{21}$$

Em ambientes cujos sinais sejam conjuntamente estacionários, pode-se provar que a solução ótima para o qual o algoritmo LMS deve convergir é a descrita pelo *filtro de Wiener*.

2.3.1 LMS

O arcabouço teórico descrito até o momento serviu como motivação e base para o desenvolvimento do algoritmo LMS, cujo surgimento eliminou a necessidade dos métodos anteriores de possuirem conhecimento das estatísticas dos sinais, para obter alguma solução. Na verdade, este é um dos objetivos dos métodos estocásticos em geral, ou seja, evitar a necessidade de ter conhecimento das estatísticas dos sinais, como covariância e correlação cruzada, que em casos reais podem não se encontrar disponíveis. Além disso, o LMS permitiu desenvolver aplicações práticas reais que aqueles métodos não permitiram e impulsionar o crescimento da área de filtragem adaptativa com o surgimento de inúmeros outros métodos.

Os métodos estocásticos, como o LMS, obtém estas estimativas através de um mecanismo recursivo de aprendizagem [18]. Uma propriedade emergente dos métodos estocásticos é o fato de apresentarem um mecanismo de rastreio (*tracking mechanism*), que permite acompanhar as variações das estatísticas dos sinais, bem como dos sistemas envolvidos no ambiente de aprendizado.

De acordo com [17], o algoritmo LMS é um esquema prático para realização de um filtro de *Wiener*, sem precisar resolver sua equação, sendo uma implementação estocástica do *Steepest Descent*, que substitui a função custo $\xi = \mathbb{E}[e^2(k)]$ por uma estimativa instantanea $\hat{\xi}(k) = e^2(k)$.

A derivação do algoritmo LMS pode ser obtida por duas abordagens distintas, a partir do *Steepest Descent* [17] como citado no parágrafo anterior ou por meio do método dos Multiplicadores de *Lagrange* [12].

Seguindo a primeira abordagem, ao substituir a função custo por sua estimativa instantânea obtém-se:

$$\boldsymbol{w}(k+1) = \boldsymbol{w}(k) - \beta \nabla_{\boldsymbol{w}(k)} \frac{e^2(k)}{2} \tag{22}$$

Sabendo-se, pela regra da cadeia, que

$$\nabla_{\boldsymbol{w}(k)} \frac{e^2(k)}{2} = e(k) \nabla_{\boldsymbol{w}(k)} e(k) \tag{23}$$

e efetuando a substituição $e(k) = d(k) - y(k)$ na Equação (23), chega-se à equação de atualização do LMS:

$$\boldsymbol{w}(k+1) = \boldsymbol{w}(k) + \beta e(k) \boldsymbol{x}(k). \tag{24}$$

2.3.2 *Excess Mean-Squared Error* - MSE

Os coeficientes do filtro adaptativo regido pelo algoritmo LMS convergem *na média* para os valores ótimos, quando determinadas restrições são satisfeitas. Devido à estocasticidade do estimador adaptativo, os coeficientes adaptativos apresentam oscilações. Tais variabilidade são capturadas por estatísticas de segunda ordem, das quais dependem

métricas tais como o MSD e o MSE. Tais oscilações podem ser atribuídas ao chamado ruído de gradiente [2]. O (EMSE) pode ser definido como sendo a diferença entre o erro quadrático médio $J(k)$ produzido pelo LMS e o mínimo erro quadrático médio J_{min} produzido pelo filtro de *Wiener* [12]. Matematicamente o EMSE do algoritmo LMS é formulado como [2]:

$$\xi_{ex} = \lim_{k\to\infty} \Delta\xi(k) \simeq \frac{\beta\sigma_n^2 \text{Tr}[\boldsymbol{R}]}{1 - \beta\text{Tr}[\boldsymbol{R}]}. \tag{25}$$

2.3.3 LMS Normalizado

Quando o tamanho do filtro adaptativo é muito grande, o algoritmo LMS pode requerer um passo de aprendizagem muito pequeno para garantir estabilidade. De acordo com [12], numa situação dessa, o LMS sofre de um problema chamado de "amplificação de ruído do gradiente". Uma forma de contornar este problema consiste na adoção de estratégias normalizadas, das quais o algoritmo (NLMS) (*Normalized Least Mean Square*) é o mais conhecido. Desta forma a equação do algoritmo LMS se torna:

$$\boldsymbol{w}(k+1) = \frac{\beta}{\delta + \boldsymbol{x}^T(k)\boldsymbol{x}(k)} e(k)\boldsymbol{x}(k), \tag{26}$$

onde o termo $\delta \in \mathbb{R}_+$ é um valor pequeno, introduzido para evitar eventuais divisões por zero. O algoritmo NLMS tem a grande vantagem de ser estável quando $0 < \beta \leq 1$, fazendo com que o limite de estabilidade em relação a β independa do tamanho do filtro ou de estatísticas do sinal de entrada, as quais são, por vezes, difíceis de serem estimadas.

2.3.4 Funções custo em filtragem adaptativa

Na literatura de filtragem adaptativa, tanto em artigos ou em livros, é comum encontrarmos a função custo em seu modelo tradicional, que depende do erro obtido do processo de filtragem a cada iteração do algoritmo, por exemplo

$$e(k) = d(k) - \boldsymbol{w}^T(k)\boldsymbol{x}(k) = [\boldsymbol{w}^\star]^T \boldsymbol{x}(k) - \boldsymbol{w}^T(k)\boldsymbol{x}(k) + \nu(k) \tag{27}$$

No entanto, existem diversos outros modelos de função custo, mais adequadas a outras situações em que o modelo tradicional não atende de forma satisfatória.

Seguindo este pensamento, pode-se encontrar na literatura e também em [23], algumas funções não lineares comuns, conforme Tabela 1. Percebe-se que o aumento da complexidade matemática da função erro aumenta imediatamente o desafio de analisar determinado algoritmo, além de implicar outros problemas como, por exemplo, o aumento do custo computacional. Por outro lado, tais funções não lineares podem possuir propriedades desejáveis, como aumento da robustez a ruído impulsivo e redução da probabilidade de divergência. Em termos mais gerais, a equação de atualização de diversos algoritmos pode então ser expressa como:

$$\boldsymbol{w}(k+1) = \boldsymbol{w}(k) + \beta \boldsymbol{x}(k) f[e(k)]. \tag{28}$$

Algoritmo	$f[e(k)]$
LMS	$e(k)$
LMF	$e^3(k)$
LMF Family	e^{2k+1}
LMMN	$\alpha_0 e(k) + \alpha_1 e^3(k)$
Sign Error	sign$[e(k)]$
LMLS	$\alpha \frac{e^3(k)}{1+\alpha e^2(k)}$
LLAD	$\frac{\alpha e(k)}{1+\alpha \lvert e(k) \rvert}$
LHCAF	tanh$(\alpha e(k))$

Tabela 1 – Exemplos de funções do erro adotadas nas equações de atualização de diferentes algoritmos, com α, α_0 e α_1 denotando hiperparâmetros adicionais.

2.4- Hipóteses adotadas na filtragem adaptativa

Na literatura é comum encontrarmos algumas premissas adotadas na aplicação de algoritmos de filtragem adaptativa, que objetivam tornar a modelagem tratável matematicamente. Estas premissas residem tanto no sinal de entrada, no sinal de erro e também na relação entre eles [21].

- *Independence Assumption* (IA) ou premissa de independência: Praticamente ubíqua no campo das chamadas aproximações estocásticas, esta premissa pressupõe a independência estatística entre os coeficientes do filtro e o sinal de entrada e, de forma alternativa, a independência entre sinais de entrada consecutivos. Tal hipótese é costumeiramente violada na prática quando estruturas transversais são adotadas, já que estas introduzem um acoplamento determinístico entre vetores de entrada consecutivos. Ela pode ser contornada mediante recurso à chamada análise de esperança exata [21], a qual implica um alto custo computacional e não provê relações de causa e efeito interpretáveis para o comportamento do algoritmo.
- *Noise Assumption* (NA) ou premissa do ruído: A hipótese NA assume que o ruído possui média igual a zero e é i.i.d. (independente e identicamente distribuído) e estatisticamente independente do sinal de entrada. Cabe ressaltar que NA é uma hipótese fisicamente plausível, pois é aplicável por exemplo quando há predominantemente ruído térmico no sistema [24].

Doravante também consideraremos que o filtro adota uma estrutura transversal e que o tamanho da função de transferência é igual ao tamanho N do filtro.

2.5- Comentários Finais

Neste capítulo, uma visão concisa de uma importante família de algoritmos de filtragem adaptativa foi apresentada, de modo a sedimentar alguns conceitos básicos, os quais serão mobilizados no próximo capítulo para detalhar os algoritmos de norma mista, que são o foco do modelo estocástico proposto.

3- Algoritmos de Norma Mista

3.1- Introdução

Como vimos no Capítulo 2, o algoritmo LMS pode ser derivado por meio da técnica do gradiente estocástico (vide Eq. (19)). Em geral, a função custo adotada nos diversos algoritmos de filtragem adaptativa apresenta uma relação quadrática com o erro instantâneo, sendo muitas das vezes escolhido o erro quadrático instantâneo, conforme a Eq. (21).

Em contextos de processamento de sinais não gaussianos (entre os quais os sinais de ruído de natureza impulsiva são um caso particular), funções custo quadráticas não costumam ser as mais adequadas [8]. Embora o algoritmo Sign-LMS possa ser adotado neste contexto, ele apresenta como desvantagem uma baixa taxa de convergência [25]. Assim, há uma busca por algoritmos que apresentem robustez à interferência, sem perder taxa de convergência, tampouco desempenho assintótico [25].

Um dos primeiros trabalhos neste sentido [26] combinou as normas de segunda e quarta potências, respectivamente adotadas pelos algoritmos LMS e (LMF) (*Least Mean Fourth*), pretendendo obter vantagens quando o valor absoluto do erro é menor que 1 (caso em que o algoritmo LMF é mais robusto, devido à quarta potência de sua norma) e quando o valor absoluto do erro é maior que 1, contexto no qual o LMF pode se tornar instável. Em [26] esta combinação resultou no algoritmo (LMMN) *(Least Mean Mixed-norm)*, o qual busca minimizar a função custo abaixo:

$$G_{\text{LMMN}}[e(k)] \triangleq \overline{\lambda} e^2(k) + (1-\overline{\lambda}) e^4(k), \tag{29}$$

onde $\overline{\lambda} \in [0,1]$ é um parâmetro de mistura arbitrável. Quando $\overline{\lambda} = 1$, o algoritmo se degenera no LMS e quando $\overline{\lambda} = 0$, o algoritmo equivale ao LMF.

Em [27], um novo algoritmo é proposto, o (RMN) (*Robust Mixed-Norm*), que combina o sign-LMS com o princípio (LAD) (*Least Absolute Difference*), com a vantagem de ser robusto em cenários de identificação de sistemas em que o sinal desejado seja

corrompido por ruído impulsivo. A função custo e atualização do algoritmo RMN são dados por:

$$G_{\mathsf{RMN}}[e(k)] \triangleq \overline{\lambda} e^2(k) + (1-\overline{\lambda})(k)|e(k)|, \tag{30}$$

o qual também possui sua versão normalizada, (NRMN) (*Normalized Robust Mixed-Norm*) que se propõe a aumentar a taxa de convergência e reduzir o erro em regime permanente, se comparado aos algoritmos LMS, NLMS, LAD e RMN [27]. Uma generalização de (30) adota a função custo com parâmetros adicionais p e q [28], dando origem ao algoritmo (GMN) (*General Mixed-Norm*):

$$G_{\mathsf{GMN}}[e(k)] \triangleq \overline{\lambda}|e(k)|^p + (1-\overline{\lambda})(k)|e(k)|^q. \tag{31}$$

Tanto o algoritmo RMN quanto o GMN utilizam combinações de potências do sinal de erro, exigindo muitas vezes a especificação de parâmetros adicionais (tal como $\overline{\lambda}$). O algoritmo (LSE) (*Least Sum of Exponentials*) considera uma função custo baseada numa soma de exponenciais [29]:

$$G_{\mathsf{LSE}}[e(k)] \triangleq \frac{1}{2}\left\{\mathsf{exp}[e(k)] + \mathsf{exp}[-e(k)]\right\} = \mathsf{cosh}[e(k)], \tag{32}$$

a qual é uma combinação linear de infinitas potências *pares* do sinal de erro $e(k)$, cujos pesos prescindem de ajuste individual, já que

$$G_{\mathsf{LSE}}[e(k)] = 1 + \frac{1}{2!}e^2(k) + \frac{1}{4!}e^4(k) + \frac{1}{6!}e^6(k) + \ldots = \sum_{n=0}^{\infty}\frac{1}{(2n)!}e^{2n}(k). \tag{33}$$

Uma classe de algoritmos bastante adequada para estes propósitos apelam a funções não lineares para limitar eventuais flutuações transitórias na estimativa do erro [30]. Tais funções costumam apresentar uma propriedade em comum, a de retorno decrescente com o aumento da magnitude do sinal de erro [29]. Um destes exemplos é o algoritmo (SSLMS) (*Step-size Scaler* LMS), o qual propõe a função custo [31]:

$$G_{\mathsf{SSLMS}}[e(k)] \triangleq \frac{1}{\overline{\lambda}}\frac{1-\mathsf{exp}\left[-\overline{\lambda}(e(k)/\|\boldsymbol{x}(k)\|^2)\right]}{1+\mathsf{exp}\left[-\overline{\lambda}e(k)/\|\boldsymbol{x}(k)\|^2\right]}, \tag{34}$$

a qual é dependente do parâmetro ajustável $\overline{\lambda} \in \mathbb{R}_+$.

3.1.1 Algoritmo de Norma Mista Contínua

Como vimos, a seleção de uma função custo apropriada (a qual frequentemente depende do sinal de erro) é um elemento crucial na proposição de novos algoritmos [32]. No contexto da presença de ruído impulsivo, muitas das abordagens se relacionam, direta ou indiretamente, com a otimização de diferentes normas (tais como ℓ_1 e ℓ_2) na função custo [33], sendo a equação (29) um exemplo ilustrativo desta característica.

Neste contexto, a referência [10] propõe o algoritmo adaptativo de norma mista contínua, ou *Continuous Mixed p-Norm Adaptive Algorithm* (CMPN), que é o foco deste trabalho.

O algoritmo CMPN efetua uma mistura entre as normas, fato que justifica o termo *mixed-norm* no nome do algoritmo. A proposta deste algoritmo, segundo seu autor [10], é ser robusto contra ruído impulsivo, generalizando a ideia do RMN, que minimiza a norma do erro com base na Equação (30).

O algoritmo CMPN propõe como função custo

$$G_{\text{CMPN}}(k) = \int_1^2 \lambda_k(p)\,|e(k)|^p\,dp, \tag{35}$$

com $\lambda_k(p)$ denotando uma função densidade de probabilidade, herdando todas as propriedades de uma função densidade de probabilidade válida, tal como $\int_{-\infty}^{\infty} \lambda_k(p)dp = 1$.

Utilizando a técnica do gradiente estocástico (vista no Capítulo 2), temos:

$$\boldsymbol{w}(k+1) = \boldsymbol{w}(k) - \beta\nabla_{\boldsymbol{w}(k)}G(k) \tag{36}$$

temos que

$$\frac{\partial G(k)}{\partial \boldsymbol{w}_i(k)} = \int_1^2 \lambda_k(p)\frac{\partial}{\partial \boldsymbol{w}_i(k)}\,|e(k)|^p\,dp. \tag{37}$$

Assumindo, por falta de informação, que a função densidade de probabilidade $\lambda_k(p)$ seja uniforme no intervalo [1,2], a Equação (37), obtém-se a equação de atualização do algoritmo CMPN:

$$\boldsymbol{w}(k+1) = \boldsymbol{w}(k) + \beta\gamma_k \text{sign}[e(k)]\boldsymbol{x}(k), \tag{38}$$

sendo γ_k determinado pela equação:

$$\gamma_k = \frac{(2\,|e(k)| - 1)\ln(|e(k)|) - |e(k)| + 1}{\ln^2(|e(k)|)}. \tag{39}$$

Assim, o algoritmo CMPN pode ser considerado uma versão do algoritmo sign-LMS, com um passo de aprendizagem variável.

Para propósitos da análise a ser efetuada no Capítulo 5, a Equação (38) pode ser mais conveniente descrita por

$$\boldsymbol{w}(k+1) = \boldsymbol{w}(k) + \beta \boldsymbol{x}(k) f[e(k)], \tag{40}$$

com a função $f[e(k)]$ sendo dada por:

$$f[e(k)] = \frac{[(2\,|e(k)| - 1)\ln(|e(k)|) - |e(k)| + 1]\text{sign}[e(k)]}{\ln^2(|e(k)|)}, \tag{41}$$

a qual apresenta uma característica não linear bastante pronunciada, conforme pode ser visto na Figura 6.

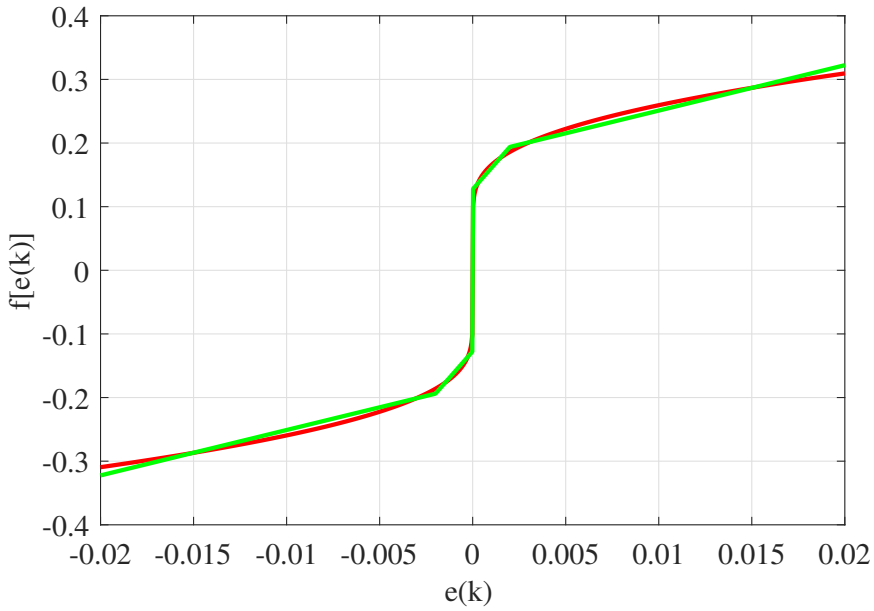


Figura 6 – Função erro $f[e(k)]$ conforme dado pela Equação (41) (em vermelho). Em verde: aproximação linear por partes conforme Equação (112) .

Não é o objetivo desta dissertação demonstrar em condições genéricas a superioridade do algoritmo CMPN perante algoritmos tradicionais de filtragem adaptativa, já que tal superioridade já foi alvo de análise na literatura pregressa. Para efeitos didáticos, será

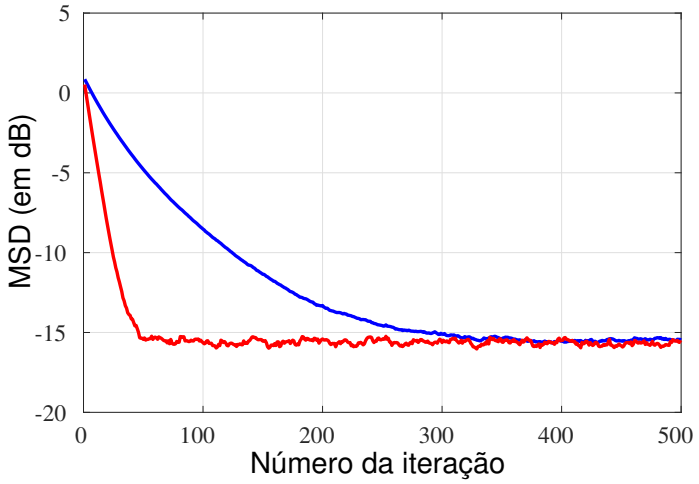


Figura 7 – Evolução do MSD ao longo das iterações dos algoritmos CMPN (em vermelho) e LMS (em azul). Os fatores de aprendizado adotados são $\beta_{\text{LMS}} = 1,7 \times 10^{-2}$ e $\beta_{\text{CMPN}} = 5 \times 10^{-2}$, escolhidos de modo a equalizar o desempenho assintótico.

apresentada uma simulação num cenário submetido a ruído impulsivo, para tornar mais concreta para o leitor tal vantagem.

Seja um cenário cujo filtro ótimo tenha seus coeficientes dados por:

$$\boldsymbol{w}^{\star} = \begin{bmatrix} 1 & 0,3 & 0,25 \end{bmatrix}^{T}, \tag{42}$$

com um sinal de entrada obtido pela filtragem de um sinal branco, gaussiano, de média zero e variância unitária pelo filtro $H(z) = 1 - 0,8z^{-1}$. Seja um ruído de medição $\nu(k)$ impulsivo descrito por

$$\nu(k) = \begin{cases} \mathcal{N}(0, \sigma_{\nu}^{2}), & \text{com probabilidade } 1 - P_{\text{imp}} \\ \mathcal{N}(0, \overline{\sigma}_{\nu}^{2}), & \text{com probabilidade } P_{\text{imp}} \end{cases}, \tag{43}$$

com $\overline{\sigma}_{\nu}^{2} \gg \sigma_{\nu}^{2}$ e $\mathcal{N}(\mu, \sigma^{2})$ sendo uma variável aleatória gaussiana de média $\mu \in \mathbb{R}$ e variância $\sigma^{2} \in \mathbb{R}_{+}$. A probabilidade P_{imp} corresponde à probabilidade de haver uma amostra de ruído impulsivo, a qual é modelada por uma variável aleatória gaussiana com alta variância. As curvas do MSD são apresentadas na Figura 7, obtida com 20.000 ensaios independentes de Monte Carlo[1], $\sigma_{\nu}^{2} = 10^{-6}$, $\sigma_{\nu}^{2} = 10$, $P_{\text{imp}} = 0,01$. Observa-se claramente que a taxa de convergência do algoritmo CMPN supera a do LMS no cenário considerado.

[1] O ruído impulsivo introduz grande variabilidade nas métricas de desempenho. Por esta razão, nesta simulação foi necessário adotar um número excepcionalmente elevado de ensaios de Monte Carlo.

O algoritmo CMPN efetua de modo elegante uma combinação entre normas do sinal do erro. Algumas extensões dele têm sido propostas na literatura. A referência [34] conjuga a técnica CMPN com a norma ℓ_0, almejando otimizar a identificação de sistemas esparsos. A função densidade de probabilidade $\lambda_k(p)$ (originalmente uniforme) em (37) foi generalizada para uma distribuição linear em [35]. O algoritmo foi generalizado para aplicações de controle ativo de ruído em [36]. Uma função custo correlata, denominada *soft-root-sign*, é apresentada em [37]. A inserção de um fator ajustável de mistura atualizado pela normal ℓ_1 do erro, para uma função custo similar, é proposta em [38]. Uma adaptação para um algoritmo de norma afim é descrita em [39].

A complexidade da não linearidade computada pelo algoritmo CMPN é demasiado elevada, o que impossibilitou, até o presente momento, o desenvolvimento de modelos teóricos capazes de prever o desempenho do algoritmo. O objetivo deste trabalho reside em encontrar, pela primeira vez na literatura, uma equação cuja solução descreva o desempenho assintótico do algoritmo CMPN. Antes de descrevermos a solução proposta nesta dissertação, cumpre-nos apresentar alguns tópicos fundamentais relativos ao desenvolvimento de modelos estocásticos de algoritmos de filtragem adaptativa. Este é o tema do próximo capítulo.

4- Técnicas de Análise de Algoritmos de Filtragem Adaptativa

A adoção prática dos algoritmos de filtragem adaptativa depende intensamente da existência de garantias teóricas, as quais são em geral fornecidas por modelos estocásticos. Tais modelos podem prever a região na qual o algoritmo é estável (geralmente em termos do passo de aprendizagem β), a taxa de convergência, a probabilidade de divergência e o desempenho assintótico (em termos, por exemplo, da variância do ruído de medição e do passo de aprendizagem), que formam assim as garantias necessárias para a aplicação dos algoritmos.

Neste capítulo, alguns conceitos e metodologias popularmente empregados para o desenvolvimento de modelos estocásticos relativos a algoritmos de filtragem adaptativa são apresentados[1]. Existem inúmeras técnicas de análise na literatura; iremos aqui tratar de algumas das mais importantes, de modo a trazer ao leitor uma perspectiva importante acerca desta área. Uma dessas análises (que apela a princípios de conservação de energia) nos será de especial interesse, já que será adaptada, no próximo capítulo, para a análise do algoritmo CMPN.

4.1- Análise de Primeira Ordem do algoritmo LMS

Denotando a solução ótima de *Wiener* por $\boldsymbol{w}^\star$ (vide Eq. (14)), descrevemos a diferença entre esta e a estimativa $\boldsymbol{w}(k)$ como um vetor desvio, definido por

$$\tilde{\boldsymbol{w}}(k) \triangleq \boldsymbol{w}^\star - \boldsymbol{w}(k), \tag{44}$$

de modo que uma recursão para $\tilde{\boldsymbol{w}}(k)$ pode ser obtida, a partir da equação de atualização do LMS (21), obtendo-se:

$$\tilde{\boldsymbol{w}}(k+1) = \tilde{\boldsymbol{w}}(k) - \beta e(k)\boldsymbol{x}(k). \tag{45}$$

[1] Este capítulo é fortemente baseado em [2].

Assumindo que o sinal de referência, num contexto de identificação de sistemas lineares, respeita o seguinte modelo

$$d(k) = [\boldsymbol{w}^{\star}]^{T} \boldsymbol{x}(k) + \nu(k), \tag{46}$$

temos que (45) pode ser reescrita como

$$\tilde{\boldsymbol{w}}(k+1) = \tilde{\boldsymbol{w}}(k) - \beta \boldsymbol{x}(k)[\boldsymbol{x}^{T}(k)\boldsymbol{w}^{\star} + \overbrace{\nu(k)}^{\triangleq e_0(k)} - \boldsymbol{x}^{T}(k)\boldsymbol{w}(k)], \tag{47}$$

onde $e_0(k) \in \mathbb{R}$ denota o erro ótimo (obtido quando o filtro assume o valor ótimo) descrito por:

$$e_0(k) \triangleq d(k) - (\boldsymbol{w}^{\star})^{T} \boldsymbol{x}(k) = \nu(k) \tag{48}$$

Utilizando-se (44), a Equação (47) pode ser reescrita, dando origem a

$$\tilde{\boldsymbol{w}}(k+1) = \left[\boldsymbol{I} - \beta \boldsymbol{x}(k)\boldsymbol{x}^{T}(k)\right] \tilde{\boldsymbol{w}}(k) - \beta e_0(k)\boldsymbol{x}(k). \tag{49}$$

A aplicação do operador de valor esperado em (49) leva a

$$\mathbb{E}\left[\tilde{\boldsymbol{w}}(k+1)\right] = \mathbb{E}\left\{\left[\boldsymbol{I} - \beta \boldsymbol{x}(k)\boldsymbol{x}^{T}(k)\right] \tilde{\boldsymbol{w}}(k)\right\} - \beta \mathbb{E}\left[e_0(k)\boldsymbol{x}(k)\right], \tag{50}$$

uma recursão cujo cômputo analítico é de difícil solução, já que exige o cálculo de momentos conjuntos das variáveis aleatórias envolvidas. Para se contornar este obstáculo matemático, normalmente se adotam hipóteses estatísticas, cuja escolha deve levar em conta a necessidade de se obter resultados fechados com a demanda de não se perder acurácia nas predições.

Uma das hipóteses mais utilizadas na análise de algoritmos de filtragem adaptativa é conhecida como *hipótese da independência*[2], a qual assume que o vetor de entrada $\boldsymbol{x}(k)$ é independente da estimativa adaptativa $\boldsymbol{w}(k)$ [12]. Utilizando as hipóteses da independência e a referente ao ruído (descritas na Seção 2.4) em (50), encontramos:

$$\mathbb{E}\left[\tilde{\boldsymbol{w}}(k+1)\right] = (\boldsymbol{I} - \beta \boldsymbol{R})\, \mathbb{E}\left[\tilde{\boldsymbol{w}}(k)\right], \tag{51}$$

onde $\mathbb{E}\left[e_0(k)\boldsymbol{x}(k)\right] = \boldsymbol{0}$ devido ao princípio da ortogonalidade (vide Equação (15)). Ex-

[2]Tal hipótese foi descrita na Seção 2.4.

pressando a recursão (51) em termos do valor esperado do valor inicial do vetor de desvios, temos

$$\mathbb{E}\left[\tilde{\boldsymbol{w}}(k+1)\right] = (\boldsymbol{I} - \beta\boldsymbol{R})\,\mathbb{E}\left[\tilde{\boldsymbol{w}}(0)\right]. \tag{52}$$

Descrita em termos dos elementos individuais do valor esperado do vetor de desvios $\tilde{\boldsymbol{w}}(k)$, a Equação (51) pode ser reescrita como

$$\mathbb{E}\left[\tilde{w}_i(k+1)\right] = (1 - \beta r_{ii})\mathbb{E}\left[\tilde{w}_i(k)\right] - \beta \sum_{j=0, j\neq i}^{N-1} r_{ij}\mathbb{E}\left[\tilde{w}_j(k)\right], \tag{53}$$

a qual revela um acoplamento entre as dinâmicas dos diferentes elementos $\mathbb{E}\left[\tilde{w}_i(k)\right]$.

Devido à característica hermitiana da matriz $\boldsymbol{R}$, ela pode ser decomposta como

$$\boldsymbol{R} = \boldsymbol{Q}\boldsymbol{\Lambda}\boldsymbol{Q}^T, \tag{54}$$

com $\boldsymbol{\Lambda}$ sendo uma matriz diagonal contendo em seu i-ésimo elemento o i-ésimo autovalor $\lambda_i \in \mathbb{R}_+$ de $\boldsymbol{R}$, e $\boldsymbol{Q} \in \mathbb{R}^{N\times N}$ uma matriz ortogonal que contém em sua i-ésima coluna o i-ésimo autovetor $\boldsymbol{q}_i \in \mathbb{R}^N$ de $\boldsymbol{R}$.

Definindo-se $\breve{\boldsymbol{w}}(k)$ como uma versão *rotacionada* de $\tilde{\boldsymbol{w}}(k)$, ou seja

$$\breve{\boldsymbol{w}}(k) \triangleq \mathbb{E}\left[\boldsymbol{Q}^T\tilde{\boldsymbol{w}}(k)\right], \tag{55}$$

a recursão (51) pode ser mobilizada para provar que

$$\mathbb{E}\left[\breve{\boldsymbol{w}}(k+1)\right] = (\boldsymbol{I} - \beta\boldsymbol{\Lambda})\mathbb{E}\left[\breve{\boldsymbol{w}}(k)\right], \tag{56}$$

a qual gera uma versão de (53) cuja dinâmica é agora *desacoplada* (já que $\boldsymbol{\Lambda}$ é diagonal):

$$\mathbb{E}\left[\breve{w}_i(k+1)\right] = (1 - \beta\lambda_i)\mathbb{E}\left[\breve{w}_i(k)\right], \tag{57}$$

a qual pode ser utilizada para provar que o algoritmo LMS é estável (em termos de estatísticas de primeira ordem) caso as seguintes restrições sejam respeitadas:

$$0 < \beta < \frac{1}{\lambda_{\text{max}}}, \tag{58}$$

onde λ_{max} é o maior autovalor da matriz $\boldsymbol{R}$. Infelizmente, na prática, as estatísticas do

sinal de entrada não são conhecidas *a priori*, de modo que o cálculo do limitante superior em (58) é de difícil execução.

4.2- Análise de Segunda Ordem do algoritmo LMS

A metodologia de análise da seção anterior, sintetizada na Equação (56), é caracterizada como uma análise de primeira ordem, já que as estatísticas dos coeficientes adaptativos $\mathbb{E}[\tilde{w}_i(k)]$ referem-se apenas a valores médios. Tal propriedade torna tal análise muitas das vezes pouco informativa, já que ela nada consegue prever acerca da variância dos coeficientes adaptativos. A variabilidade de tais coeficientes é uma medida importante, já que em alguns contextos ela pode divergir (indo para valores infinitos ao longo das iterações), gerando um fenômeno conhecido como *divergência*, o qual tem como efeito a completa inutilização do algoritmo.

Para contornar tal limitação, é imperioso incluir estatísticas de segunda ordem na análise. Uma das técnicas mais populares de análise de segunda ordem consiste no cômputo de uma recursão referente à matriz de autocorrelação dos desvios, definida por

$$\boldsymbol{R}_{\tilde{\boldsymbol{w}}}(k) \triangleq \mathbb{E}\left[\tilde{\boldsymbol{w}}(k)\tilde{\boldsymbol{w}}^T(k)\right]. \tag{59}$$

Multiplicando-se (59) por sua transposta e aplicando o operador de valor esperado, obtemos:

$$\begin{aligned}
\boldsymbol{R}_{\tilde{\boldsymbol{w}}}(k+1) \;=\; & \mathbb{E}\left\{\left[\boldsymbol{I}-\beta\boldsymbol{x}(k)\boldsymbol{x}^T(k)\right]\tilde{\boldsymbol{w}}(k)\tilde{\boldsymbol{w}}^T(k)\left[\boldsymbol{I}-\beta\boldsymbol{x}(k)\boldsymbol{x}^T(k)\right]^T\right\} \\
& +\mathbb{E}\left\{\boldsymbol{I}-\beta\boldsymbol{x}(k)\boldsymbol{x}^T(k)\tilde{\boldsymbol{w}}(k)\beta e_0(k)\boldsymbol{x}^T(k)\right\} \\
& +\beta\mathbb{E}\left\{e_0(k)\boldsymbol{x}(k)\tilde{\boldsymbol{w}}^T(k)\left[\boldsymbol{I}-\beta\boldsymbol{x}(k)\boldsymbol{x}^T(k)\right]^T\right\} \\
& +\beta^2\mathbb{E}\left[e_0^2(k)\boldsymbol{x}(k)\boldsymbol{x}^T(k)\right].
\end{aligned} \tag{60}$$

Se considerarmos que $e_0(k)$ é independente de $\tilde{\boldsymbol{w}}(k)$ e é ortogonal a $\boldsymbol{x}(k)$, temos:

$$\begin{aligned}\boldsymbol{R}_{\tilde{\boldsymbol{w}}}(k+1) &= \boldsymbol{R}_{\tilde{\boldsymbol{w}}}(k) - \beta\mathbb{E}\left[\boldsymbol{x}(k)\boldsymbol{x}^T(k)\tilde{\boldsymbol{w}}(k)\tilde{\boldsymbol{w}}^T(k)\right] - \beta\mathbb{E}\left[\tilde{\boldsymbol{w}}(k)\tilde{\boldsymbol{w}}^T(k)\boldsymbol{x}(k)\boldsymbol{x}^T(k)\right] \\ &\quad +\beta^2\mathbb{E}\left[\boldsymbol{x}(k)\boldsymbol{x}^T(k)\tilde{\boldsymbol{w}}(k)\tilde{\boldsymbol{w}}^T(k)\boldsymbol{x}(k)\boldsymbol{x}^T(k)\right] \\ &\quad +\beta^2\mathbb{E}\left[e_0^2(k)\boldsymbol{x}(k)\boldsymbol{x}^T(k)\right], \end{aligned} \tag{61}$$

equação que ainda não é capaz de estabelecer uma recursão de $\boldsymbol{R}_{\tilde{\boldsymbol{w}}}(k+1)$ em termos de $\boldsymbol{R}_{\tilde{\boldsymbol{w}}}(k)$. Para este propósito, assumiremos, pela hipótese da independência, que $\tilde{\boldsymbol{w}}(k)$ e $\boldsymbol{x}(k)$ são independentes. Tal hipótese tende a ser acurada quando β assume valores baixos, já que as amostras mais recentes do sinal de entrada $x(k)$ em tese influenciariam muito pouco os atuais coeficientes adaptativos $\tilde{\boldsymbol{w}}(k)$. Desse modo, após algumas simplificações, a Equação (61) se transforma em

$$\boldsymbol{R}_{\tilde{\boldsymbol{w}}}(k+1) = \boldsymbol{R}_{\tilde{\boldsymbol{w}}}(k) - \beta\boldsymbol{R}\boldsymbol{R}_{\tilde{\boldsymbol{w}}}(k) - \beta\boldsymbol{R}_{\tilde{\boldsymbol{w}}}(k) + \beta^2\boldsymbol{A}(k) + \beta^2\sigma_\nu^2\boldsymbol{R}, \tag{62}$$

onde

$$\boldsymbol{A}(k) \triangleq \mathbb{E}\left\{\boldsymbol{x}(k)\boldsymbol{x}^T(k)\mathbb{E}\left[\tilde{\boldsymbol{w}}(k)\tilde{\boldsymbol{w}}^T(k)\right]\boldsymbol{x}(k)\boldsymbol{x}^T(k)\right\} \tag{63}$$

e σ_ν^2 é a variância do ruído de medição em (46). Assumindo que os sinais de entrada são conjuntamente gaussianos, a matriz $\boldsymbol{A}(k)$ em (63) pode ser reescrita como [40]

$$\boldsymbol{A}(k) = 2\boldsymbol{R}\boldsymbol{R}_{\tilde{\boldsymbol{w}}(k)}\boldsymbol{R} + \boldsymbol{R}\mathsf{Tr}\left\{\boldsymbol{R}\boldsymbol{R}_{\tilde{\boldsymbol{w}}}(k)\right\}. \tag{64}$$

A Equação (62) apresenta um formato mais conveniente caso a multipliquemos à esquerda por $\boldsymbol{Q}^T$ e à direita por $\boldsymbol{Q}$, o que, após algumas manipulações matemáticas, dá origem a

$$\begin{aligned}\breve{\boldsymbol{R}}_{\tilde{\boldsymbol{w}}}(k+1) &= \breve{\boldsymbol{R}}_{\tilde{\boldsymbol{w}}}(k) - \beta\boldsymbol{\Lambda}\breve{\boldsymbol{R}}_{\tilde{\boldsymbol{w}}}(k) - \beta\breve{\boldsymbol{R}}_{\tilde{\boldsymbol{w}}}(k)\boldsymbol{\Lambda} + 2\beta^2\boldsymbol{\Lambda}\breve{\boldsymbol{R}}_{\tilde{\boldsymbol{w}}}(k)\boldsymbol{\Lambda} \\ &\quad +\beta^2\boldsymbol{\Lambda}\mathsf{Tr}\left\{\boldsymbol{\Lambda}\breve{\boldsymbol{R}}_{\tilde{\boldsymbol{w}}}(k)\right\} + \beta^2\sigma_\eta^2\boldsymbol{\Lambda}, \end{aligned} \tag{65}$$

onde $\breve{\boldsymbol{R}}_{\tilde{\boldsymbol{w}}}(k) \triangleq \mathbb{E}\left[\boldsymbol{Q}^T\tilde{\boldsymbol{w}}(k)\tilde{\boldsymbol{w}}^T(k)\boldsymbol{Q}\right]$. Utilizando diversas manipulações clássicas na literatura, a Equação (65) pode ser mobilizada para provar que o limite superior de estabilidade teórico para β é

$$\beta < \frac{1}{2\mathsf{Tr}[\boldsymbol{R}]}, \tag{66}$$

o qual é um limite mais restrito do que o preconizado pela análise de primeira ordem (vide Equação (58)), já que

$$\mathsf{Tr}[\boldsymbol{R}] = \sum_{i=0}^{N-1} \lambda_i < \lambda_{\mathsf{max}}, \tag{67}$$

já que os autovalores λ_i, referentes à matriz $\boldsymbol{R}$, são não negativos.

A recursão (62) permite prever a dinâmica da evolução da matriz $\boldsymbol{R}_{\tilde{\boldsymbol{w}}}(k)$, cuja estimativa é importante para prever o MSE ao longo das iterações, já que este pode ser aproximado por meio de [41]

$$\mathsf{MSE}(k) \approx \sigma_\nu^2 + \mathsf{Tr}\left[\boldsymbol{R}\boldsymbol{R}_{\tilde{\boldsymbol{w}}}(k)\right]. \tag{68}$$

4.3- Fluxo de Energia

Uma técnica de análise que será mobilizada no próximo capítulo para análise de regime permanente do algoritmo CMPN é baseada em argumentos de conservação (ou balanceamento) de energia, originalmente proposta em [42]. A técnica atenta para o fluxo de energia ao longo de cada iteração do filtro adaptativo, baseando-se numa relação fundamental de variância do erro. Tal técnica foi estendida em [43] para uma análise de transiente. Por motivos didáticos, com o objetivo de explicitar conceitos importantes para a análise proposta neste trabalho, iremos apresentar a seguir a análise de transiente do algoritmo LMS baseada nesta técnica.

Definamos o erro *a priori* sem ruído ponderado como

$$e_{\mathsf{a}}^{\Sigma}(k) \triangleq \boldsymbol{x}^T(k)\boldsymbol{\Sigma}\tilde{\boldsymbol{w}}(k), \tag{69}$$

onde $\boldsymbol{\Sigma}$ é uma matriz simétrica positiva definida qualquer. No caso particular em que $\boldsymbol{\Sigma} = \boldsymbol{I}$, definimos:

$$e_{\mathsf{a}}(k) \triangleq e_{\mathsf{a}}^{\boldsymbol{I}}(k) = \boldsymbol{x}^T(k)$$

Existe uma relação entre o erro $e(k)$ e o erro *a priori* $e_{\mathsf{a}}(k)$, já que

$$e(k) = d(k) - y(k) = [\boldsymbol{w}^\star]^T\boldsymbol{x}(k) + \nu(k) - \boldsymbol{w}^T(k)\boldsymbol{x}(k) = e_{\mathsf{a}}(k) + \nu(k), \tag{71}$$

de modo que, assumindo-se que o erro $\nu(k)$ é estatisticamente independente das demais variáveis aleatórias, temos

$$\mathsf{MSE}(k) = \mathbb{E}\left[e^2(k)\right] = \mathbb{E}\left[e_{\mathsf{a}}^2(k)\right] + \sigma_\nu^2, \tag{72}$$

o que significa que a predição do valor esperado de $e_{\mathsf{a}}^2(k)$ nos permite quantificar o MSE.

No caso do LMS, a conjunção de (21) e de (44) dá origem à seguinte recursão para o vetor de desvios:

$$\tilde{\boldsymbol{w}}(k+1) = \tilde{\boldsymbol{w}}(k) - \beta \boldsymbol{x}(k) e(k). \tag{73}$$

Multiplicando ambos os lados de (73) por $\boldsymbol{x}^T(k)\boldsymbol{\Sigma}$[3] dá origem a

$$\boldsymbol{x}^T(k)\boldsymbol{\Sigma}\tilde{\boldsymbol{w}}(k+1) = \boldsymbol{x}^T(k)\boldsymbol{\Sigma}\tilde{\boldsymbol{w}}(k) - \beta \boldsymbol{x}^T(k)\boldsymbol{\Sigma}\boldsymbol{x}(k) e(k), \tag{74}$$

ou, equivalentemente:

$$e_{\mathsf{p}}^{\boldsymbol{\Sigma}}(k) = e_{\mathsf{a}}^{\boldsymbol{\Sigma}}(k) - \beta \|\boldsymbol{x}(k)\|_{\boldsymbol{\Sigma}}^2 e(k). \tag{75}$$

A matriz Σ, presente nas Equações (74) e (75), tem como propósito calcular diferentes medidas de performance de filtro adaptativos [23], no entanto, é apresentada neste texto apenas para fins didáticos.

Isolando o erro $e(k)$ em (75) e substituindo o resultado em (73), encontramos, após algumas poucas manipulações algébricas:

$$\tilde{\boldsymbol{w}}(k+1) + \frac{\boldsymbol{x}(k) e_{\mathsf{a}}^{\boldsymbol{\Sigma}}(k)}{\|\boldsymbol{x}(k)\|_{\boldsymbol{\Sigma}}^2} = \tilde{\boldsymbol{w}}(k) + \frac{\boldsymbol{x}(k) e_{\mathsf{p}}^{\boldsymbol{\Sigma}}(k)(k)}{\|\boldsymbol{x}(k)\|^2}. \tag{76}$$

Seja $\|\boldsymbol{w}\|_{\boldsymbol{A}}^2 \triangleq \boldsymbol{w}^T \boldsymbol{A} \boldsymbol{w}$ a *energia do vetor $\boldsymbol{w}$ ponderada pela matriz $\boldsymbol{A}$*. Calculando a energia ponderada de ambos os lados da Equação (76), temos, após algumas simplificações:

$$\|\tilde{\boldsymbol{w}}(k+1)\|_{\boldsymbol{\Sigma}}^2 = \|\tilde{\boldsymbol{w}}(k)\|_{\boldsymbol{\Sigma}}^2 - 2\beta e_{\mathsf{a}}^{\boldsymbol{\Sigma}}(k) e(k) + \beta^2 \|\boldsymbol{x}(k)\|_{\boldsymbol{\Sigma}}^2 e^2(k), \tag{77}$$

a qual é uma equação *exata* que se aplica a todas iterações de qualquer realização do processo de aprendizagem do algoritmo LMS.

Um comportamento médio (isto é, que leve em consideração todas as possíveis

[3] Cumpre notar que há grande flexibilidade na escolha de $\boldsymbol{\Sigma}$, bastando apenas que tal matriz seja simétrica e positiva definida.

realizações do aprendizado) pode ser obtido mediante a aplicação do operador de valor esperado em (77):

$$\mathbb{E}\left[\|\tilde{\boldsymbol{w}}(k+1)\|^2_{\boldsymbol{\Sigma}}\right] = \mathbb{E}\left[\|\tilde{\boldsymbol{w}}(k)\|^2_{\boldsymbol{\Sigma}}\right] - 2\beta \underbrace{\mathbb{E}\left[e_{\mathsf{a}}^{\boldsymbol{\Sigma}}(k)e(k)\right]}_{=\textcircled{a}} + \beta^2 \underbrace{\mathbb{E}\left[\|\boldsymbol{x}(k)\|^2_{\boldsymbol{\Sigma}}e^2(k)\right]}_{=\textcircled{b}}, \tag{78}$$

onde pode-se demonstrar (consulte [43] para maiores detalhes) as seguintes identidades:

$$\textcircled{a} = \mathbb{E}\left[\|\tilde{\boldsymbol{w}}(k)\|^2_{\boldsymbol{\Sigma}\boldsymbol{x}(k)\boldsymbol{x}^T(k)}\right] \tag{79}$$

$$\textcircled{b} = \mathbb{E}\left[\|\boldsymbol{x}(k)\|^2_{\boldsymbol{\Sigma}}\right]\mathbb{E}\left[e^2(k)\right]. \tag{80}$$

Aplicando-se (79) e (80) em (78), temos:

$$\mathbb{E}\left[\|\tilde{\boldsymbol{w}}(k+1)\|^2_{\boldsymbol{\Sigma}}\right] = \mathbb{E}\left[\|\tilde{\boldsymbol{w}}(k)\|^2_{\boldsymbol{\Sigma}}\right] - 2\beta\mathbb{E}\left[\|\tilde{\boldsymbol{w}}(k)\|^2_{\boldsymbol{\Sigma}\boldsymbol{x}(k)\boldsymbol{x}^T(k)}\right] + \beta^2\mathbb{E}\left[\|\boldsymbol{x}(k)\|^2_{\boldsymbol{\Sigma}}\right]\mathbb{E}\left[e^2(k)\right]. \tag{81}$$

Utilizando-se a hipótese da independência (abordada na Seção 2.4), temos:

$$\mathbb{E}\left[\|\tilde{\boldsymbol{w}}(k+1)\|^2_{\boldsymbol{\Sigma}}\right] = \mathbb{E}\left[\|\tilde{\boldsymbol{w}}(k)\|^2_{\boldsymbol{\Sigma}}\right] - 2\beta\mathbb{E}\left[\|\tilde{\boldsymbol{w}}(k)\|^2_{\boldsymbol{\Sigma}\boldsymbol{R}}\right] + \beta^2\mathbb{E}\left[\|\boldsymbol{x}(k)\|^2_{\boldsymbol{\Sigma}}\right]\mathbb{E}\left[e^2(k)\right]. \tag{82}$$

Utilizando em (82) a Equação (72), e o fato de que

$$\mathbb{E}\left[e_{\mathsf{a}}^2(k)\right] = \mathbb{E}\left[\|\tilde{\boldsymbol{w}}(k)\|^2_{\boldsymbol{R}}\right], \tag{83}$$

temos:

$$\begin{aligned}\mathbb{E}\left[\|\tilde{\boldsymbol{w}}(k+1)\|^2_{\boldsymbol{\Sigma}}\right] &= \mathbb{E}\left[\|\tilde{\boldsymbol{w}}(k)\|^2_{\boldsymbol{\Sigma}}\right] - 2\beta\mathbb{E}\left[\|\tilde{\boldsymbol{w}}(k)\|^2_{\boldsymbol{\Sigma}\boldsymbol{R}}\right] + \beta^2\mathbb{E}\left[\|\boldsymbol{x}(k)\|^2_{\boldsymbol{\Sigma}}\right]\mathbb{E}\left[\|\tilde{\boldsymbol{w}}(k)\|^2_{\boldsymbol{R}}\right] \\ &\quad +\beta^2\sigma_\nu^2\mathbb{E}\left[\|\boldsymbol{x}(k)\|^2_{\boldsymbol{\Sigma}}\right],\end{aligned} \tag{84}$$

onde a matriz $\boldsymbol{\Sigma}$ pode ser escolhida. Fazendo a escolha $\boldsymbol{\Sigma} = \boldsymbol{I}$, temos:

$$\mathbb{E}\left[\|\tilde{\boldsymbol{w}}(k+1)\|^2\right] = \mathbb{E}\left[\|\tilde{\boldsymbol{w}}(k)\|^2\right] + (\beta^2\mathbb{E}\left[\|\boldsymbol{x}(k)\|^2\right] - 2\beta)\mathbb{E}\left[\|\tilde{\boldsymbol{w}}(k)\|^2_{\boldsymbol{R}}\right] + \beta^2\sigma_\nu^2\mathbb{E}\left[\|\boldsymbol{x}(k)\|^2\right], \tag{85}$$

o que infelizmente não é uma recursão auto-contida, já que à direita temos uma

dependência com relação a $\mathbb{E}\left[\|\tilde{\boldsymbol{w}}(k)\|^2_{\boldsymbol{R}}\right]$. Fazendo-se $\Sigma = \boldsymbol{R}$, obtemos:

$$\begin{aligned}\mathbb{E}\left[\|\tilde{\boldsymbol{w}}(k+1)\|^2_{\boldsymbol{R}}\right] &= \mathbb{E}\left[\|\tilde{\boldsymbol{w}}(k)\|^2_{\boldsymbol{R}}\right] - 2\beta\mathbb{E}\left[\|\tilde{\boldsymbol{w}}(k)\|^2_{\boldsymbol{R}^2}\right] + \beta^2\mathbb{E}\left[\|\boldsymbol{x}(k)\|^2_{\boldsymbol{R}}\right]\mathbb{E}\left[\|\tilde{\boldsymbol{w}}(k)\|^2_{\boldsymbol{R}}\right] \\ &\quad +\beta^2\sigma_\nu^2\mathbb{E}\left[\|\boldsymbol{x}(k)\|^2_{\boldsymbol{R}}\right]. \end{aligned} \tag{86}$$

Observa-se entretanto que as recursões (85) e (86) ainda não permitem obter as grandezas de interesse $\mathbb{E}\left[\|\tilde{\boldsymbol{w}}(k)\|^2\right]$ e $\mathbb{E}\left[\|\tilde{\boldsymbol{w}}(k)\|^2_{\boldsymbol{R}}\right]$, pois há a necessidade também de se ter uma recursão em termos de $\mathbb{E}\left[\|\tilde{\boldsymbol{w}}(k)\|^2_{\boldsymbol{R}^2}\right]$. Assim, torna-se necessária a inserção de uma recursão em termos $\mathbb{E}\left[\|\tilde{\boldsymbol{w}}(k+1)\|^2_{\boldsymbol{R}^2}\right]$, o que exigiria uma recursão em termos de $\mathbb{E}\left[\|\tilde{\boldsymbol{w}}(k+1)\|^2_{\boldsymbol{R}^3}\right]$, e assim por diante. Em termos gerais, podemos estabelecer a seguinte recursão para a escolha $\boldsymbol{\Sigma} = \boldsymbol{R}^n$:

$$\begin{aligned}\mathbb{E}\left[\|\tilde{\boldsymbol{w}}(k+1)\|^2_{\boldsymbol{R}^n}\right] &= \mathbb{E}\left[\|\tilde{\boldsymbol{w}}(k)\|^2_{\boldsymbol{R}^n}\right] - 2\beta\mathbb{E}\left[\|\tilde{\boldsymbol{w}}(k)\|^2_{\boldsymbol{R}^{n+1}}\right] + \beta^2\mathbb{E}\left[\|\boldsymbol{x}(k)\|^2_{\boldsymbol{R}^n}\right]\mathbb{E}\left[\|\tilde{\boldsymbol{w}}(k)\|^2_{\boldsymbol{R}}\right] \\ &\quad +\beta^2\sigma_\nu^2\mathbb{E}\left[\|\boldsymbol{x}(k)\|^2_{\boldsymbol{R}^n}\right]. \end{aligned} \tag{87}$$

Embora tais procedimentos recursivos pareçam à primeira vista não ter fim, podemos aplicar o teorema de Cayley-Hamilton, obtendo:

$$\boldsymbol{R}^N = -p_0\boldsymbol{I} - p_1\boldsymbol{R} - \ldots - p_{N-1}\boldsymbol{R}^{N-1}, \tag{88}$$

onde

$$p(x) \triangleq \det(x\boldsymbol{I} - \boldsymbol{R}) = p_0 + p_1 x + \ldots + p_{N-1}x^{N-1} + x^N \tag{89}$$

é o polinômio característico de $\boldsymbol{R}$, o que implica a identidade:

$$\|\tilde{\boldsymbol{w}}(k)\|^2_{\boldsymbol{R}^N} = -p_0\|\tilde{\boldsymbol{w}}(k)\|^2 - p_1\|\tilde{\boldsymbol{w}}(k)\|^2_{\boldsymbol{R}} - \ldots - p_{N-1}\|\tilde{\boldsymbol{w}}(k)\|^2_{\boldsymbol{R}^{N-1}}, \tag{90}$$

de modo que podemos terminar o processo recursivo, já que $\|\tilde{\boldsymbol{w}}(k)\|^2_{\boldsymbol{R}^N}$ depende de $\|\tilde{\boldsymbol{w}}(k)\|^2_{\boldsymbol{R}^n}$, com $n < N$. Assim, pode-se estabelecer o comportamento em transiente do filtro por meio da seguinte recursão por uma equação de espaços de estados linear e invariante no tempo:

$$\boldsymbol{\mathcal{W}}(k+1) = \boldsymbol{\mathcal{A}}\boldsymbol{\mathcal{W}}(k) + \beta^2\sigma_\nu^2\boldsymbol{y}, \tag{91}$$

onde

$$\boldsymbol{\mathcal{W}}(k) \triangleq \begin{bmatrix} \mathbb{E}\left[\|\tilde{\boldsymbol{w}}(k)\|^2\right] \\ \mathbb{E}\left[\|\tilde{\boldsymbol{w}}(k)\|^2_{\boldsymbol{R}}\right] \\ \vdots \\ \mathbb{E}\left[\|\tilde{\boldsymbol{w}}(k)\|^2_{\boldsymbol{R}^{N-1}}\right] \end{bmatrix}, \tag{92}$$

$$\boldsymbol{\mathcal{A}} \triangleq \begin{bmatrix} 1 & \beta^2\mathbb{E}\left[\|\boldsymbol{x}(k)\|^2\right] - 2\beta & 0 & 0 & \dots & 0 \\ 0 & 1 + \beta^2\mathbb{E}\left[\|\boldsymbol{x}(k)\|^2_{\boldsymbol{R}}\right] & -2\beta & 0 & \dots & 0 \\ 0 & \beta^2\mathbb{E}\left[\|\boldsymbol{x}(k)\|^2_{\boldsymbol{R}^2}\right] & 1 & -2\beta & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 2\beta p_0 & \beta^2\mathbb{E}\left[\|\boldsymbol{x}(k)\|^2_{\boldsymbol{R}^{N-1}}\right] + 2\beta p_1 & 2\beta p_2 & 2\beta p_3 & \dots & 1 + 2\beta p_{N-1} \end{bmatrix}, \tag{93}$$

$$\boldsymbol{y} \triangleq \begin{bmatrix} \mathbb{E}\left[\|\boldsymbol{x}(k)\|^2\right] \\ \mathbb{E}\left[\|\boldsymbol{x}(k)\|^2_{\boldsymbol{R}}\right] \\ \mathbb{E}\left[\|\boldsymbol{x}(k)\|^2_{\boldsymbol{R}^2}\right] \\ \vdots \\ \mathbb{E}\left[\|\boldsymbol{x}(k)\|^2_{\boldsymbol{R}^{N-1}}\right] \end{bmatrix}. \tag{94}$$

A modelagem obtida por meio do sistema de espaço de estados (91) permite-nos prever estabilidade do aprendizado quando os módulos de todos os autovalores da matriz de transição $\boldsymbol{\mathcal{A}}$ são inferiores à unidade. Ademais, efetuando-se sucessivamente iterações, é possível prever tanto a evolução do MSD (contido no primeiro elemento de $\boldsymbol{\mathcal{W}}(k)$) quanto do MSE, o qual pode ser estimado a partir do segundo elemento do vetor $\boldsymbol{\mathcal{W}}(k)$ (acrescido da variância do ruído σ_ν^2).

Como exemplo didático, seja o caso em que o filtro ótimo é dado por

$$\boldsymbol{w}^\star = \begin{bmatrix} 1 & 0,5 & -0,1 \end{bmatrix}^T, \tag{95}$$

o sinal de entrada é obtido pela filtragem de um sinal branco, gaussiano de média zero e variância unitária pelo filtro $H(z) = 1 - 0,8z^{-1}$. O ruído de medição é branco, gaussiano, de média zero e variância $\sigma_\nu^2 = 10^{-6}$. O fator de aprendizagem é $\beta = 5 \times 10^{-2}$. Para esta

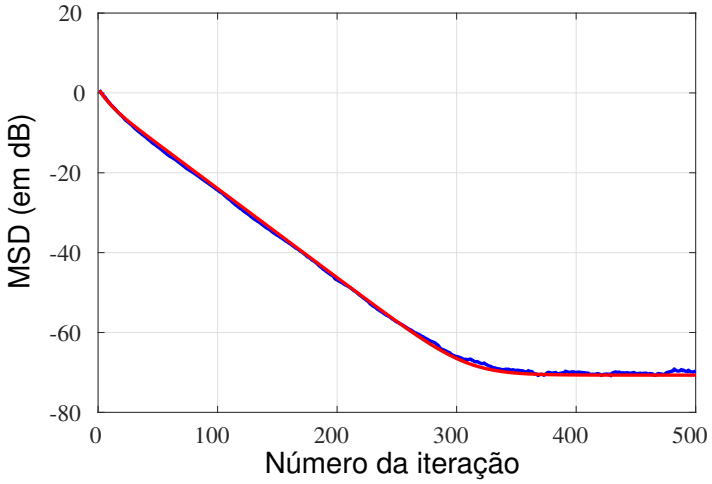


Figura 8 – Evolução do MSD ao longo das iterações. Em vermelho: curva teórica; em azul: curva simulada.

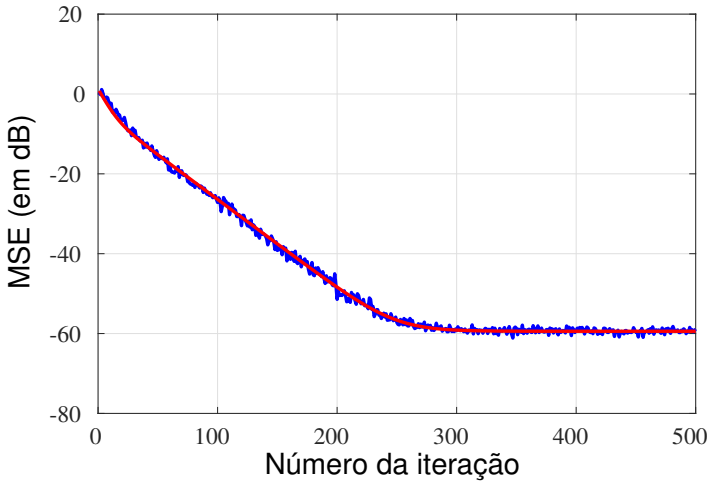


Figura 9 – Evolução do MSE ao longo das iterações. Em vermelho: curva teórica; em azul: curva simulada.

configuração, a matriz $\boldsymbol{\mathcal{A}}$ e o vetor $\boldsymbol{y}$ são:

$$\boldsymbol{\mathcal{A}} = \begin{bmatrix} 1 & -0,087684 & 0 \\ 0 & 1,0266 & -0,1 \\ -0,23117 & 0,74362 & 0,508 \end{bmatrix}, \tag{96}$$

$$\boldsymbol{y} = \begin{bmatrix} 4,92622 \\ 10,6513 \\ 25,8963 \end{bmatrix}. \tag{97}$$

As Figuras 8 e 9 comparam o modelo teórico (91) para o MSD e o MSE com

com a evolução empírica destas métricas, calculada com 100 ensaios independentes de Monte Carlo. Observa-se uma notável aderência das curvas teóricas com as curvas simuladas de execução do algoritmo LMS. No caso do algoritmo CMPN, entretanto, a não linearidade envolvida torna uma análise de transiente (e mesmo de regime permanente) extremamente desafiadora.

4.4- Comentários Finais

Neste capítulo, algumas técnicas de análise de algoritmos de filtragem adaptativas foram apresentadas, focando no algoritmo clássico LMS. A mais clássica destas técnicas, a qual determina uma recursão da matriz de autocorrelação dos desvios $\boldsymbol{R}_{\tilde{\boldsymbol{w}}}(k)$, foi derivada, com alguns *insights* dela extraídos comentados. Infelizmente, esta técnica não é muito adequada para a modelagem de algoritmos de filtragem adaptativa que dependem de maneira extremamente não linear do erro, como é o caso do nosso algoritmo de interesse (CMPN). Assim, o próximo capítulo mobiliza a outra técnica de análise apresentada neste capítulo, baseada em balanceamento de energia, para prever o desempenho assintótico do algoritmo CMPN. Esta predição, viabilizada por uma aproximação da linearidade da equação de atualização do algoritmo CMPN por uma função linear por partes, consiste na contribuição desta dissertação.

5- Análise de Regime Permanente do Algoritmo Adaptativo de Norma Mista

Dentre as técnicas de análise apresentadas no Capítulo 4, descobrimos que a técnica de balanceamento de energia permite, mediante a estratégia de aproximação da função não linear envolvida na equação de atualização do algoritmo CMPN, obter predições para o regime permanente deste algoritmo. Este capítulo por conseguinte detalha este achado.

5.1- Análise de Regime Permanente do Algoritmo CMPN

Inicialmente, consideremos a equação de atualização de um algoritmo adaptativo com uma não linearidade aplicada ao erro $f[e(k)]$:

$$\boldsymbol{w}(k+1) = \boldsymbol{w}(k) + \beta \boldsymbol{x}(k) f[e(k)], \tag{98}$$

a qual contempla o algoritmo CMPN como um caso particular, conforme demonstram as Equações (40) e (41).

A partir de (98), podemos deduzir

$$\tilde{\boldsymbol{w}}(k+1) = \tilde{\boldsymbol{w}}(k) - \beta f[e(k)] \boldsymbol{x}(k) \tag{99}$$

Multiplicando ambos os lados de (99) à esquerda por $\boldsymbol{x}^T(k)\boldsymbol{\Sigma}$, encontramos:

$$\underbrace{\boldsymbol{x}^T(k)\boldsymbol{\Sigma}\tilde{\boldsymbol{w}}(k+1)}_{=e_{\mathsf{p}}^{\boldsymbol{\Sigma}}(k)} = \underbrace{\boldsymbol{x}^T(k)\boldsymbol{\Sigma}\tilde{\boldsymbol{w}}(k)}_{=e_{\mathsf{a}}^{\boldsymbol{\Sigma}}(k)} - \beta f[e(k)] \underbrace{\boldsymbol{x}^T(k)\boldsymbol{\Sigma}\boldsymbol{x}(k)}_{\|\boldsymbol{x}(k)\|_{\boldsymbol{\Sigma}}^2}, \tag{100}$$

de modo que os erros *a priori* e *a posteriori* são relacionados pela identidade seguinte

$$e_{\mathsf{p}}^{\boldsymbol{\Sigma}}(k) = e_{\mathsf{a}}^{\boldsymbol{\Sigma}}(k) - \beta f[e(k)] \|\boldsymbol{x}(k)\|_{\boldsymbol{\Sigma}}^2 \tag{101}$$

$$\Rightarrow \beta f[e(k)] = \frac{e_{\mathsf{a}}^{\boldsymbol{\Sigma}}(k)}{\|\boldsymbol{x}(k)\|_{\boldsymbol{\Sigma}}^2} - \frac{e_{\mathsf{p}}^{\boldsymbol{\Sigma}}(k)}{\|\boldsymbol{x}(k)\|_{\boldsymbol{\Sigma}}^2} \tag{102}$$

Aplicando (102) em (99), temos:

$$\tilde{\boldsymbol{w}}(k+1) = \tilde{\boldsymbol{w}}(k) - \frac{e_{\mathsf{a}}^{\boldsymbol{\Sigma}}(k)\boldsymbol{x}(k)}{\|\boldsymbol{x}(k)\|_{\boldsymbol{\Sigma}}^2} + \frac{e_{\mathsf{p}}^{\boldsymbol{\Sigma}}(k)\boldsymbol{x}(k)}{\|\boldsymbol{x}(k)\|^2} \tag{103}$$

$$\Rightarrow \underbrace{\tilde{\boldsymbol{w}}(k+1) + \frac{e_{\mathsf{a}}^{\boldsymbol{\Sigma}}(k)\boldsymbol{x}(k)}{\|\boldsymbol{x}(k)\|_{\boldsymbol{\Sigma}}^2}}_{\mathsf{LE}} = \underbrace{\tilde{\boldsymbol{w}}(k) + \frac{e_{\mathsf{p}}^{\boldsymbol{\Sigma}}(k)\boldsymbol{x}(k)}{\|\boldsymbol{x}(k)\|^2}}_{\mathsf{LD}}, \tag{104}$$

onde “LE” denota “lado esquerdo” e “LD” denota “lado direito”. A energia do lado esquerdo (ponderada pela matriz $\boldsymbol{\Sigma}$) é igual a

$$\|\tilde{\boldsymbol{w}}(k+1)\|_{\boldsymbol{\Sigma}}^2 + \frac{2e_{\mathsf{a}}^{\boldsymbol{\Sigma}}(k)e_{\mathsf{p}}^{\boldsymbol{\Sigma}}(k)}{\|\boldsymbol{x}(k)\|_{\boldsymbol{\Sigma}}^2} + \frac{\left[e_{\mathsf{a}}^{\boldsymbol{\Sigma}}(k)\right]^2}{\|\boldsymbol{x}(k)\|_{\boldsymbol{\Sigma}}^2}, \tag{105}$$

enquanto que a energia do lado direito é:

$$\|\tilde{\boldsymbol{w}}(k)\|_{\boldsymbol{\Sigma}}^2 + \frac{2e_{\mathsf{a}}^{\boldsymbol{\Sigma}}(k)e_{\mathsf{p}}^{\boldsymbol{\Sigma}}(k)}{\|\boldsymbol{x}(k)\|_{\boldsymbol{\Sigma}}^2} + \frac{\left[e_{\mathsf{p}}^{\boldsymbol{\Sigma}}(k)\right]^2}{\|\boldsymbol{x}(k)\|_{\boldsymbol{\Sigma}}^2} \tag{106}$$

Como as energias dos lados esquerdo e direito são iguais (veja (104)), temos que:

$$\|\tilde{\boldsymbol{w}}(k+1)\|_{\boldsymbol{\Sigma}}^2 + \frac{\left[e_{\mathsf{a}}^{\boldsymbol{\Sigma}}(k)\right]^2}{\|\boldsymbol{x}(k)\|_{\boldsymbol{\Sigma}}^2} = \|\tilde{\boldsymbol{w}}(k)\|_{\boldsymbol{\Sigma}}^2 + \frac{\left[e_{\mathsf{p}}^{\boldsymbol{\Sigma}}(k)\right]^2}{\|\boldsymbol{x}(k)\|_{\boldsymbol{\Sigma}}^2} \tag{107}$$

$$e_{\mathsf{p}}^{\boldsymbol{\Sigma}}(k) = e_{\mathsf{a}}^{\boldsymbol{\Sigma}}(k) - \beta f[e(k)]\|\boldsymbol{x}(k)\|_{\boldsymbol{\Sigma}}^2 \tag{108}$$

$$\left[e_{\mathsf{p}}^{\boldsymbol{\Sigma}}(k)\right]^2 = \left[e_{\mathsf{a}}^{\boldsymbol{\Sigma}}(k)\right]^2 - 2\beta f[e(k)]e_{\mathsf{a}}^{\boldsymbol{\Sigma}}(k)\|\boldsymbol{x}(k)\|_{\boldsymbol{\Sigma}}^2 + \beta^2 f^2[e(k)]\|\boldsymbol{x}(k)\|_{\boldsymbol{\Sigma}}^4 \tag{109}$$

A aplicação de (109) em (107) leva a

$$\|\tilde{\boldsymbol{w}}(k+1)\|_{\boldsymbol{\Sigma}}^2 = \|\tilde{\boldsymbol{w}}(k)\|_{\Sigma}^2 - 2\beta f[e(k)]e_{\mathsf{a}}^{\boldsymbol{\Sigma}}(k) + \beta^2 f^2[e(k)]\|\boldsymbol{x}(k)\|_{\boldsymbol{\Sigma}}^2, \tag{110}$$

a qual é uma equação exata, aplicável a todas as iterações do algoritmo em qualquer execução dada. Para que possamos extrair implicações sobre o comportamento médio do algoritmo, cabe-nos aplicar o operador de valor esperado, obtendo:

$$\mathbb{E}\left[\|\tilde{\boldsymbol{w}}(k+1)\|_{\boldsymbol{\Sigma}}^2\right] = \mathbb{E}\left[\|\tilde{\boldsymbol{w}}(k)\|_{\boldsymbol{\Sigma}}^2\right] - 2\beta \underbrace{\mathbb{E}\left\{e_{\mathsf{a}}^{\boldsymbol{\Sigma}}(k)f[e(k)]\right\}}_{=\textcircled{\mathrm{I}}} + \beta^2 \underbrace{\mathbb{E}\left\{\|\boldsymbol{x}(k)\|_{\boldsymbol{\Sigma}}^2 f^2[e(k)]\right\}}_{=\textcircled{\mathrm{II}}} \tag{111}$$

Devido à extrema não linearidade da função $f[e(k)]$ (verificável em (41)), os cálculos de Ⓘ e Ⓘ Ⓘ não apresentam solução analítica. Esta, acreditamos ser, uma das razões pela qual o algoritmo CMPN não foi ainda modelado teoricamente na literatura correlata. Para contornar este problema, nesta dissertação propomos aproximar a função $f[e(k)]$ por uma versão não linear mais simples, a qual apresenta a propriedade de ser *linear por partes*. Assim, almejando-se a obtenção de fórmulas analíticas, propomos a seguinte aproximação:

$$f[e(k)] \approx \begin{cases} 0, & \text{se } e(k) = 0 \\ b_0 + a_0 e(k), & \text{se } 0 < e(k) \leq \delta_1 \\ b_1 + a_1 e(k), & \text{se } \delta_1 < e(k) \\ -b_0 + a_0 e(k), & \text{se } -\delta_1 \leq e(k) < 0 \\ -b_1 + a_1 e(k), & \text{se } e(k) < -\delta_1 \end{cases}, \tag{112}$$

e $f[e(k)] = -f[-e(k)]$ se $e(k) < 0$.

Devemos garantir que a aproximação é contínua; ou seja, descabe haver descontinuidades no ponto $e(k) = \delta_1$. Matematicamente, tal restrição implica

$$b_1 + a_1\delta_1 = b_0 + a_0\delta_1 \Rightarrow b_1 = b_0 + a_0\delta_1 - a_1\delta_1, \tag{113}$$

significando que basta calcular por *least-squares* ((LS)) o valor de a_1 e depois calcular b_1 usando (113). A Figura 10 compara a função não linear original com sua aproximação linear por partes proposta.

Assim, entendendo-se que há $\overline{N}$ pontos para o cômputo de a_0 e b_0 via LS, devemos encontrá-los de modo a minimizar a energia dos resíduos da aproximação

$$\begin{bmatrix} e(k_0) & 1 \\ e(k_1) & 1 \\ \vdots & \vdots \\ e(k_{\overline{N}}) & 1 \end{bmatrix} \begin{bmatrix} a_0 \\ b_0 \end{bmatrix} \approx \begin{bmatrix} f[e(k_0)] \\ f[e(k_1)] \\ \vdots \\ f[e(k_{\overline{N}})] \end{bmatrix} \tag{114}$$

No caso de a_1, temos que

$$a_1 e(k) + b_1 = f[e(k)]. \tag{115}$$

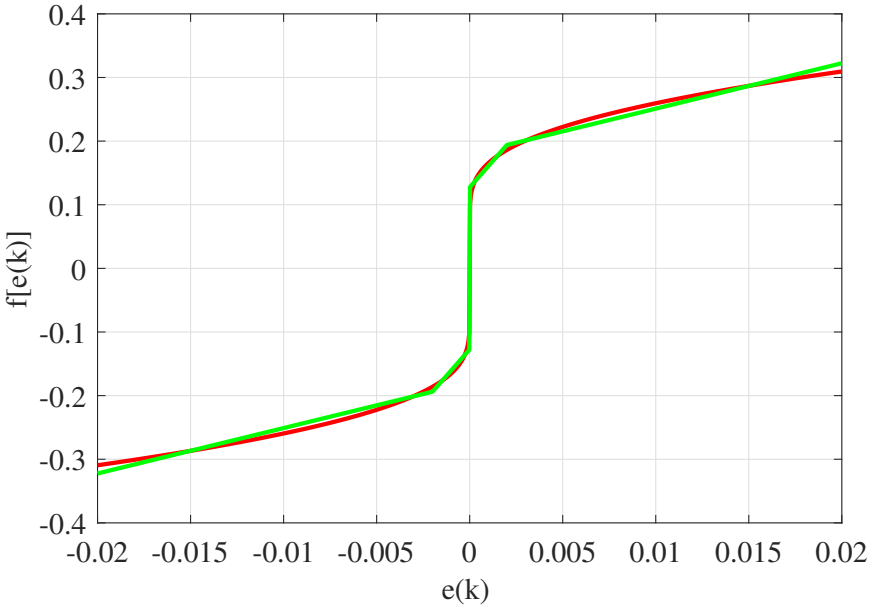


Figura 10 – Função erro $f[e(k)]$ conforme dado pela Equação (41) (em vermelho). Em verde: aproximação linear por partes conforme Equação (112) .

Usando (113), temos

$$a_1 e(k) + b_0 + a_0\delta_1 - a_1\delta_1 = f[e(k)], \tag{116}$$

de modo que, assumindo que δ_1, b_0 e a_0 já foram encontrados via LS, a_1 deve minimizar a energia do resíduo do seguinte sistema

$$\underbrace{\begin{bmatrix} e(k_{\overline{N}+1}) - \delta_1 \\ e(k_{\overline{N}+2}) - \delta_1 \\ \vdots \\ e(k_L) - \delta_1 \end{bmatrix}}_{=\overline{\boldsymbol{x}}} a_1 \approx \underbrace{\begin{bmatrix} f[e(k_{\overline{N}+1})] - b_0 - a_0\delta_1 \\ f[e(k_{\overline{N}+2})] - b_0 - a_0\delta_1 \\ \vdots \\ f[e(k_L)] - b_0 - a_0\delta_1 \end{bmatrix}}_{=\overline{\boldsymbol{y}}}, \tag{117}$$

de modo que a solução LS de (118) é

$$a_1 = \frac{\overline{\boldsymbol{y}}^T \overline{\boldsymbol{x}}}{\overline{\boldsymbol{x}}^T \overline{\boldsymbol{x}}}. \tag{118}$$

5.1.1 Cálculo do termo Termo I da Equação (111)

No caso da aproximação linear por partes (112), o termo $h_{\mathsf{G}}[\xi]$, onde $\xi \triangleq \mathbb{E}[e_{\mathsf{a}}^2(k)]$, pode ser expresso por

$$h_{\mathsf{G}}\left[\xi\right] = \frac{\mathbb{E}\left\{e_a(k) f[e_a(k) + \nu(k)]\right\}}{\xi}, \tag{119}$$

cujo *numerador* pode ser encontrado, assumindo-se uma distribuição genérica $p_\nu(\nu)$ para o ruído $\nu(k)$, a partir de

$$\mathbb{E}\left\{e_a(k) f[e_a(k) + \nu(k)]\right\} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} e_{\mathsf{a}} f[e_{\mathsf{a}} + \nu] \frac{1}{\sqrt{2\pi\xi}} \mathsf{exp}\left[-\frac{e_{\mathsf{a}}^2}{2\xi}\right] p_\nu(\nu) de_{\mathsf{a}} d\nu, \tag{120}$$

onde a distribuição de $e_{\mathsf{a}}(k)$ é suposta gaussiana.

Para obtermos $h_{\mathsf{G}}[\xi]$, cabe-nos calcular (120) para cada um dos intervalos da aproximação linear por partes (112).

Observação: É importante ressaltar que a função não linear do algoritmo CMPN (vide Equação (41)) não permite a obtenção de fórmulas analíticas para integrais como (120), enquanto que a nossa aproximação linear por partes (vide Equação (112)) torna tal obtenção factível. Assim, conjugando-se a análise via balanceamento de energia e a aproximação linear por partes (112) iremos lograr, de maneira inédita, estimativas teóricas para o desempenho assintótico do algoritmo CMPN.

Assumindo operação em regime permanente, a função $f[e(k)]$ na Equação (111) pode ser aproximada por (112). Logo, os termos Ⓘ e Ⓘ Ⓘ em (111) passam a ser calculados analiticamente, viabilizando, pela primeira vez na literatura, uma estimativa teórica do desempenho assintótico do algoritmo CMPN.

Utilizando o Teorema de *Price* [44], que estabelece a relação descrita pela Equação (121), no caso em que x e y sejam variáveis aleatórias conjuntamente gaussianas e independentes de uma terceira variável z:

$$E[x f[y + z]] = \frac{E[xy]}{E[y^2]} E[y f[y + z]], \tag{121}$$

podemos expressar Ⓘ como

$$\begin{aligned} \text{Ⓘ} &= \mathbb{E}\left\{e_{\mathsf{a}}^{\boldsymbol{\Sigma}}(k) f[e(k)]\right\} \\ &= \mathbb{E}\left\{e_{\mathsf{a}}^{\boldsymbol{\Sigma}}(k) e_{\mathsf{a}}(k)\right\} \frac{\mathbb{E}\left\{e_{\mathsf{a}}(k) f\left[e_{\mathsf{a}}(k)+\nu(k)\right]\right\}}{\mathbb{E}\left[e_{\mathsf{a}}^2(k)\right]} \\ &= \mathbb{E}\left\{e_{\mathsf{a}}^{\boldsymbol{\Sigma}}(k) e_{\mathsf{a}}(k)\right\} h_{\mathsf{G}}\left\{\mathbb{E}\left[e_{\mathsf{a}}^2(k)\right]\right\}, \end{aligned} \tag{122}$$

Deste modo, o termo Ⓘ pode ser reescrito:

$$\text{Ⓘ} = \mathbb{E}\left[\|\tilde{\boldsymbol{w}}(k)\|^2_{\boldsymbol{\Sigma}\boldsymbol{x}^T(k)\boldsymbol{x}(k)}\right] h_{\mathsf{G}}\left\{\mathbb{E}\left[e_{\mathsf{a}}^2(k)\right]\right\}. \tag{123}$$

Comentário: Infelizmente, o Teorema de *Price* assume gaussianidade dos sinais envolvidos (incluindo o sinal de erro). Isso significa que a fórmula teórica a ser obtida é válida apenas para o caso de ruído gaussiano. Assim, nesta dissertação encontraremos uma fórmula que prevê o desempenho assintótico do algoritmo CMPN sob condições "normais" de ruído gaussiano. Mesmo sendo inédita tal fórmula, cabe ressaltar que não fomos capazes de prever o desempenho do algoritmo CMPN sob condições de ruído impulsivo. O resultado obtido, entretanto, não é desprovido de mérito, já que na literatura não há predições teóricas para o algoritmo CMPN em nenhuma destas condições. Portanto, o resultado a ser obtido caracteriza o desempenho do algoritmo CMPN em condições normais, o que dá mais respaldo ao projetista, o qual sabe, por meio de simulações extensas e por ampla discussão pregressa na literatura, que mesmo em situações anômalas, com a presença de ruído impulsivo, o algoritmo CMPN ainda terá um desempenho melhor do que algoritmos que se valem de funções custo mais clássicas (como funções custo quadráticas).

5.1.2 Cálculo do Termo II da Equação (111)

Assumindo que os termos $\|\boldsymbol{x}(k)\|^2_{\boldsymbol{\Sigma}}$ e $f^2[e(k)]$ sejam estatisticamente independentes:

$$\begin{aligned} \textcircled{II} &= \mathbb{E}\left\{\|\boldsymbol{x}(k)\|^2_{\boldsymbol{\Sigma}} f^2[e(k)]\right\} \\ &= \mathbb{E}\left\{\|\boldsymbol{x}(k)\|^2_{\boldsymbol{\Sigma}}\right\} h_{\mathsf{U}}\left\{\mathbb{E}\left[e^2_{\mathsf{a}}(k)\right]\right\}, \end{aligned} \tag{124}$$

A aplicação de (123) e de (124) em (111) leva a

$$\begin{aligned} \mathbb{E}\left[\|\tilde{\boldsymbol{w}}(k+1)\|^2_{\boldsymbol{\Sigma}}\right] &= \mathbb{E}\left[\|\tilde{\boldsymbol{w}}(k)\|^2_{\boldsymbol{\Sigma}}\right] - 2\beta\mathbb{E}\left[\|\tilde{\boldsymbol{w}}(k)\|^2_{\boldsymbol{\Sigma}\boldsymbol{x}^T(k)\boldsymbol{x}(k)}\right] h_{\mathsf{G}}\left\{\mathbb{E}\left[e^2_{\mathsf{a}}(k)\right]\right\} \\ &\quad +\beta^2\mathbb{E}\left\{\|\boldsymbol{x}(k)\|^2_{\boldsymbol{\Sigma}}\right\} h_{\mathsf{U}}\left\{\mathbb{E}\left[e^2_{\mathsf{a}}(k)\right]\right\}, \end{aligned} \tag{125}$$

em que

$$h_{\mathsf{U}}[\xi] = \left\{\mathbb{E}\left[e^2_{\mathsf{a}}(k)\right]\right\} = \mathbb{E}[f^2[e(i)]], \tag{126}$$

e

$$f(e(i)) = f[e_a(k) + \nu(k)]. \tag{127}$$

Agregando-se a hipótese da independência em (125), obtemos:

$$\begin{aligned} \mathbb{E}\left[\|\tilde{\boldsymbol{w}}(k+1)\|^2_{\boldsymbol{\Sigma}}\right] &= \mathbb{E}\left[\|\tilde{\boldsymbol{w}}(k)\|^2_{\boldsymbol{\Sigma}}\right] - 2\beta h_{\mathsf{G}}\left\{\mathbb{E}\left[\|\tilde{\boldsymbol{w}}(k)\|^2\right]\right\}\mathbb{E}\left[\|\tilde{\boldsymbol{w}}(k)\|^2_{\boldsymbol{\Sigma}\boldsymbol{R}}\right] \\ &\quad +\beta^2\mathbb{E}\left\{\|\boldsymbol{x}(k)\|^2_{\boldsymbol{\Sigma}}\right\} h_{\mathsf{U}}\left\{\mathbb{E}\left[\|\tilde{\boldsymbol{w}}(k)\|^2_{\boldsymbol{R}}\right]\right\}. \end{aligned} \tag{128}$$

Neste momento, para simplificar a complexidade das equações, iremos inserir uma hipótese adicional: a de que o sinal de entrada é branco. Esta hipótese é comum na literatura, facilitando a obtenção de resultados analíticos [45; 46; 47; 48]. Utilizando esta hipótese, $\boldsymbol{\Sigma} = \boldsymbol{I}$ e $\boldsymbol{R} = \sigma_x^2\boldsymbol{I}$, com σ_x^2 sendo a variância do sinal de entrada, dando origem a:

$$\begin{aligned} \mathbb{E}\left[\|\tilde{\boldsymbol{w}}(k+1)\|^2\right] &= \mathbb{E}\left[\|\tilde{\boldsymbol{w}}(k)\|^2\right] - 2\beta\sigma_x^2 h_{\mathsf{G}}\left\{\sigma_x^2\mathbb{E}\left[\|\tilde{\boldsymbol{w}}(k)\|^2\right]\right\}\mathbb{E}\left[\|\tilde{\boldsymbol{w}}(k)\|^2\right] \\ &\quad +\beta^2\sigma_x^2 N h_{\mathsf{U}}\left\{\sigma_x^2\mathbb{E}\left[\|\tilde{\boldsymbol{w}}(k)\|^2\right]\right\}. \end{aligned} \tag{129}$$

Já que estamos interessados no regime assintótico, assumindo que o algoritmo converge, temos que

$$\lim_{k\to\infty} \mathbb{E}\left[\|\tilde{\boldsymbol{w}}(k+1)\|^2\right] = \lim_{k\to\infty} \mathbb{E}\left[\|\tilde{\boldsymbol{w}}(k)\|^2\right] \tag{130}$$

E por conseguinte (131) dá origem a

$$\sigma_x^2 h_{\mathsf{G}}\left\{\sigma_x^2\mathbb{E}\left[\|\tilde{\boldsymbol{w}}(k)\|^2\right]\right\}\mathbb{E}\left[\|\tilde{\boldsymbol{w}}(k)\|^2\right] = \frac{\beta}{2}\sigma_x^2 N h_{\mathsf{U}}\left\{\sigma_x^2\mathbb{E}\left[\|\tilde{\boldsymbol{w}}(k)\|^2\right]\right\} \tag{131}$$

Fazendo $\xi \triangleq \sigma_x^2\mathbb{E}\left[\|\tilde{\boldsymbol{w}}(k)\|^2\right] = \mathbb{E}[e_{\mathsf{a}}^2(k)]$, temos o seguinte resultado:

$$\xi = \frac{\beta}{2}\mathsf{Tr}[\boldsymbol{R}]\frac{h_{\mathsf{U}}[\xi]}{h_{\mathsf{G}}[\xi]} \tag{132}$$

A Equação (132) prediz que o valor assintótico do erro quadrático médio em excesso $\mathbb{E}[e_{\mathsf{a}}^2(k)]$ (suficiente para estimar o MSE assintótico, segundo (72)) é a solução de uma equação de ponto fixo (descrita por (132)). Cabe ressaltar que embora relação (132) não seja inédita, ela somente pode ser calculada analiticamente por conta da nossa aproximação linear por partes (112).

Uma equação de ponto fixo pode ser resolvida iterativamente de maneira bastante simples, calculando-se, a partir de uma estimativa inicial ξ_0, um novo valor refinado

$$\xi_1 = \frac{\beta}{2}\mathsf{Tr}[\boldsymbol{R}]\frac{h_{\mathsf{U}}[\xi_0]}{h_{\mathsf{G}}[\xi_0]} \tag{133}$$

e refinando o resultado sucessivamente, segundo

$$\xi_{n+1} = \frac{\beta}{2}\mathsf{Tr}[\boldsymbol{R}]\frac{h_{\mathsf{U}}[\xi_n]}{h_{\mathsf{G}}[\xi_n]}, \tag{134}$$

até que a diferença $|\xi_{n+1} - \xi_n|$ seja desprezível.

5.1.3 Cálculo das integrais

Para efeito do cálculo das integrais presentes na Equação (120) serão considerados os valores para as constantes em (112) descritos na Tabela 2.

Tais valores foram obtidos por meio das Equações (113) e (118), com um valor de

Constantes	
a_0	33.2974
a_1	7.1362
b_0	$.0.1273$
b_1	0.1796
δ_1	0.002

Tabela 2 – Valores das constantes adotadas na aproximação linear por partes (112).

δ_1 que gerasse uma aproximação razoável, conforme Figura 10.

5.2- Comentários Finais

Este capítulo descreveu uma aproximação linear por partes da não linearidade adotada no algoritmo CMPN. Quando conjugada com a técnica de análise de balanceamento de energia, esta aproximação dá origem a uma equação de ponto fixo, cuja solução é uma estimativa teórica do desempenho assintótico do algoritmo CMPN. Cabe ressaltar a inexistência na literatura correlatada de uma predição teórica do desempenho do algoritmo CMPN, dada a sua extrema não linearidade. No próximo capítulo, a estimativa teórica obtida é avaliada por meio de comparação com resultados oriundos de simulação.

6- Resultados

No capítulo anterior, vimos que a aproximação linear por partes descrita pela Equação (112) da função custo do algoritmo CMPN descrita através da Equação (41), quando conjugada à técnica de análise de filtragem adaptativa via balanceamento de energia, dá origem a uma equação de ponto fixo, cuja solução dá origem a uma estimativa teórica inédita do desempenho assintótico do algoritmo CMPN. Cabe ressaltar que tal aproximação não é factível para a geração de um novo algoritmo, já que assume valores muito discrepantes com relação à função não linear original quando a magnitude do sinal de erro é elevada. Assim, tal aproximação somente é válida para uma análise teórica do regime assintótico, durante o qual o erro, assumindo-se que o algoritmo converge para uma boa solução, assume magnitudes relativamente baixas. Por essa razão, a análise empreendida foi restrita para o regime de operação em regime permanente do algoritmo. A aferição da qualidade da estimativa teórica será aferida mediante comparações com resultados simulados, por meio de diversos cenários. Em todos eles, o ruído aditivo é um ruído gaussiano, de média zero e variância σ_ν^2. O sinal de entrada é um ruído branco, gaussiano e de variância unitária. O MSE simulado em regime permanente foi estimado por meio de duas operações de média: (i) média ao longo de K_1 ensaios independente de Monte Carlo; (ii) média das últimas K_2 iterações do processo de aprendizagem. Em todos os cenários a seguir, as escolhas $K_1 = 50$ e $K_2 = 100$ foram adotadas, exceto quando tais valores são informados explicitamente.

6.1- Cenários das simulações

De posse da equação de ponto fixo, conforme equação (132), foram gerados 6 cenários distintos para simulação do algoritmo em regime permanente, conforme proposta inicial. A idéia de cada cenário é o seguinte:

- Cenário 1: Variação do valor de β (passo de aprendizagem).

- Cenário 2: Variação do tamanho do filtro, com 8 coeficientes.
- Cenário 3: Variação da variância do ruído e tamanho do filtro, com 5 coeficientes.
- Cenário 4: Sinal de entrada colorido.
- Cenário 5: Idem ao cenário 4 porém com filtro com mais coeficientes.
- Cenário 6: Sinal de referência com ruído impulsivo.

Em todos os cenários a quantidade de iterações e repetições foram iguais a 2×10^6 e 50 respectivamente.

6.1.1 Cenário 1

No Cenário 1, iremos levantar a qualidade da predição teórica numa ampla faixa de passos de aprendizagem. Assim, $\beta = 10^c$, com $c \in \{-5; -4,75; -4,5; \ldots; -2\}$. Para cada valor de β, o algoritmo iterou por 2×10^6 vezes. O vetor de coeficientes ótimos é descrito por[1]

$$\boldsymbol{w}^\star = \begin{bmatrix} 0,5377 \\ 1,8339 \\ -2,2588 \\ 0,8622 \\ 0,3188 \\ -1,3077 \\ -0,4336 \\ 0,3426 \\ 3,5784 \\ 2,7694 \end{bmatrix}, \tag{135}$$

a variância do ruído é $\sigma_\nu^2 = 10^{-6}$ e a equação de ponto fixo (134) é iterada até que a diferença (em dB) entre duas estimativas, em iterações consecutivas, seja menor do que $0,01$. O valor inicial da estimativa ξ no procedimento recursivo é 10^{-5}. A Figura

[1] Ao longo dos cenários, as funções de transferência foram geradas por meio de um gerador de números aleatórios gaussianos. Os valores são apresentados aqui por razões de reprodutibilidade do experimento.

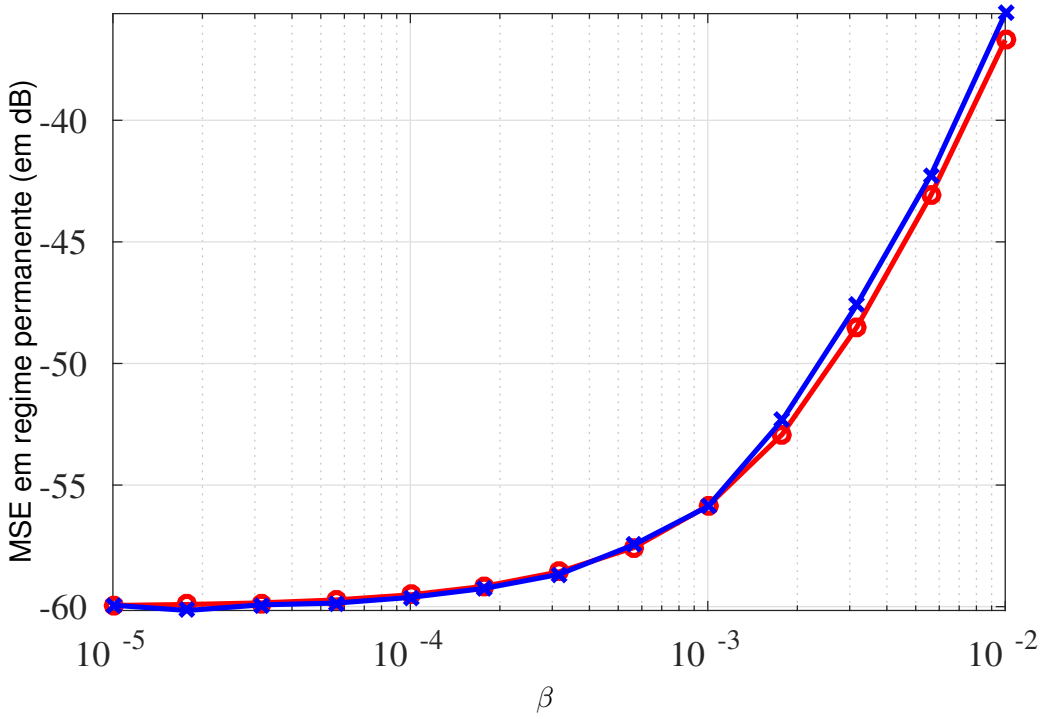


Figura 11 – Comparação do MSE assintótico entre a estimativa teórica (em vermelho) e o resultado de simulações (em azul), referente ao Cenário 1.

11 apresenta a comparação entre o resultado teórico e a curva levantada por meio de simulações. Observa-se aderência entre as curvas, especialmente quando β assume valores menores. À medida que β assume valores elevados, a hipótese da independência torna-se menos precisa, o que explica o relativo descolamento da predição teórica com relação às simulações.

6.1.2 Cenário 2

O objetivo da simulação do Cenário 2 é aquilatar se mudanças no vetor de coeficientes ótimo a estimar ou se alterações de seu tamanho interferem na qualidade da estimativa. Assim, o Cenário 2 preserva todas as configurações do Cenário 1, exceto pelo filtro ótimo de coeficientes, que agora tem comprimento $N = 8$ (enquanto que no

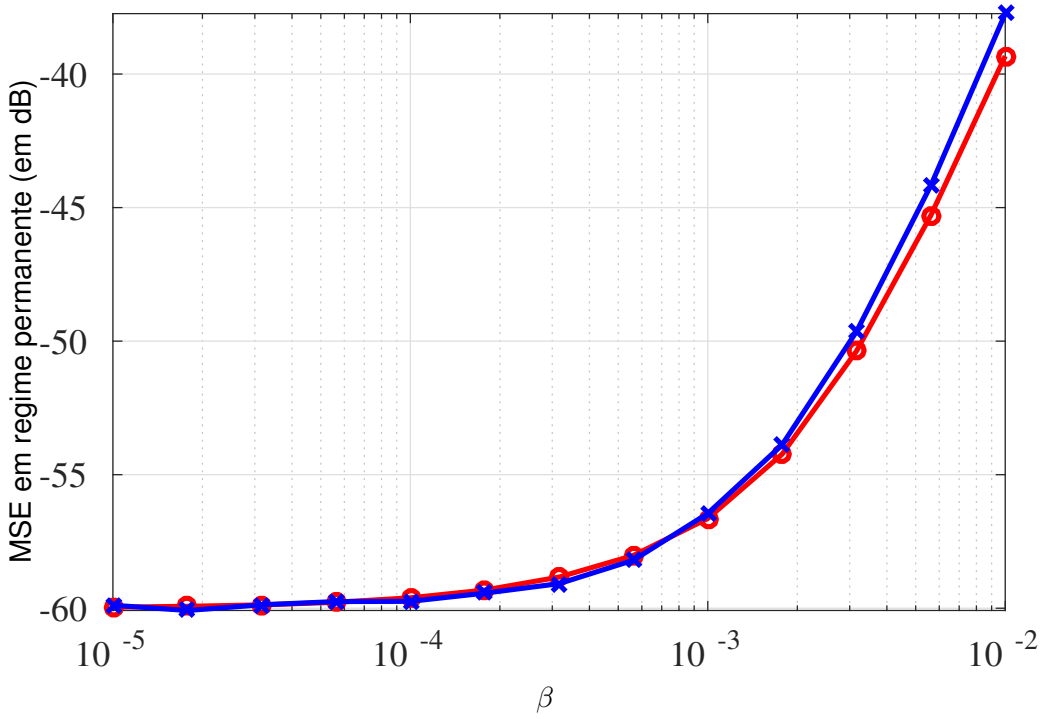


Figura 12 – Comparação do MSE assintótico entre a estimativa teórica (em vermelho) e o resultado de simulações (em azul), referente ao Cenário 2.

Cenário 1 era $N = 10$), com seus coeficientes sendo descritos por:

$$\boldsymbol{w}^{\star} = \begin{bmatrix} -1,3499 \\ 3,0349 \\ 0,7254 \\ -0,0631 \\ 0,7147 \\ -0,2050 \\ -0,1241 \\ 1,4897 \end{bmatrix}. \tag{136}$$

A Figura 12 permite comparar a previsão teórica do desempenho em regime permanente com os resultados oriundos de simulação. Não observa-se diferença na acurácia com relação ao Cenário 1, indicando que o modelo proposto é robusto a alterações no vetor de coeficientes ótimo.

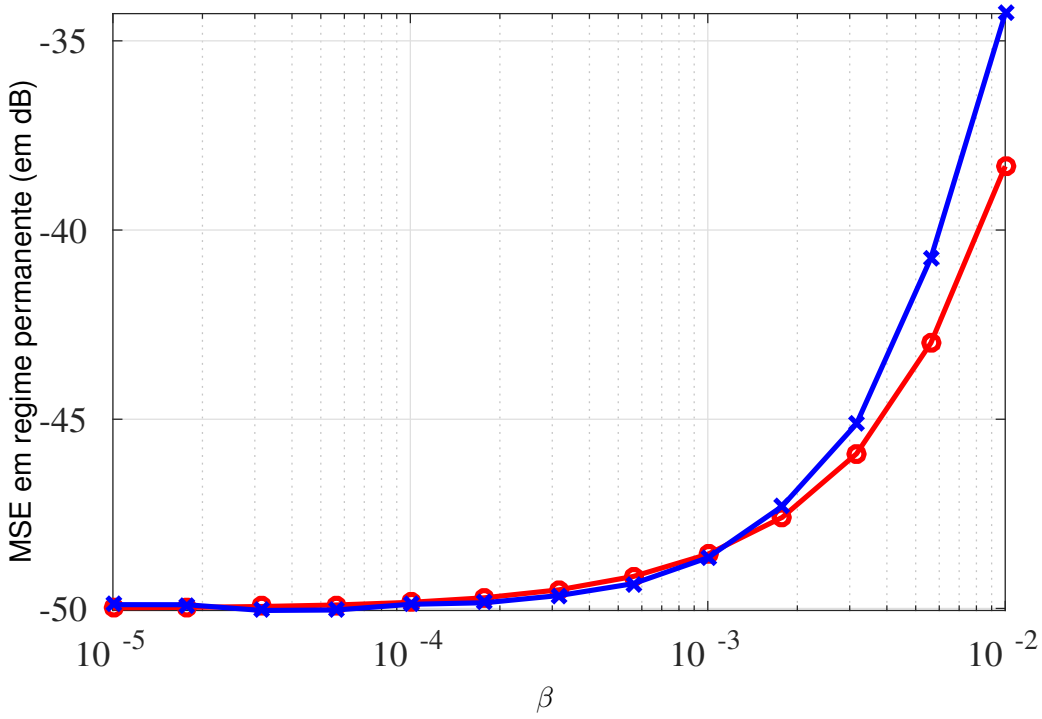


Figura 13 – Comparação do MSE assintótico entre a estimativa teórica (em vermelho) e o resultado de simulações (em azul), referente ao Cenário 4.

6.1.3 Cenário 3

O Cenário 3 é uma simulação cujo objetivo é aferir se o aumento da variância do ruído pode interferir negativamente na qualidade da estimativa teórica. Neste cenário, a variância do ruído é variada no intervalo $[10^{-6}, 10^{-3}]$, enquanto o fator de aprendizagem assume valores no conjunto $\{10^{-5}, 10^{-4}, 10^{-3}\}$. O filtro ótimo no Cenário 3 é descrito por

$$\boldsymbol{w}^{\star} = \begin{bmatrix} 0,5377 & 1,8339 & -2,2588 & 0,8622 & 0,3188 \end{bmatrix}^{T}, \tag{137}$$

com o MSE simulado em regime permanente sendo estimado com $K_1 = 50$ e $K_2 = 100$. A Tabela 3 apresenta o MSE teórico e o simulado, sem uma perda de aderência clara quando σ_{ν}^2 assume valores altos. Esse resultado pode ser explicado pelo fato de que o ruído aditivo adere perfeitamente a uma distribuição gaussiana, a qual é explicitamente considerada no modelo estocástico proposto.

$\log_{10}(\sigma_\nu^2)$	$\log_{10}(\beta)$	MSE simulado (dB)	MSE teórico (dB)
-6	-5	-59,9262	-59,9750
-6	-4	-59,7655	-59,7538
-6	-3	-58,0437	-57,8237
-5,75	-5	-57,6268	-57,4801
-5,75	-4	-57,3147	-57,3039
-5,75	-3	-55,8271	-55,7471
-5,5	-5	-55,0389	-54,9839
-5,5	-4	-55,0219	-54,8416
-5,5	-3	-53,7373	-53,5838
-5,25	-5	-52,520	-52,4873
-5,25	-4	-52,3489	-52,3742
-5,25	-3	-51,5629	-51,3594
-5	-5	-49,9860	-49,9901
-5	-4	-49,9514	-49,9014
-5	-3	-49,1469	-49,0882
-4,75	-5	-47,4327	-47,4923
-4,75	-4	-47,4254	-47,4233
-4,75	-3	-46,9576	-46,7770
-4,5	-5	-45,0881	-44,994
-4,5	-4	-44,9259	-44,9405
-4,5	-3	-44,5020	-44,4300
-4,25	-5	-42,4424	-42,4954
-4,25	-4	-42,4099	-42,4537
-4,25	-3	-42,0923	-42,0511
-4	-5	-39,9913	-39,9964
-4	-4	-39,8817	-39,9638
-4	-3	-39,5965	-39,6452
-3,75	-5	-37,6987	-37,4971
-3,75	-4	-37,3887	-37,4713
-3,75	-3	-37,4818	-37,2169
-3,5	-5	-35,0643	-34,9977
-3,5	-4	-35,1406	-34,9769
-3,5	-3	-34,7810	-34,7709
-3,25	-5	-32,4811	-32,4981
-3,25	-4	-32,5215	-32,4810
-3,25	-3	-32,4931	-32,3109
-3	-5	-30,0369	-29,9984
-3	-4	-30,0229	-29,9840
-3	-3	-29,9919	-29,8405

Tabela 3 – MSE simulado e teórico em regime permanente no Cenário 3.

6.1.4 Cenário 4

Na prática, há correlação entre os sinais, sendo assim a simulação realizada neste cenário considerou este contexto . Esta configuração pode prejudicar a qualidade do modelo teórico proposto, já que assumimos na análise que o sinal de entrada é branco. O Cenário 4 analisa essa configuração. O ruído aditivo é gaussiano com variância $\sigma_\nu^2 = 10^{-5}$, o filtro ótimo tem comprimento $N = 5$ e coeficientes descritos por

$$\boldsymbol{w}^\star = \begin{bmatrix} 0,5143 & 0,5366 & -0,6512 & 1,1274 & -0,8034 \end{bmatrix}^T. \tag{138}$$

O sinal de entrada é obtido pela filtragem de um sinal gaussiano de média zero e variância unitária pelo filtro de colorimento $H(z) = 1 - 0,8z^{-1}$. Neste Cenário, ao contrário dos pregressos, a variância do sinal de entrada não é mais unitária, já que a saída do filtro de colorimento apresenta uma variância superior à da entrada.

O resultado é apresentado na Figura 13, onde ainda se mantém uma boa aderência entre as curvas teórica e empírica. Entretanto, para valores de β elevados, observa-se uma discrepância maior do que a notada nos cenários pretéritos. Talvez tal discrepância não se deva ao fato de o sinal de entrada ser colorido, porque para filtros adaptativos menores (no Cenário 4 temos $N = 5$), a hipótese de independência estatística entre os termos $\|\boldsymbol{x}(k)\|_{\boldsymbol{\Sigma}}^2$ e $f^2[e(k)]$ (que dá origem a (124)) é muito forte. No próximo Cenário, aumentamos o comprimento do filtro para verificar se um filtro adaptativo com comprimento maior e sinal de entrada colorido também apresenta esta discrepância para valores altos de β.

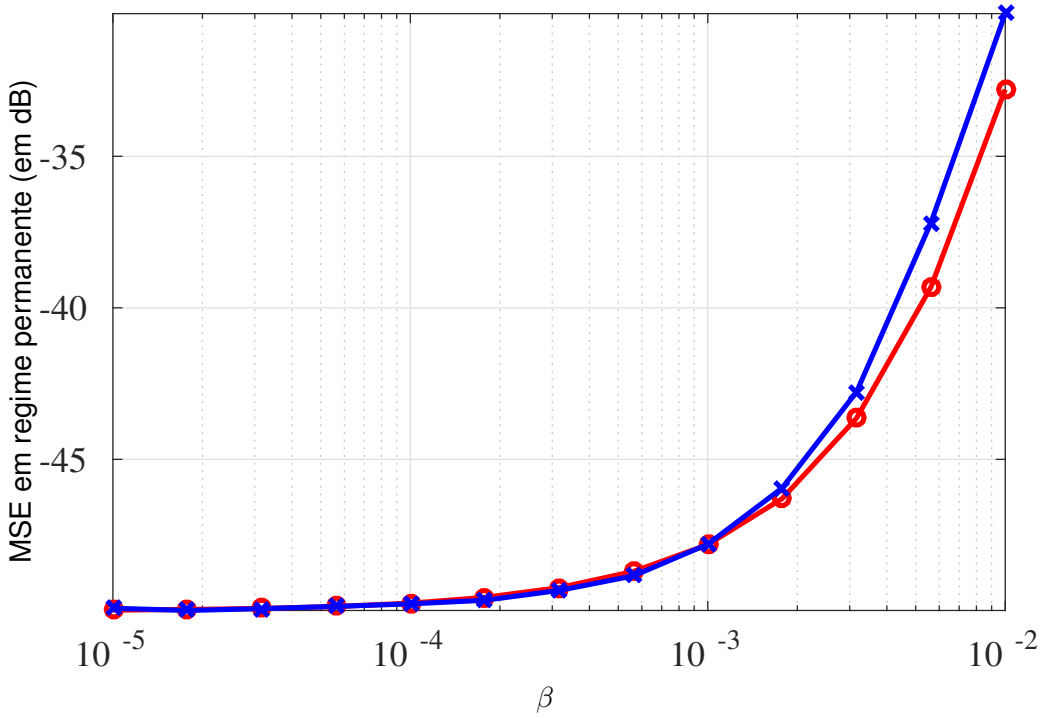


Figura 14 – Comparação do MSE assintótico entre a estimativa teórica (em vermelho) e o resultado de simulações (em azul), referente ao Cenário 5.

6.1.5 Cenário 5

O Cenário 5 é basicamente idêntico ao Cenário 4, exceto pelo fato de apresentar um vetor de coeficientes ótimo com mais coeficientes:

$$\boldsymbol{w}^{\star} = \begin{bmatrix} -0,4234 & 0,3616 & -0,3519 & 0,2695 & -2,5644 & 0,4659 & 1.8536 & 1.0393 \end{bmatrix}^{T}. \tag{139}$$

A Figura 14 apresenta a comparação entre a predição teórica proposta e o desempenho simulado do algoritmo. Com filtro de tamanho maior, observa-se uma menor discrepância entre a teoria e a simulação. Isto significa que o modelo proposto, embora tenha sido obtido assumindo-se que o sinal de entrada é branco, é capaz de fornecer estimativas acuradas (especialmente para valores baixos de β) mesmo quando o sinal de entrada é colorido.

6.1.6 Cenário 6

Nas simulações pretéritas, o ruído de medição presente no sinal de referência era branco e gaussiano. Embora tal hipótese seja comumente satisfeita na prática, a grande vantagem do algoritmo CMPN se dá na presença de ruído impulsivo. Infelizmente, o modelo teórico proposto é restrito a ruído de natureza gaussiana. O Cenário 6 pretende aferir se o modelo proposto apresenta resultados muito distantes da realidade quando o algoritmo opera sob ruído impulsivo, situação na qual o algoritmo apresenta grande robustez.

Neste cenário em particular, devido à natureza impulsiva do ruído, os resultados são bastante ruidosos, necessitando o aumento do número de ensaios independentede Monte Carlo. Assim, para este cenário temos $K_1 = K_2 = 500$.

O sinal de entrada é branco, gaussiano e de variância unitária. O fator de aprendizagem assume valores no conjunto $\beta \in \{10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}\}$. Os coeficientes do filtro ótimo são dados por

$$\boldsymbol{w}^\star = \begin{bmatrix} -0,4234 & 0,3616 & -0,3519 & 0,2695 & -2,5644 & 0,4659 & 1,8536 & 1,0393 \end{bmatrix}^T. \tag{140}$$

O ruído de medição $\nu(k)$ é dado por

$$\nu(k) = (1 - \omega(k))\varphi(k) + \omega(k)\phi(k), \tag{141}$$

onde $\omega(k)$ é um processo de Bernoulli especificado pela probabilidade Pr$[\omega(k) = 1]$, e $\varphi(k)$ e $\phi(k)$ são ruídos gaussianos brancos com médias nulas e variâncias $\sigma_\varphi^2 = 1$ e $\sigma_\phi^2 = 10^{-4}$, respectivamente. Quando $\omega(k) = 1$, temos um ruído de alta variância (já que $\sigma_\varphi^2 = 1$), o que deve ocorre com probabilidade Pr$[\omega(k) = 1]$, modelando a existência, em algumas amostras, de um ruído impulsivo. Quando Pr$[\omega(k) = 1] = 0$ inexiste ruído impulsivo, degenerando a configuração nos casos contemplados nos cenários anteriores. No modelo teórico, a variância do ruído (supostamente gaussiano) foi arbitrada de modo a igualar a variância do ruído impulsivo distribuído consoante (141).

A Tabela 4 apresenta os resultados para o cenário considerado, para diferentes probabilidades de ocorrência de ruído impulsivo. Observa-se que mesmo com uma maior

Pr$[\omega(k)=1]$	Resultado	$\beta=10^{-5}$	$\beta=10^{-4}$	$\beta=10^{-3}$	$\beta=10^{-2}$
0	Teórico	-39,994	-39,942	-39,439	-34,809
0	Empírico	-40,01	-39,934	-39,515	-34,49
0,01	Teórico	-19,955	-19,94	-19,793	-18,129
0,01	Empírico	-20,236	-20,232	-20,089	-19,713
0,02	Teórico	-16,966	-16,953	-16,816	-15,249
0,02	Empírico	-16,893	-16,912	-17,018	-16,882
0,03	Teórico	-15,213	-15,2	-15,067	-13,543
0,03	Empírico	-15,272	-15,285	-15,12	-15,116
0,04	Teórico	-13,967	-13,954	-13,824	-12,326
0,04	Empírico	-13,854	-13,949	-13,994	-13,865
0,05	Teórico	-13	-12,987	-12,859	-11,379
0,05	Empírico	-13,022	-13,074	-13,037	-12,812
0,06	Teórico	-12,21	-12,197	-12,07	-10,603
0,06	Empírico	-12,192	-12,179	-12,138	-12,038
0,07	Teórico	-11,541	-11,529	-11,403	-9,946
0,07	Empírico	-11,571	-11,45	-11,463	-11,411
0,08	Teórico	-10,962	-10,95	-10,825	-9,375
0,08	Empírico	-11,01	-11,041	-10,917	-10,85
0,09	Teórico	-10,451	-10,439	-10,315	-8,872
0,09	Empírico	-10,449	-10,524	-10,426	-10,446
0,1	Teórico	-9,994	-9,982	-9,858	-8,421
0,1	Empírico	-10,027	-10	-10,014	-9,912

Tabela 4 – MSE (em dB) em regime permanente do algoritmo CMPN num cenário com presença de ruído impulsivo,

presença de ruído impulsivo (10% das amostras corrompidas é um valor considerado alto na literatura), o modelo ainda prevê com bastante precisão o desempenho assintótico do algoritmo. Tal resultado confirma a robustez do algoritmo a ruído impulsivo. Cabe entretanto observar que o modelo apresenta discrepâncias maiores com relação aos resultados de simulação quando o fator de aprendizagem assume valores altos (isto é, $\beta = 10^{-2}$). Por conseguinte, poderíamos dizer que o modelo estocástico previsto consegue prever o desempenho do algoritmo CMPN sob condição de ruído impulsivo quando o fator de aprendizagem assume valores pequenos.

6.2- Comentários Finais

Neste capítulo, as predições do modelo estocástico detalhado no Capítulo 5 foram comparadas com resultados de simulação. Observa-se que variando-se o filtro ótimo (bem como seu tamanho) e a variância do ruído aditivo, o modelo é capaz de prever o desempenho assintótico do algoritmo em uma ampla faixa de valores de β. Mesmo quando o sinal de entrada é colorido (violando uma hipótese utilizada na obtenção do modelo estocástico), ainda assim a análise teórica angaria predições bastante acuradas. Por fim o modelo proposto também foi submetido a simulações em um cenário em que o sinal de entrada conta com presença de ruído impulsivo, prevendo precisamente o desempenho assintótico do algoritmo CMPN, porém na condição em que a variável β não seja configurada para valores de maior magnitude.

7- Conclusões

O algoritmo de norma ℓ_p continuamente misturada (CMPN) tem atraído grande atenção da comunidade científica, dada a elegância com que conjuga as normas ℓ_1 e ℓ_2 do erro na função custo. Sua equação de atualização é regida por uma não linearidade cuja complexidade explica a falta de resultados teóricos com relação ao desempenho do algoritmo, seja com relação ao seu transiente, seja com relação ao seu regime permanente.

Nesta dissertação, foi adotada uma metodologia inédita para a obtenção de uma estimativa teórica do desempenho assintótico do algoritmo, sob condições de ruído gaussiano. Tal contribuição deu origem ao artigo [49]. Ela se baseia na combinação de dois procedimentos. O primeiro deles consiste numa aproximação da função não linear do erro da equação de atualização do algoritmo CMPN por uma função linear por partes. Os parâmetros desta função linear por partes foram ajustados por meio da técnica de *Least Squares* ou mínimos quadrados. Tal aproximação é então conjugada com um método específico de análise teórica de algoritmos de filtragem adaptativa, que recorre a balanços do fluxo de energia ao longo do processo de aprendizagem do algoritmo. Para obtenção de resultados analíticos compatíveis ao desempenho médio do algoritmo, algumas hipóteses simplificadoras foram empregadas, entre as quais: hipótese de independência entre o sinal de entrada e os coeficientes adaptativos do filtro num determinado instante, independência estatística entre o sinal de ruído (supostamente gaussiano) e as demais variáveis aleatórias, bem como a hipótese de que o sinal de entrada é branco.

A metodologia dá origem a uma equação de ponto fixo, a qual fornece, após determinado número de iterações, uma estimativa teórica do desempenho assintótico do algoritmo. Tal estimativa teórica foi confrontada com resultados de simulações, tendo apresentado uma boa aderência com relação ao desempenho real do algoritmo.

O modelo proposto é capaz de prever o desempenho do algoritmo CMPN em cenários submetidos a ruído impulsivo quando o fator de aprendizagem β assume valores baixos. Decerto um interessante trabalho futuro seria encontrar um modelo estocástico para modelar o desempenho do algoritmo CMPN submetido a ruído impulsivo em cenários

com β elevado, já que diversas vantagens de algoritmos de filtragem adaptativa são aparentes quando β assume valores altos, para garantir uma alta taxa de convergência. Investigações nesta linha (utilizando outra metodologia de análise teórica) podem ser encontradas em [50; 51]. Outra possível direção para futuras investigações reside na adoção de aproximações que sejam válidas ao longo do transiente (o que permitiria uma análise de transiente), bem como a análise da capacidade de rastreamento do algoritmo, necessária quando a função de transferência a identificar é variante no tempo.

Referências Bibliográficas

[1] K. Vijetha and R. N. B., “Low power low area vlsi implementation of adaptive fir filter using da for decision feed back equalizer,” *Microprocessors and Microsystems*, vol. 93, p. 104577, 2022.

[2] H. D. P, S, *Adaptive Filtering. Algorithms and Practical Implementation*. Springer Science+Business Media, 3th ed., 2008.

[3] K. Ichige, M. Iwaki, and R. Ishii, “Accurate estimation of minimum filter length for optimum fir digital filters,” *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 47, no. 10, pp. 1008–1016, 2000.

[4] T. Parks and J. McClellan, “Chebyshev approximation for nonrecursive digital filters with linear phase,” *IEEE Transactions on Circuit Theory*, vol. 19, no. 2, pp. 189–194, 1972.

[5] L. Rabiner and O. Herrmann, “On the design of optimum fir low-pass filters with even impulse response duration,” *IEEE Transactions on Audio and Electroacoustics*, vol. 21, no. 4, pp. 329–336, 1973.

[6] W. B. Davenport and W. L. Root, *An introduction to the theory of random signals and noise*. New York: IEEE Press, 1987.

[7] S. Pralgauskaite, V. Palenskis, J. Matukas, D. Seliuta, I. Kašalynas, and G. Valušis, “White noise peculiarities in diode structures,” in *2013 22nd International Conference on Noise and Fluctuations (ICNF)*, pp. 1–4, 2013.

[8] L. Shi and Y. Lin, “Convex combination of adaptive filters under the maximum correntropy criterion in impulsive interference,” *IEEE Signal Processing Letters*, vol. 21, no. 11, pp. 1385–1388, 2014.

[9] O. M. Abdelrhman, Y. Dou, and S. Li, “Performance analysis of the standard constrained maximum versoria criterion based adaptive algorithm,” *IEEE Signal Processing Letters*, vol. 30, pp. 125–129, 2023.

[10] H. Zayyani, “Continuous mixed p-norm adaptive algorithm for system identification,” IEEE Signal Processing Letters, Vol 21,2014.

[11] H. Nascimento, V and A. Sayed, “On the learning mechanism of adaptive filters,” *IEEE Transactions on Information Theory*, vol. 48, no. 6, 2000.

[12] S. Haykin, *Adaptive Filter Theory*. Pearson Education, 5th ed., 2014.

[13] M. Rupp, “A family of adaptive filter algorithms with decorrelating properties,” *IEEE Transactions on Signal Processing*, vol. 46, no. 3, pp. 771–775, 1998.

[14] B. Usevitch and M. Orchard, “Adaptive filtering using filter banks,” *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 43, no. 3, pp. 255–265, 1996.

[15] P. Lara, F. Igreja, L. D. T. J. Tarrataca, D. B. Haddad, and M. R. Petraglia, “Exact expectation evaluation and design of variable step-size adaptive algorithms,” *IEEE Signal Processing Letters*, vol. 26, no. 1, pp. 74–78, 2019.

[16] S. S. B. Widrow, *Adaptive Signal Processing*. Pearson Education, 5th ed., 1985.

[17] B. B, F, *Adaptive Filters, Theory and Applications*. John Wiley Sons, 1998.

[18] S. A, H, *Adaptive Filters Processing*. Wiley Interscience, 2008.

[19] G. Y. Chen, Y and A. Hero III, “Sparse lms for system identification,” *Acoustics, Speech, And Signal Processing*, 1988.

[20] K. Mayyas and T. Aboulnasr, “Leaky lms algorithm: Mse analysis for gaussian data,” *IEEE TRANSACTIONS ON SIGNAL PROCESSING*, vol. 45, no. 4, 1997.

[21] D. T. Pedro Lara and D. Haddad, “Exact expectation analysis of the deficient-length lms algorithm,” *Signal Processing*, pp. 54–64, 2019.

[22] T. T. P. Silva, F. Igreja, P. Lara, L. Tarrataca, A. Kar, and D. B. Haddad, “On the skewness of the lms adaptive weights,” *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 68, no. 8, pp. 3022–3026, 2021.

[23] T. Al-Naffouri and A. Sayed, “Transient analysis of adaptive filters with error nonlinearities,” *IEEE Transactions on Signal Processing*, vol. 51, 3 2003.

[24] T. T. P. Silva, P. Lara, F. Igreja, F. D. V. R. Oliveira, L. Tarrataca, and D. B. Haddad, “An exact expectation model for the lms tracking abilities,” *IEEE Transactions on Signal Processing*, vol. 68, pp. 5882–5893, 2020.

[25] S. Koike, “Adaptive threshold nonlinear algorithm for adaptive filters with robustness against impulse noise,” *IEEE Transactions on Signal Processing*, vol. 45, no. 9, pp. 2391–2395, 1997.

[26] O. T. J.A, Chambers and C. A.G, “Least mean mixed-norm adaptive filtering,” 1994.

[27] C. J.A and A. A, “A robust mixed-norm adaptive filter algorithm,” 1997.

[28] W. Ma, J. Duan, W. Man, J. Liang, and B. Chen, “General mixed-norm-based diffusion adaptive filtering algorithm for distributed estimation over network,” *IEEE Access*, vol. 5, pp. 1090–1102, 2017.

[29] M. O. Sayin, N. Denizcan Vanli, and S. S. Kozat, “Improved convergence performance of adaptive algorithms through logarithmic cost,” in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4513–4517, 2014.

[30] Y. Zou, S.-C. Chan, and T.-S. Ng, “Least mean m-estimate algorithms for robust adaptive filtering in impulse noise,” *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 47, no. 12, pp. 1564–1569, 2000.

[31] I. Song, P. Park, and R. W. Newcomb, “A normalized least mean squares algorithm with a step-size scaler against impulsive measurement noise,” *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 60, no. 7, pp. 442–445, 2013.

[32] B. Chen, L. Xing, H. Zhao, N. Zheng, and J. C. Prı´ncipe, “Generalized correntropy for robust adaptive filtering,” *IEEE Transactions on Signal Processing*, vol. 64, no. 13, pp. 3376–3387, 2016.

[33] L. R. Vega, H. Rey, J. Benesty, and S. Tressens, “A stochastic model for a new robust nlms algorithm,” in *2007 15th European Signal Processing Conference*, pp. 375–379, 2007.

[34] S. Guan, Z. Li, and H. Zhang, “Continuous mixed p-norm adaptive algorithm with reweighted l0-norm constraint,” in *2017 First International Conference on Electronics Instrumentation Information Systems (EIIS)*, pp. 1–5, 2017.

[35] L. Shi, H. Zhao, and Y. Zakharov, “Generalized variable step size continuous mixed p -norm adaptive filtering algorithm,” *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 66, no. 6, pp. 1078–1082, 2019.

[36] Y. Zhu, H. Zhao, X. Zeng, and B. Chen, “Robust generalized maximum correntropy criterion algorithms for active noise control,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 1282–1292, 2020.

[37] V. Patel, S. S. Bhattacharjee, and M. G. Christensen, “Generalized soft-root-sign based robust sparsity-aware adaptive filters,” *IEEE Signal Processing Letters*, vol. 30, pp. 200–204, 2023.

[38] M. Lee, S. H. Ryu, C. Park, and P. Park, “Combined step-size generalized continuous mixed p-norm algorithm,” in *2023 Third International Symposium on Instrumentation, Control, Artificial Intelligence, and Robotics (ICA-SYMP)*, pp. 5–8, 2023.

[39] X. Huang, Y. Li, X. Han, W. Shi, and H. Tu, “Affine-projection q-rényi algorithm for channel estimation under different vehicle velocities and impulsive interference,” *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 70, no. 3, pp. 1234–1238, 2023.

[40] J. E. Mazo, “On the independence theory of equalizer convergence,” *The Bell System Technical Journal*, vol. 58, no. 5, pp. 963–993, 1979.

[41] N. N. Siqueira, T. P. Ferreira, L. C. Resende, R. M. S. Pimenta, D. B. Haddad, and M. R. Petraglia, “Transient analysis of the set-membership lms algorithm,” *IEEE Communications Letters*, vol. 25, no. 4, pp. 1298–1302, 2021.

[42] N. Yousef and A. Sayed, “A unified approach to the steady-state and tracking analyses of adaptive filters,” *IEEE Transactions on Signal Processing*, vol. 49, no. 2, pp. 314–324, 2001.

[43] T. Al-Naffouri and A. Sayed, “Transient analysis of data-normalized adaptive filters,” *IEEE Transactions on Signal Processing*, vol. 51, no. 3, pp. 639–652, 2003.

[44] A. Papoulis, *Probability, random variables, and stochastic processes*. McGraw-Hill, 1991.

[45] N. J. Bershad, E. Eweda, and J. C. Bermudez, “Stochastic analysis of soft limiters in the lms algorithm for stationary white gaussian inputs—a unified theory,” *Signal Processing*, vol. 142, pp. 27–35, 2018.

[46] N. J. Bershad and J. C. Bermudez, “Mean-square stability of the normalized least-mean fourth algorithm for white gaussian inputs,” *Digital Signal Processing*, vol. 21, no. 6, pp. 694–700, 2011.

[47] K. Shi and P. Shi, “Convergence analysis of sparse LMS algorithms with l1-norm penalty based on white input signal,” *Signal Processing*, vol. 90, pp. 3289–3293, Dec. 2010.

[48] G. Su, J. Jin, Y. Gu, and J. Wang, “Performance analysis of l_0 norm constraint least mean square algorithm,” *IEEE Transactions on Signal Processing*, vol. 60, no. 5, pp. 2223–2235, 2012.

[49] E. S. D. Oliveira and D. B. Haddad, “Análise do algoritmo CMPN em regime permanente,” in *Proceeding Series of the Brazilian Society of Computational and Applied Mathematics*, SBMAC, Dec. 2022.

[50] P. Guo, Y. Yu, T. Yang, H. He, and R. C. de Lamare, “Robust nlms algorithms with combined step-size against impulsive noises,” *Digital Signal Processing*, vol. 128, p. 103609, 2022.

[51] Y. Yu, T. Yang, H. Chen, R. C. de Lamare, and Y. Li, “Sparsity-aware ssaf algorithm with individual weighting factors: Performance analysis and improvements in acoustic echo cancellation,” *Signal Processing*, vol. 178, p. 107806, 2021.