Universidade Federal do Maranhão - UFMA
Coordenação de Pós-Graduação em Engenharia Elétrica

Dissertação de Mestrado

ALGORITMO RECURSIVO BASEADO EM UMA FUNÇÃO NÃO LINEAR DO ERRO

CRISTIANE CRISTINA SOUSA DA SILVA

São Luis - MA, Brasil

6 de março de 2009

Sumário

ALGORITMO RECURSIVO BASEADO EM UMA FUNÇÃO NÃO LINEAR DO ERRO

Dissertação de mestrado submetida à coordenação do Curso de Pós-Graduação de Engenharia de Eletricidade da UFMA como parte dos requisitos para obtenção do título de Mestre em Engenharia de Eletricidade na área de Automação e Controle.

CRISTIANE CRISTINA SOUSA DA SILVA

FEVEREIRO, 2009

Silva, Cristiane Cristina Sousa da
Algoritmo recursivo baseado em uma função não linear do erro / Cristiane Cristina Sousa da Silva. – São Luís, 2009.

47f.

Orientador: Allan Kardec Duailibe Barros Filho.
Impresso por computador (Fotocópia).
Dissertação (Mestrado) – Universidade Federal do Maranhão, Curso de Pós-Graduação em Engenharia Elétrica. São Luís, 2009.

1. Sinais – Processamento. 2. Filtragem adaptativa. 3. Algoritmos adaptativos. I. Barros Filho, Allan Kardec Duailibe, orient. II. Título.

CDU 004.8

ALGORITMO RECURSIVO BASEADO EM UMA FUNÇÃO NÃO LINEAR DO ERRO

Cristiane Cristina Sousa da Silva

Dissertação aprovada em 13 de fevereiro de 2009.

Prof. Allan Kardec Duailibe Barros Filho, Ph. D.
(Orientador)

Prof. João Viana da Fonseca Neto, Dr.
(Co-Orientador)

Prof. João Marcos Travassos Romano, Dr.
(Membro da Banca Examinadora)

Prof. Aristófanes Corrêa Silva, Dr.
(Membro da Banca Examinadora)

ALGORITMO RECURSIVO BASEADO EM UMA FUNÇÃO NÃO LINEAR DO ERRO

MESTRADO

Área de Concentração: AUTOMAÇÃO E CONTROLE

CRISTIANE CRISTINA SOUSA DA SILVA

Orientador: Prof. Dr. Allan Kardec Duailibe Barros Filho

Curso de Pós-Graduação
em Engenharia de Eletricidade da
Univesidade Federal do Maranhão

AGRADECIMENTOS

Ao professor Allan Kardec Duailibe Barros Filho pela motivação, apoio, carinho e dedicação, fundamentais para a orientação deste trabalho. Pela oportunidade de crescimento e apredizado. Gostaria de ratificar a sua competência, participação com discussões, correções e sugestões que fizeram com que concluíssemos este trabalho.

Ao professor João Viana Fonseca Neto, Co-Orientador deste trabalho, pelo incentivo, orientação, e crédito durante todas as fases do curso de mestrado.

Ao professor Marcos Antonio F. de Araújo, Co-Orientador deste trabalho, pela atenção, carinho, amizade e dedicação a mim dispensados em todas as fases do curso de mestrado.

Ao professor Aristófanes Corrêa Silva pelo crédito, incentivo e amizade.

A todos os meus amigos do PIB pelo companherismo e amizades sinceras.

A Deus pelas oportunidades que me foram dadas.

À toda a minha família pelo carinho, apoio e compreensão ao longo do ano em que o presente trabalho foi desenvolvido. Em especial ao meu marido Marcio John Moreira e meu filho Gustavo Henrique Sousa S. Moreira.

À CAPES pela bolsa a mim concedida.

Resumo

Muitos dos filtros adaptativos são baseados no método do Erro quadrático médio (*Mean Square Error - MSE*). O desenvolvimento desses filtros nos garante recuperar apenas informações de segunda ordem dos sinais a serem filtrados, ou seja, só consegue recuperar totalmente informações de sinais Gaussianos. No entanto, os sinais naturais ou artificiais não são necessariamente gaussianos. Desta forma, a utilização de estatística de alta ordem, como uma forma de extrair mais informações dos sinais, tem se demonstrado de grande valia em sistemas adaptativos [7][8][9].

Neste trabalho, nós apresentamos o desenvolvimento de um algoritmo adaptativo baseado em funções não lineares inspirado na dedução do algoritmo *Recursive Lest Square (RLS)* [1]. Tal desenvolvimento baseia-se na utilização de estatísticas de alta ordem para a obtenção de mais informações dos sinais envolvidos no processo, com o objetivo de melhorar a performance de um filtro adaptativo. Chamaremos esse novo algoritmo de Recursivo não Linear - RNL.

Deduzimos equações, baseadas em uma função não linear, para a obtenção de critérios que garantam a convergência. Também fazemos um estudo da covariância do vetor peso em regime estacionário e determinamos equações que calculem o desajuste e o tempo de aprendizagem do processo adaptativo do algoritmo RNL.

Apresentamos o algoritmo não linear recursivo, que utiliza como critério a função $\varepsilon_n = \sum_{j=1}^{M}\sum_{i=1}^{n}\left\{\lambda^{n-i}\left[e_i\right]^{2j}\right\}$, sendo M e n inteiros positivos. Foram feitas simulações com este algoritmo para validar a teoria apresentada e estudamos o comportamento da convergência do algoritmo RNL. O resultado mostrou que o algoritmo RNL possui uma rápida convergência para o mesmo desajuste quando comparado com o algoritmo *RLS*. .

Palavras-chaves: Processamento de sinais. Filtragem Adaptativa. Algoritmos Adaptativos

Capítulo 1

Introdução

Os sinais envolvidos em processamentos de filtragem, predição e estimação são sinais aleatórios, os quais são caracterizados por suas propriedades estatísticas. O projeto de filtro para o processamento de sinais aleatórios requer o conhecimento prévio de algumas informações sobre as propriedades estatísticas dos sinais envolvidos. Quando isso é possível, trata-se o problema no âmbito do processamento estatístico de sinais. Nos casos em que tais informações são desconhecidas e não podem ser estimadas em tempo real, a melhor solução é o emprego de filtros adaptativos.

A filtragem adaptativa constitui uma ferramenta fundamental no processamento de sinais digitais. Ela é aplicada atualmente em um grande número de problemas de engenharia. Esta técnica tem sido explorada com sucesso em problemas de economia, engenharia biomédica, equalização de canais, sistemas de controle e em telecomunicações. Em especial, na área biomédica, diversas aplicações podem ser encontradas, tais como: cancelamento de interferências do coração doador no eletrocardiograma (ECG) durante o transplante de coração; cancelamento da influencia materna em ECG fetal.

O trabalho em filtragem adaptativa envolve o estudo de algoritmos e de estruturas de filtragem de forma a melhorar o desempenho dos sistemas adaptativos existentes. Entre os diversos algoritmos existentes na literatura, pode-se citar o *Recursive Least Square (RLS)*, nessa técnica, o estimador médio é atualizado com

base em um conjunto de valores previamente simulados em vez de ser atualizado com um único valor.

Um sistema adaptativo é aquele cuja estrutura é alterável (através do ajuste dos seus coeficientes) de tal forma que seu comportamento melhore de acordo com algum critério de desempenho através da exposição ao ambiente no qual será inserido. O ajuste dos coeficientes do filtro adaptativo é realizado através da implementação de um algoritmo, devidamente escolhido, cujo objetivo é atender a requisitos dos sistemas. Estes algoritmos são definidos como algoritmos adaptativos.

Neste trabalho, nós apresentamos o desenvolvimento de um algoritmo adaptativo que utiliza como função de custo uma não linearidade par. Chamaremos esse novo algoritmo de Recursivo não linear.

1.1 Motivação

Muitos dos filtros desenvolvidos, em filtragem adaptativa, são baseados no método do Erro quadrático médio (*Mean Square Error - MSE*) conseguindo, deste modo, recuperar apenas informações de segunda ordem dos sinais a serem filtrados, ou seja, só conseguem recuperar totalmente informações de sinais gaussianos.

No entanto, os sinais naturais (biomédicos, geoprocessamento) ou artificiais (FM, AM, telecomunicação em geral) não são necessariamente gaussianos, longe disso. Assim, objetivando extrair mais informações dos sinais envolvidos no processo de adaptação, propomos o desenvolvimento de um novo algoritmo adaptativo baseado em funções não lineares inspirado no algoritmo *RLS* padrão.

1.2 Organização do texto

Este trabalho está organizado da seguinte forma: No capítulo 2, apresentamos o combinador linear adaptativo, fazemos uma revisão da superfície quadrática e um breve comentário sobre algoritmos de gradiente estocástico e de mínimos quadrados.

No capítulo 3, revisamos o algoritmo *RLS* mostrando a sua dedução, convergência do vetor peso, tempo de aprendizagem e desajuste final.

No capítulo 4, desenvolvemos um novo algoritmo, que utiliza como função de

custo $\varepsilon_n = \sum_{j=1}^{M} \sum_{i=1}^{n} \left\{ \lambda^{n-i} \left[e_i \right]^{2j} \right\}$, sendo M e n inteiros positivos. Obtemos expressão para garantir a convergência e determinamos o desajuste;

No capítulo 5, simulações computacionadas foram realizadas para comparar o desempenho do algoritmo proposto com o *RLS* padrão, aplicando as equações desenvolvidas no capítulo 4 . Apresentamos também as discussões do trabalho.

No capítulo 6, são apresentadas as conclusões e sobre o algoritmo desenvolvido, assim como algumas das possibilidades futuras de expansão.

Capítulo 2

Filtragem Adaptativa

2.1 Introdução

Os filtros Adaptativos representam uma parte significativa no processamento de sinais digitais. Historicamente, a abordagem paramétrica de sinais tem sido explorada com sucesso em problemas de comunicações, controle robótica, radar, sismologia e engenharia biomédica. Os chamados problemas de filtragem podem ser identificados e caracterizados mais especificamente pelos termos de filtragem, suavização, predição [1].

O principal objetivo da filtragem de sinais é melhorar a qualidade do sinal de acordo com um critério de desempenho. Os sinais podem ser considerados tanto no domínio do tempo com no domínio da freqüência. A diferença dos filtros adaptativos aos demais é o seu desempenho auto-ajustável e variante no tempo.

Muitos algoritmos adaptativos utilizam-se do erro quadrático médio como função de custo que deseja-se minimizar. O erro quadrático médio é uma função convexa dos componentes do vetor peso e gera uma superfície hiperparabolóide que garante a existência de um mínimo global. O problema consiste em determinar procedimentos de tal forma a encontrar esse mínimo, o mais rápido possível e com o menor erro final.

2.2 O Combinador Linear Adaptativo

A estrutura mais usada na implementação de filtros adaptativos é o combinador linear adaptativo (CLA), mostrado na figura (2.1). Pode-se observar que o filtro adaptativo possui uma entrada única, $\mathbf{u}_i$ (no tempo i), definida como

$$\mathbf{u}_i = \begin{bmatrix} u_i, & u_{i-1}, & ..., & u_{i-(L-1)} \end{bmatrix}^T \tag{2.1}$$

sendo L o tamanho do filtro.

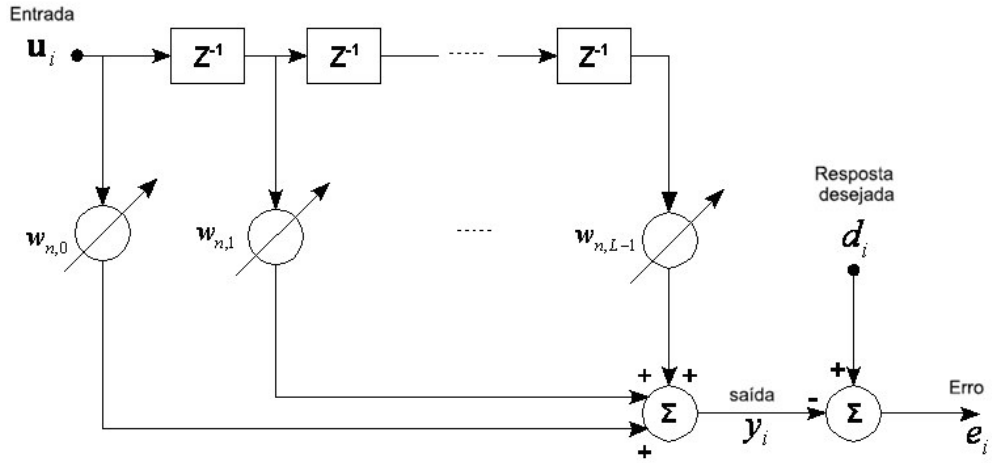

Figura 2.1: Combinador Linear Adaptativo - forma transversal

$\mathbf{w}_n$, é o vetor peso no tempo n, definido por

$$\mathbf{w}_n = \begin{bmatrix} w_{n0} & w_{n1} & \dots & w_{n(L-1)} \end{bmatrix}^T. \tag{2.2}$$

e a saída, y_i, é igual ao produto interno de $\mathbf{u}_i$ por $\mathbf{w}_n$:

$$y_i = \mathbf{u}_i^T \mathbf{w}_n = \mathbf{w}_n^T \mathbf{u}_i \tag{2.3}$$

Conforme visto na figura (2.1), o sinal de erro, no instante i, é dado por

$$e_i = d_i - y_i. \tag{2.4}$$

Substituindo (2.3) nesta expressão, temos:

$$e_i = d_i - \mathbf{u}_i^T \mathbf{w}_n = d_i - \mathbf{w}_n^T \mathbf{u}_i. \tag{2.5}$$

Existem vários algoritmos e abordagens que podem ser utilizados, dependendo dos requisitos do problema. No entanto, existem duas abordagens principais para o desenvolvimento de algoritmos de filtros adaptativos [1]. Discutiremos essas duas abordagens nas próximas seções.

2.3 Algoritmos de Gradiente Estocástico

O erro quadrático médio, como já foi dito, é uma função convexa dos componentes do vetor peso e gera uma superfície hiperparabolóide que garante a existência de um mínimo global, o que representa a solução ótima de Wiener. Esta solução pode ser encontrada pelo bem conhecido método de otimização, denominado "método de decida mais íngreme", que utiliza o vetor gradiente para descer gradualmente passo a passo para o mínimo da função erro. As chamadas equações de Wiener-Hopf, em forma matricial, definem esta solução ótima de Wiener. Vamos, agora determinar a solução ótima a partir do critério do erro quadrático médio dado por [9].

$$\xi = E[e_i^2] = E[(d_i - \mathbf{w}_n^T \mathbf{u}_i)^2] \tag{2.6}$$

Considerando um ambiente estácionario, podemos desenvolver a Equação 2.6 como

$$\xi = E[d_i] + \mathbf{w}^T \varphi \mathbf{w} - 2\mathbf{z}^T \mathbf{w} \tag{2.7}$$

sendo φ a matriz de auto-correlação do sinal de entrada $\mathbf{u_i}$ e $\mathbf{z}$ a matriz de correlação cruzada do sinal de entrada $\mathbf{u_i}$ com o sinal desejado d_i.

Pode-se observar que o erro quadrático médio é uma função quadrática dos pesos, cujo gráfico é uma superfície côncava hiperparabolóica, conforme vemos na figura 2.2, onde consideramos apenas dois pesos. Esta função, obviamente, nunca pode ser negativa.

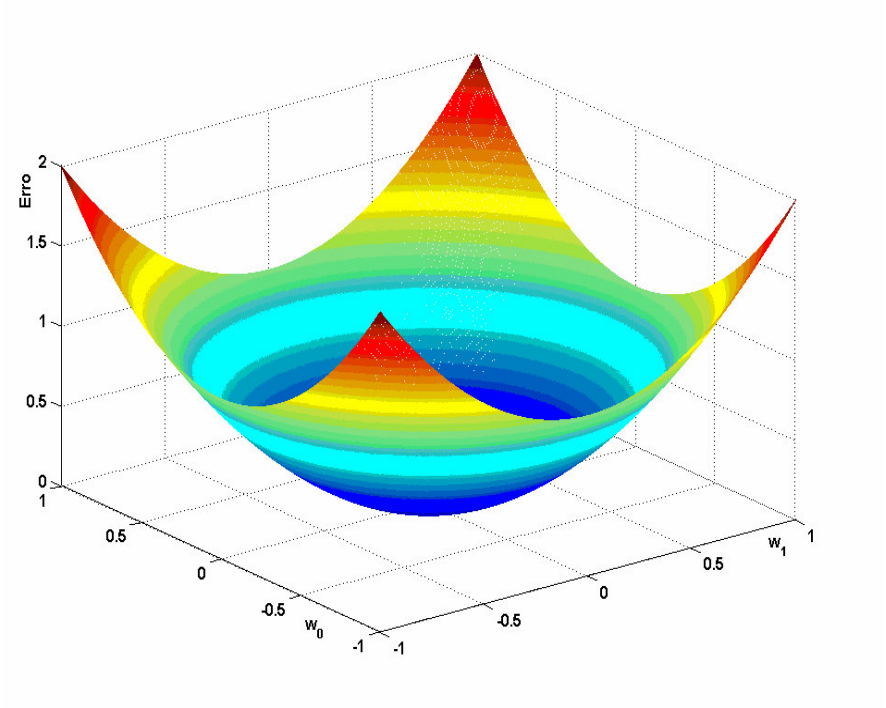

Figura 2.2: Porção de uma superfície quadrática tridimensional, juntamente com alguns contornos. O erro quadrático médio está plotado na vertical, w_0 e w_1 variam de -1 a 1

O gradiente da superfície de desempenho do erro quadrático médio, designado por ∇, pode ser obtido derivando (3.2) para obter o erro quadrático médio mínimo. O vetor peso é ajustado para seu valor ótimo, $\mathbf{w}_*$, onde o gradiente é zero, ou seja:

$$\mathbf{w}_* = \varphi^{-1}\mathbf{z} \tag{2.8}$$

Esta equação é uma expressão matricial da equação de Wiener-Hopf.

Um sistema modificado das equações de Wiener-Hopf, usado para adaptar os pesos do filtro em direção ao mínimo é o algoritmo chamado "*Least Mean Squares (LMS)*", inventado por B. Widrow e M.E.Hoff Jr em 1959. Este algoritmo calcula o gradiente da função de erro de valores instantâneos da matriz de correlação das entradas e do vetor de correlação cruzada entre as entradas e a resposta desejada. O algoritmo *LMS* é muito simples. Nesta seção, faremos uma breve discussão deste algoritmo.

Para desenvolver o algoritmo *LMS*, usamos o própio e_i^2 como uma estimativa de ξ_i. Então, a cada iteração, no processo adaptativo, nós teremos uma estimação do

gradiente da forma

$$\hat{\nabla}_n = \begin{bmatrix} \frac{\partial e_n^2}{\partial w_0} \\ \vdots \\ \frac{\partial e_n^2}{\partial w_L} \end{bmatrix} = 2e_n \begin{bmatrix} \frac{\partial e_n}{\partial w_0} \\ \vdots \\ \frac{\partial e_n}{\partial w_L} \end{bmatrix} = -2e_n \mathbf{U}_n. \tag{2.9}$$

As derivadas de e_i, em relação aos pesos, seguem, diretamente de 2.4.

A partir de 2.9 temos o algoritmo *LMS* [9]

$$\mathbf{W}_n = \mathbf{W}_{n-1} + \mu e_n \mathbf{U}_n. \tag{2.10}$$

sendo μ o passo de adaptação. Tal parâmetro é uma constante que comanda a velocidade de convergência do algoritmo.

2.4 Algoritmos de Mínimos Quadrados

É sabido que na dedução do algoritmo *LMS* o objetivo é minimizar o quadrado médio da estimação erro. No método dos mínimos quadrados, por outro lado, no instante $n > 0$, os pesos do filtro adaptativo são calculados tal que a quantidade

$$\zeta_n = \sum_{i=1}^{n} \rho_i . \left| e_i \right|^2 \tag{2.11}$$

é minimizado, daí o nome mínimos quadrados

O algoritmo *Recursive Least Squares - RLS*, pode ser visto como um caso especial do filtro de *Kalman* [1], que é uma forma dos mínimos quadrados. Tais algoritmos têm como vantagem a baixa sensibilidade à natureza do sinal de entrada e uma maior velocidade de convergência quando comparado com os algoritmos de gradiente estocástico. O algoritmo mais popular desta família é o *RLS*. No próximo capítulo faremos a dedução deste algoritmo.

2.5 Conclusão do Capítulo

Neste capítulo, realizou-se uma revisão da superfície quadrática gerada quando se utiliza o erro quadrático médio (EQM) como critério aplicado sobre o erro em um filtro adaptativo. Mostramos a derivação do algoritmo *LMS*.

Capítulo 3

O algoritmo Mínimos Quadrados Recursivo (*Recursive Least Square-RLS*)

3.1 Introdução

Em implementações recursivas do método dos mínimos quadrados começamos com condições iniciais conhecidas e utilizamos a informação contida em novas amostras de dados para atualizar as estimativas passadas. Assim, encontramos que o tamanho do dado observável é variável. Desta forma expressamos a função de custo para ser minimizada como ε_n sendo n o tamanho variável do dado observável. Portanto é comum introduzir um fator peso na definição da função de custo.

3.2 Dedução do algoritmo *RLS*

No algoritmo *RLS* o fator ponderação ρ_i é escolhido como sendo

$$\rho_i = \lambda^{n-i} \tag{3.1}$$

sendo $0 << \lambda < 1$ uma constante positiva a ser escolhida.

O método dos mínimos quadrados padrão visto anteriormente coresponde ao caso em que $\lambda = 1$. O parâmetro λ é denominado fator de equecimento. Claramente quando $\lambda < 1$, os fatores de ponderação definidos 3.1 dá um peso maior às amostras

recentes das estimativas do erro (e assim, à amostras recentes do dado observado) comparado com as amostras mais antigas. Em outras palavras, a escolha de $\lambda < 1$ resulta em um esquema que dá mais ênfase às amostras recentes do dado observado e tende a esquecer as amostras antigas.

Substituindo 3.1 em 2.11 obtem-se a função de custo a ser mínimizada na dedução do algoritmo *RLS*.

$$\varepsilon_n = \sum_{i=1}^{n} \lambda^{n-i}.\,|e_i|^2 \tag{3.2}$$

Supondo que φ seja uma matriz não singular, o valor ótimo do vetor peso, $\hat{\mathbf{w}}_n$, para que a função de custo atinja este valor mínimo é definido pela equação normal escrita em forma matricial

$$\hat{\mathbf{w}}_n = \varphi_n^{-1}\mathbf{z}_n \tag{3.3}$$

Sendo

$$\varphi_n = \sum_{i=1}^{n} \lambda^{n-i}.\mathbf{u}_i\mathbf{u}_i^T \tag{3.4}$$

a matriz de auto-correlação do sinal de entrada $\mathbf{u_i}$ e

$$\mathbf{z}_n = \sum_{i=1}^{n} \lambda^{n-i}.\mathbf{u}_i d_i \tag{3.5}$$

a matriz de correlação cruzada do sinal de entrada $\mathbf{u_i}$ com o sinal desejado d_i.

Expandindo a Equação 3.4 e isolando o termo $i = n$, temos

$$\begin{aligned} \varphi_n &= \lambda\left[\sum_{i=1}^{n-1} \lambda^{n-1-i}.\mathbf{u}_i.\mathbf{u}_i^T\right] + \mathbf{u}_n.\mathbf{u}_n^T \\ \varphi_n &= \lambda\left[\varphi_{n-1}\right] + \mathbf{u}_n.\mathbf{u}_n^T \end{aligned} \tag{3.6}$$

Analogamente podemos escrever a Equação 3.5 da seguinte forma

$$\mathbf{z}_n = \lambda\left[\mathbf{z}_{n-1}\right] + \mathbf{u}_n.d_n \tag{3.7}$$

3.2.1 O lema de inversão de matrizes

Lema 1: Sejam $\mathbf{A}$ e $\mathbf{B}$ matrizes definidas positivas de dimensão $L \times L$, definimos

$$\mathbf{A} = \mathbf{B}^{-1} + \mathbf{C}\mathbf{D}^{-1}\mathbf{C}^T \tag{3.8}$$

De 3.8 obtemos

$$\mathbf{A}^{-1} = \mathbf{B} - \mathbf{B}\mathbf{C}\left(\mathbf{D} + \mathbf{C}^T\mathbf{B}\mathbf{C}\right)^{-1}\mathbf{C}^T\mathbf{B} \tag{3.9}$$

Sendo $\mathbf{D}$ matriz definida positivamente de dimensão $N \times L$ e $\mathbf{C}$ uma matriz $L \times N$.

A demonstração desse Lema é estabelecida pela multiplicação da Equação 3.8 por 3.9. Na próxima seção mostraremos como o lema de inversão de matrizes pode ser aplicado para obter uma equação recursiva.

3.2.2 O algoritmo *RLS* ponderado exponencialmente

Como a matriz de auto-correlação é positivamente definida e não singular, podemos aplicar o lema de inversão de matrizes para equação recursiva 3.4. Primeiramente faremos as seguintes identificações:

$$\begin{cases} \mathbf{A} = \varphi_n \\ \mathbf{B}^{-1} = \lambda\varphi_{n-1} \rightarrow \mathbf{B} = \lambda^{-1}\varphi_{n-1}^{-1} \\ \mathbf{C} = \mathbf{u}_n \\ \mathbf{D} = \mathbf{I} \end{cases} \tag{3.10}$$

Substituindo as definições 3.10 na equação 3.9, obtemos a relação de recursividade da matriz φ_n , ou seja:

$$\varphi_n^{-1} = \lambda^{-1}\varphi_{n-1}^{-1} - \frac{\lambda^{-2}\varphi_{n-1}^{-1}\mathbf{u}_n\mathbf{u}_n^T\varphi_{n-1}^{-1}}{1 + \lambda^{-1}\mathbf{u}_n^T\varphi_{n-1}^{-1}\mathbf{u}_n} \tag{3.11}$$

Por conveniência computacional podemos escrever as seguintes igualdades:

$$\mathbf{P}_n = \varphi_n^{-1} \tag{3.12}$$

$$\mathbf{k}_n = \frac{\lambda^{-1}\mathbf{P}_{n-1}\mathbf{u}_n}{1 + \lambda^{-1}\mathbf{u}_n^T\mathbf{P}_{n-1}\mathbf{u}_n} \tag{3.13}$$

Podemos reescrever a equação 3.11 como segue:

$$\mathbf{P}_n = \lambda^{-1}\mathbf{P}_{n-1} - \lambda^{-1}\mathbf{k}_n\mathbf{u}_n^T\mathbf{P}_{n-1} \tag{3.14}$$

Sendo $\mathbf{P}_n$ a inversa da matriz de auto-correlação de dimensão $L \times L$ e $\mathbf{k_n}$ o vetor ganho de dimensão $L \times 1$. Reorganizando a equação 3.13, temos:

$$\begin{aligned} \mathbf{k}_n + \mathbf{k}_n\lambda^{-1}\mathbf{u}_n^T\mathbf{P}_{n-1}\mathbf{u}_n &= \lambda^{-1}\mathbf{P}_{n-1}\mathbf{u}_n \\ \mathbf{k}_n &= \left[\lambda^{-1}\mathbf{P}_{n-1} - \mathbf{k}\lambda^{-1}\mathbf{u}_n^T\mathbf{P}_{n-1}\right]\mathbf{u}_n \end{aligned} \tag{3.15}$$

Substituindo a equação 3.14 em 3.15 seque-se:

$$\begin{aligned} \mathbf{k}_n &= \mathbf{P}_n\mathbf{u}_n \\ \mathbf{k}_n &= \varphi_n^{-1}\mathbf{u}_n \end{aligned} \tag{3.16}$$

3.2.3 Atualização do vetor peso

No desenvolvimento de uma equação recursiva, utilizaremos as Equações 3.3, 3.7 e 3.12 para expressar a estimativa do mínimo quadrado do vetor peso, $\hat{\mathbf{w}}_n$, no instante n como segue:

$$\begin{aligned} \hat{\mathbf{w}}_n &= \mathbf{P}_n.\left[\lambda\mathbf{z}_{n-1} + \mathbf{u}_n\right] \\ \hat{\mathbf{w}}_n &= \lambda\mathbf{P}_n.\mathbf{z}_{n-1} + \mathbf{P}_n\mathbf{u}_n \end{aligned} \tag{3.17}$$

Substituindo a equação 3.14 na equação 3.17, temos:

$$\hat{\mathbf{w}}_n = \mathbf{P}_{n-1}.\mathbf{z}_{n-1} - \mathbf{k}_n\mathbf{u}_n.\mathbf{P}_{n-1}.\mathbf{z}_{n-1} + \mathbf{P}_n\mathbf{u}_n d_n \tag{3.18}$$

Das equações 3.16 e 3.18, segue-se:

$$\hat{\mathbf{w}}_n = \hat{\mathbf{w}}_{n-1} - \mathbf{k}_n.\epsilon_n \tag{3.19}$$

Sendo ϵ_n a estimativa do erro a priori definida por:

$$\epsilon_n = d_n - \mathbf{u}_n^T.\hat{\mathbf{w}}_{n-1} \tag{3.20}$$

3.3 Convergência do algoritmo *RLS*

Estudaremos nesta seção a convergencia do algoritmo *RLS* no contexto de um problema de modelagem de sistema. Como planta consideramos um regressor múltiplo linear caracterizado pela equação

$$d_n = \mathbf{w}_*^{\mathbf{T}}\mathbf{u}_n + e_n \tag{3.21}$$

sendo $\mathbf{w}_*$ o vetor peso do regressor, $\mathbf{u}_n$ o vetor entrada, e_n é o ruido da planta e d_n a saída da planta. O erro de medição e_n do processo é branco com média zero e variância σ^2

3.3.1 O comportamento médio do vetor peso no algoritmo *RLS*

De 3.3 e 3.5 obtemos

$$\hat{\mathbf{w}}_n = \varphi_n^{-1}\sum_{i=1}^{n}\lambda^{n-i}.\mathbf{u}_i d_i \tag{3.22}$$

Substituindo 3.21 em 3.22 e usando 3.4, temos

$$\hat{\mathbf{w}}_n = \hat{\mathbf{w}}_* + \varphi_n^{-1}\sum_{i=1}^{n}\lambda^{n-i}.\mathbf{u}_i e_i \tag{3.23}$$

Aplicando o operador esperança em ambos os membros da Equação 3.23 e reconhecemos do princípio de ortogonalidade que todos os elementos do vetor $\mathbf{u}_n$ são ortogonais ao erro e_n, obtemos

$$E[\hat{\mathbf{w}}_n] = \hat{\mathbf{w}}_* \tag{3.24}$$

3.3.2 Matriz de correlação do vetor desvio

Definimos o vetor de desvio como

$$\mathbf{v}_n = \hat{\mathbf{w}}_n - \mathbf{w}_* \tag{3.25}$$

De 3.23 temos

$$\mathbf{v}_n = \varphi_n^{-1} \sum_{i=1}^{n} \lambda^{n-i}.\mathbf{u}_i e_i \tag{3.26}$$

Definimos a matriz de correlação do vetor desvio da seguinte forma

$$\mathbf{K}_n = E[\mathbf{v}_n \mathbf{v}_n^T] \tag{3.27}$$

Substituindo 3.26 em 3.27 e notando que $[\varphi_n^{-1}]^T = \varphi_n^{-1}$ e $[\lambda^{n-i}]^T = \lambda^{n-i}$, obtemos

$$\mathbf{K}_n = E\left[\left(\varphi_n^{-1} \sum_{i=1}^{n} \lambda^{n-i}.\mathbf{u}_i e_i\right)\left(\sum_{i=1}^{n} \lambda^{n-i} e_i^T \mathbf{u}_i^T\right)\varphi_n^{-1}\right] \tag{3.28}$$

Podemos observar em [2] que para o rigoroso cálculo da Equação 3.28 devemos fazer as seguintes hipóteses:

1. O vetor de entrada $\mathbf{u}_i$ constitui amostras de um processo estatístico. Assim, podemos usar as médias do tempo ao invés do conjunto de médias.

2. O fator de esquecimento λ é muito próximo de 1.

3. O tempo n o qual $\mathbf{K}_n$ é calculado é grande.

Notamos de 3.4 que φ_n é uma soma ponderada dos produtos externos

$$\mathbf{u}_n\mathbf{u}_n^T,\ \mathbf{u}_{n-1}\mathbf{u}_{n-1}^T,\ \mathbf{u}_{n-2}\mathbf{u}_{n-2}^T, ...$$

Assim, considerando as hipóteses acima, encontramos

$$\varphi_n \approx \frac{1-\lambda^n}{1-\lambda}\mathbf{R} \tag{3.29}$$

sendo $\mathbf{R} = E[\mathbf{u}_n \mathbf{u}_n^T]$ a matriz de correlação do vetor de entrada.

Substituindo 3.29 em 3.28 e notando que $E[e_n e_n^T] = \sigma^2$, obtemos

$$\begin{aligned} \mathbf{K}_n &= \sigma^2 \left(\frac{1-\lambda^n}{1-\lambda} \right)^2 \cdot \left(\frac{1-\lambda^{2n}}{1-\lambda^2} \right) \mathbf{R}^{-1} \\ &= \sigma^2 \left(\frac{1-\lambda}{1+\lambda} \right) \cdot \left(\frac{1+\lambda^n}{1-\lambda^n} \right) \mathbf{R}^{-1} \end{aligned} \tag{3.30}$$

3.3.3 Curva de apredizagem do algoritmo *RLS*

Para algoritmo *RLS* é conveniente usar o erro ϵ_n para definir o Erro Quadrático Médio(*mean-squared error-MSE*), assim podemos expressar a curva de apendizagem do algoritmo *RLS* em termos do erro a priori como

$$\xi_n = E[\epsilon_n^2] \tag{3.31}$$

sendo de 4.29 e 3.21 o erro a priori escrito da seguinte forma:

$$\epsilon_n = e_n - \mathbf{v}_{n-1}^T \mathbf{u}_n \tag{3.32}$$

Substituindo 3.32 em 3.31 e espandindo os termos, obtemos

$$\begin{aligned} \xi_n &= E\left[e_n^2\right] + E\left[\mathbf{u}_n^T \mathbf{v}_{n-1} \mathbf{v}_{n-1}^T \mathbf{u}_n\right] \\ &- E\left[\mathbf{v}_{n-1}^T \mathbf{u}_n e_n\right] - E\left[e_n \mathbf{u}_n^T \mathbf{v}_{n-1}\right] \end{aligned} \tag{3.33}$$

O primeiro valor esperado do lado direito da equação 3.33 é simplesmente a variância de e_n. Para os demais valores esperados podemos fazer as seguintes observações

1. A estimativa $\hat{\mathbf{w}}_{n-1}$, e portanto o vetor desvio $\mathbf{v_{n-1}}$, é independente do vetor de entrada $\mathbf{u}_n$; o último é assunido como sendo derivado de um processo estácionario de amplo sentido de média zero. Consequentemente, podemos usar esta independencia estatística junto com os resultados conhecidos de

álgebra matricial para expressar o segundo valor esperado do lado direito da equação 3.33 como segue-se:

$$\begin{aligned} E\left[\mathbf{u}_n^T \mathbf{v}_{n-1} \mathbf{v}_{n-1}^T \mathbf{u}_n\right] &= E\left[tr\left\{\mathbf{u}_n^T \mathbf{v}_{n-1} \mathbf{v}_{n-1}^T \mathbf{u}_n\right\}\right] \\ &= E\left[tr\left\{\mathbf{u}_n^T \mathbf{u}_n \mathbf{v}_{n-1} \mathbf{v}_{n-1}^T\right\}\right] \\ &= tr\left\{E\left[\mathbf{u}_n^T \mathbf{u}_n\right] E\left[\mathbf{v}_{n-1} \mathbf{v}_{n-1}^T\right]\right\} \\ &= tr\left\{\mathbf{R}\mathbf{K}_{n-1}\right\} \end{aligned} \tag{3.34}$$

sendo que na última linha utilizamos as definições de médias da matriz de correlação $R = E[\mathbf{u}_n \mathbf{u}_n^T]$ e da matriz de correlação do vetor desvio $\mathbf{K}_n = E[\mathbf{v}_n \mathbf{v}_n^T]$.

2. O error de medição e_n depende do vetor de entrada $\mathbf{u}_n$; isto segue de uma simpes manipulação da Equação 3.21. O vetor desvio $\mathbf{v}_{n-1}^T$ é portanto independente de $\mathbf{u}_n$ e e_n. Entretanto podemos mostrar que o terceiro valor esperado do lado direito da Equação 3.33 é zero reformulando isto como segue:

$$E\left[\mathbf{v}_{n-1}^T \mathbf{u}_n e_n\right] = E\left[\mathbf{v}_{n-1}^T\right] . E\left[\mathbf{u}_n e_n\right] \tag{3.35}$$

Reconhecemos do princípio de ortogonalidade que todos os elementos do vetor $\mathbf{u}_n$ são ortogonais ao erro de medição e_n, isto é:

$$E\left[\mathbf{v}_{n-1}^T \mathbf{u}_n e_n\right] = 0 \tag{3.36}$$

3. O quarto valor esperado do lado direito da Equação 3.33 tem a mesma forma matemática considerada no item 2. Entretanto podemos dizer que este valor esperado é igual a zero:

$$E\left[e_n \mathbf{u}_n^T \mathbf{v}_{n-1}\right] = 0 \tag{3.37}$$

Assim, reconhecendo que $E[e_n^2] = \xi_{mim}$, e usando os resultados das Equações 3.34 até 3.37 em 3.33, obtemos a seguinte fórmula para o erro quadrático médio no algoritmo *RLS*:

$$\xi_n = \xi_{min} + tr\left\{\mathbf{R}\mathbf{K}_{n-1}\right\} \tag{3.38}$$

sendo ξ_{min} o mínimo *MSE* do filtro encontrado quando uma estimativa perfeita de $\mathbf{w}_*$ é calculada. Substituindo 3.30 em 3.38 obtemos

$$\xi_n = \xi_{min} + \left(\frac{1-\lambda}{1+\lambda}\right) \cdot \left(\frac{1+\lambda^{n-1}}{1-\lambda^{n-1}}\right) \cdot L\xi_{min} \tag{3.39}$$

Este resultado descreve a curva de aprendizagem do algoritmo *RLS*. Na figura 3.1 podemos ver uma curva típica de aprendizagem resultante do uso deste algoritmo.

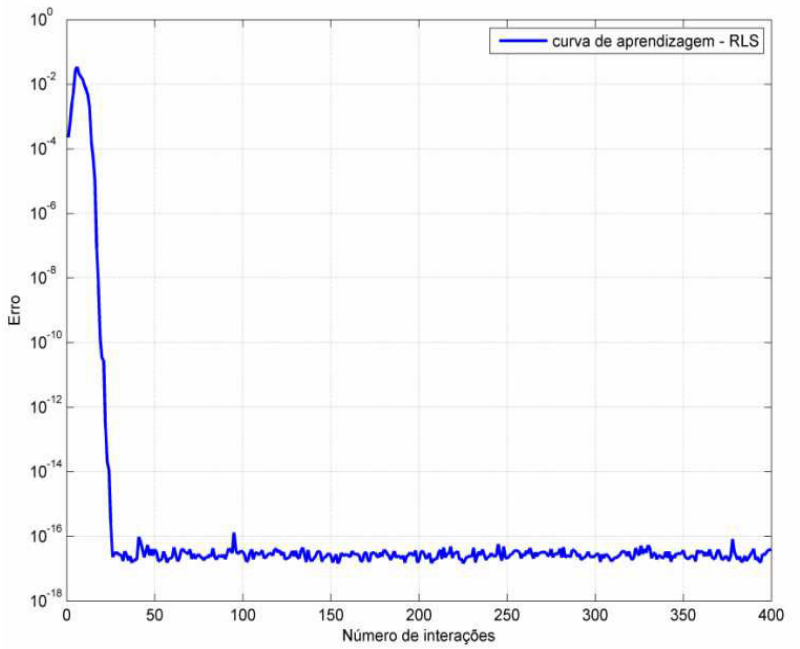

Figura 3.1: Curva de aprendizagem do algoritmo *RLS*. Na horizontal temos o número de iterações e na vertical o erro.

3.3.4 Tempo de aprendizagem

A esta altura é instrutivo que façamos uma análise do comportamento do algoritmo *RLS*. O segundo termo do lado direito da Equação 3.39 é um valor positivo

que indica o desvio de ξ_n de ξ_{min}. Notamos também que a velocidade a qual estes termos convergem é determinada pelo termo exponencial λ^{n-1}, ou equivalentemente λ^n. Desta forma, definimos o tempo de aprendizagem $\tau_{\mathcal{RLS}}$ associado com o algoritmo *RLS* usando a seguinte equação:

$$\lambda^n = e^{\frac{-n}{\tau_{\mathcal{RLS}}}} \tag{3.40}$$

Resolvendo 3.40 para $\tau_{\mathcal{RLS}}$ obtemos

$$\tau_{\mathcal{RLS}} = -\frac{1}{\ln \lambda} \tag{3.41}$$

Para simplificar 3.41 usaremos a seguinte aproximação

$$\ln \lambda = \ln(1 - (1 - \lambda)) \approx -(1 - \lambda) \tag{3.42}$$

Substituindo 3.42 em 3.41 temos

$$\tau_{\mathcal{RLS}} \approx \frac{1}{(1 - \lambda)} \tag{3.43}$$

Deste resultado, notamos que o comportamento da convergência do algoritmo *RLS* é independente dos autovalores da matriz de correlação das entradas.

3.3.5 Excesso do erro quadrático médio e o desajuste

Na figura 3.1, podemos ver que quando os pesos não são iguais a $\mathbf{w}_*$, o erro quadrático médio (ξ_n) é maior que o erro quadrático médio mínimo (ξ_{min}). Temos, assim, um excesso no erro final

Definimos, então, o excesso do erro quadrático médio, $ExcessoMSE$, como a diferença entre o erro quadrático médio atual e o erro quadrático médio mínimo [2]:

$$ExcessoMSE = \lim_{n \to \infty} \xi_n - \xi_{min} \tag{3.44}$$

Usando 3.39 e 3.44 obtemos

$$ExcessoMSE = \left(\frac{1-\lambda}{1+\lambda}\right) \cdot L\xi_{min} \tag{3.45}$$

Definimos, também, o ExcessoMSE normalizado pelo erro quadrático médio mínimo, como o desajuste $\mathcal{M}_{\mathcal{RLS}}$.

$$\mathcal{M}_{\mathcal{RLS}} = \frac{ExcessoMSE}{\xi_{min}}. \tag{3.46}$$

Substituido 3.45 em 3.46 temos

$$\mathcal{M}_{\mathcal{RLS}} = \left(\frac{1-\lambda}{1+\lambda}\right) \cdot L \tag{3.47}$$

sendo L o tamanho do filtro.

3.4 Conclusão do Capítulo

Neste capítulo, realizou-se a dedução do algoritmo *RLS* e descrevemos as equações que determinam sua condição de convergência. O tempo de aprendizagem e o desajuste também são enfatizados, pois os mesmso são utilizados como referência comparativa de outros algoritmos adaptativos.

Capítulo 4

O Algoritmo Recursivo Não Linear - RNL

4.1 Introdução

Neste capítulo desenvolveremos um algoritmo adaptativo baseado em funções não lineares inspirado no algoritmo *RLS* padrão. Chamaremos esse novo algoritmo de Recursivo Não Linear, no qual escolhemos a função $\varepsilon_n = \sum_{j=1}^{M}\sum_{i=1}^{n}\left\{\lambda^{n-i}\left[e_i\right]^{2j}\right\}$ como critério a ser aplicado sobre o erro. O nosso objetivo é determinar um algoritmo que ajuste os pesos do CLA de forma tal a minimizar esta função. Mostraremos, também, que a superfície de desempenho obtida por este critério oferece maior velocidade de convergência, bem como um menor desajuste na busca do peso mínimo.

4.2 Dedução do algoritmo RNL

A função $\varepsilon_n = \sum_{j=1}^{M}\sum_{i=1}^{n}\left\{\lambda^{n-i}\left[e_i\right]^{2j}\right\}$, sendo M e n inteiros positivos, é uma função não linear, par, contínua, simétrica, cujo gráfico está representado na figura 4.1. Esta função, como podemos ver, não tem mínimo local, apenas o mínimo global.

Uma outra característica dos elementos deste conjunto de funções é que, para um valor fixo de M, podemos determinar intervalos $[-\delta, \delta]$ onde as curvas destas função têm inclinação maior do que a curva da função quadrática, neste mesmo intervalo. Podemos observar esta característica na figura (4.2), onde temos plotados os gráficos das funções $\varepsilon^2 + \varepsilon^4 + \varepsilon^6$ e ε^2.

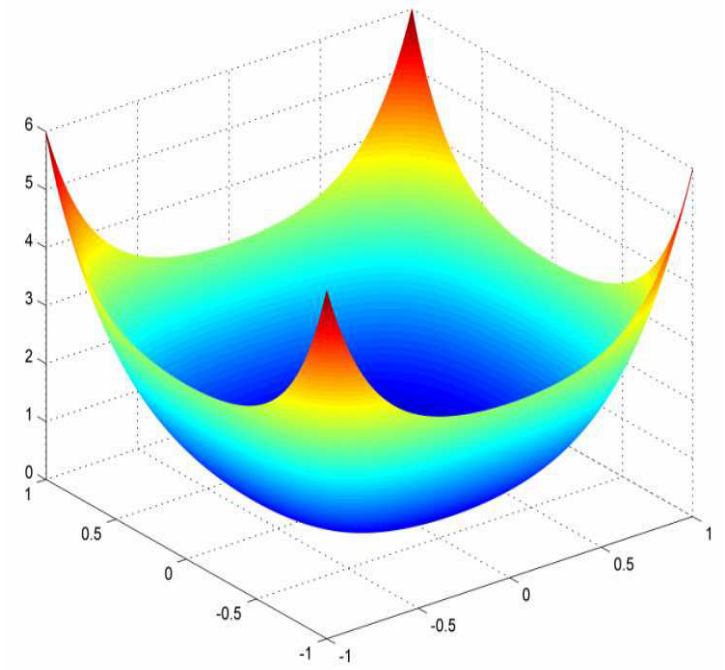

Figura 4.1: Porção da superfície gerada pela função $\varepsilon^2 + \varepsilon^4 + \varepsilon^6$ juntamente com alguns contornos.

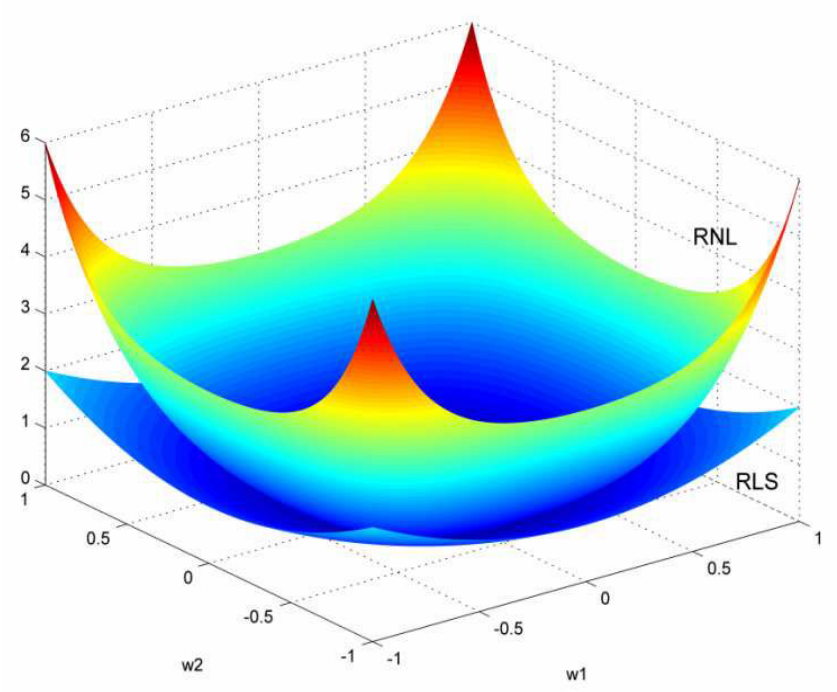

Figura 4.2: Gráficos das funções $\varepsilon^2 + \varepsilon^4 + \varepsilon^6$ e ε^2, onde podemos ver a maior inclinação da primeira, no intervalo $[-1; 1]$

Para desenvolver o algoritmo RNL, utilizamos como função de custo a função

$$\varepsilon_n = \sum_{j=1}^{M} \sum_{i=1}^{n} \left\{ \lambda^{n-i} \left[e_i \right]^{2j} \right\} \tag{4.1}$$

sendo M e n inteiros positivos e $0 << \lambda < 1$ fator peso exponencial ou fator de esquecimento.

Derivando a Equação 4.1 em relação a $\mathbf{w}$, obtemos

$$\frac{\partial \varepsilon}{\partial \mathbf{w}} = \sum_{j=1}^{M} \sum_{i=1}^{N} \left\{ \lambda^{n-i}. - \mathbf{u}_i . 2j \left(d_i - \hat{\mathbf{w}}_n^T \mathbf{u}_i \right)^{2j-1} \right\} \tag{4.2}$$

Desenvolvendo o binômio podemos reescrever 4.2 da seguinte forma

$$\begin{aligned} \frac{\partial \varepsilon}{\partial \mathbf{w}} \quad &\approx \quad - \sum_{j=1}^{M} \left\{ 2j \sum_{i=1}^{N} \left[\lambda^{n-i} \mathbf{u}_i d_i^{2j-1} \right] \right\} \\ &+ \quad \left[\sum_{j=1}^{M} \left\{ 2j \left(2j-1 \right) \sum_{i=1}^{N} \left[\lambda^{n-i} d_i^{2j-2} \mathbf{u}_i \mathbf{u}_i^T \right] \right\} \right] \mathbf{w}_n \end{aligned} \tag{4.3}$$

Assim, o valor ótimo do vetor peso, $\hat{\mathbf{w}}_n$, para que a função da Equação 4.1 atinja o valor mínimo é definido pela equação normal escrita em forma matricial

$$\left[\mathbf{z}_{n,j} \right] - \left[\phi_{n,j} \right] . \hat{\mathbf{w}}_n = 0 \tag{4.4}$$

sendo a matriz correlação do vetor de entrada de dimensão $L \times L$ agora definida por,

$$\phi_{n,j} = \sum_{j=1}^{M} \left\{ j. \left(2j-1 \right) \sum_{i=1}^{N} \left[\lambda^{n-i} . d_i^{2j-2} . \mathbf{u}_i \mathbf{u}_i^T \right] \right\} \tag{4.5}$$

E o vetor correlação cruzada entre as entradas do CLA e a resposta desejada é definido por,

$$\mathbf{z}_{n,j} = \sum_{j=1}^{M} \left\{ j \sum_{i=1}^{N} \left[\lambda^{n-i} . d_i^{2j-1} . \mathbf{u}_i \right] \right\} \tag{4.6}$$

sendo L o tamanho do filtro adaptativo.

Isolando o termo corespondente a $i = n$ da Equação 4.5, podemos escrever

$$\begin{aligned}\phi_{n,j} &= \sum_{j=1}^{M}\left\{j.\left(2j-1\right)\sum_{i=1}^{n-1}\left[\lambda^{n-i}.d_i^{2j-2}\mathbf{u}_i\mathbf{u}_i^T\right]\right\}+\sum_{j=1}^{M}\left\{j.\left(2j-1\right)d_n^{2j-2}\mathbf{u}_n.\mathbf{u}_n^T\right\}\\ &= \phi_{n-1,j}+\left[\sum_{j=1}^{M}\left\{j.\left(2j-1\right).d_n^{2j-2}\right\}\right]\mathbf{u}_n.\mathbf{u}_n^T \qquad (4.7)\end{aligned}$$

Analogamente podemos escrever a Equação 4.6 da seguinte forma

$$\mathbf{z}_{n,j}=\mathbf{z}_{n-1,j}+\left[\sum_{j=1}^{M}\left\{j.d_n^{2j-1}\right\}\right]\mathbf{u}_n \qquad (4.8)$$

4.2.1 O algoritmo RNL ponderado exponencialmente

Como a matriz de correlação é positivamente definida e não singular, podemos aplicar o lema de inversão de matrizes para equação recursiva 4.5. Primeiramente faremos as seguintes identificações:

$$\begin{cases}\mathbf{A}=\varphi_{n,j}\\ \mathbf{B}^{-1}=\varphi_{n-1,j}\rightarrow\mathbf{B}=\left[\varphi_{n-1,j}\right]^{-1}\\ \mathbf{C}=\mathbf{u}_n\\ \mathbf{D}^{-1}=\sum\limits_{j=1}^{M}\left\{j.\left(2j-1\right)d_n^{2j-2}\right\}\end{cases} \qquad (4.9)$$

Substituindo as definições 4.9 na equação 3.9, obtemos a relação de recursividade da matriz $\varphi_{n,j}$, ou seja:

$$\left[\varphi_{n,j}\right]^{-1}=\left[\varphi_{n-1,j}\right]^{-1}-\frac{\left[\varphi_{n-1,j}\right]^{-1}\mathbf{u}_n\mathbf{u}_n^T\left[\varphi_{n-1,j}\right]^{-1}}{\mathbf{D}+\mathbf{u}_n^T\left[\varphi_{n-1,j}\right]^{-1}\mathbf{u}_n} \qquad (4.10)$$

Por conveniência computacional podemos escrever as seguintes igualdades:

$$\mathbf{P}_{n,j}=\left[\varphi_{n,j}\right]^{-1} \qquad (4.11)$$

$$\mathbf{k}_{n,j}=\frac{\mathbf{P}_{n-1,j}\mathbf{u}_n}{\frac{1}{\alpha_1}+\mathbf{u}_n^T\mathbf{P}_{n-1,j}\mathbf{u}_n} \qquad (4.12)$$

sendo $\alpha_1 = \sum_{j=1}^{M} \{j.\,(2j-1)\,d_n^{2j-2}\}$

Podemos reescrever a equação 4.10 como segue:

$$\mathbf{P}_{n,j} = \mathbf{P}_{n-1,j} - \mathbf{k}_{n,j}\mathbf{u}_n^T\mathbf{P}_{n-1,j} \tag{4.13}$$

sendo $\mathbf{P}_{n,j}$ a inversa da matriz de auto-correlação de dimensão $L \times L$ e $\mathbf{k_{n,j}}$ o vetor ganho de dimensão $L \times 1$. Reorganizando a equação 4.12, temos:

$$\begin{aligned} \mathbf{k}_{n,j}\frac{1}{\alpha_1} &+ \mathbf{k}_{n,j}\mathbf{u}_n^T\mathbf{P}_{n-1,j}\mathbf{u}_n = \mathbf{P}_{n-1,j}\mathbf{u}_n \\ \mathbf{k}_{n,j}\frac{1}{\alpha_1} &= \left[\mathbf{P}_{n-1,j} - \mathbf{k}_{n,j}\mathbf{u}_n^T\mathbf{P}_{n-1,j}\right]\mathbf{u}_n \end{aligned} \tag{4.14}$$

Substituindo a equação 4.13 em 4.14 seque-se:

$$\begin{aligned} \mathbf{k}_{n,j}\frac{1}{\alpha_1} &= \mathbf{P}_{n,j}\mathbf{u}_n \\ \mathbf{k}_{n,j}\frac{1}{\alpha_1} &= \left[\varphi_{n,j}\right]^{-1}\mathbf{u}_n \end{aligned} \tag{4.15}$$

4.2.2 Atualização do vetor peso

No desenvolvimento de uma equação recursiva utilizaremos as Equações 4.4, 4.8 e 4.11 para expressar a estimativa $\hat{\mathbf{w}}_n$ do vetor peso no instante n como segue:

$$\begin{aligned} \hat{\mathbf{w}}_n &= \mathbf{P}_{n,j}\left[\mathbf{z}_{n-1,j} + \alpha_2\mathbf{u}_n\right] \\ \hat{\mathbf{w}}_n &= \mathbf{P}_{n,j}\mathbf{z}_{n-1,j} + \alpha_2\mathbf{P}_n\mathbf{u}_n \end{aligned} \tag{4.16}$$

sendo $\alpha_2 = \sum_{j=1}^{M} \{jd_n^{2j-1}\}$

Substituindo a equação 4.13 na equação 4.16, temos:

$$\hat{\mathbf{w}}_n = \mathbf{P}_{n-1,j}\mathbf{z}_{n-1,j} - \mathbf{k}_{n,j}\mathbf{u}_n^T.\mathbf{P}_{n-1,j}\mathbf{z}_{n-1,j} + \alpha_2 \cdot \mathbf{k}_{n,j} \cdot \frac{1}{\alpha_1} \tag{4.17}$$

Das equações 4.15 e 4.17, segue-se:

$$\hat{\mathbf{w}}_n = \hat{\mathbf{w}}_{n-1} - \mathbf{k}_n \epsilon_n \tag{4.18}$$

sendo ϵ_n a estimativa do erro definida por:

$$\begin{aligned} \epsilon_n &= \alpha_2 . \frac{1}{\alpha_1} - \mathbf{u}_n^T . \hat{\mathbf{w}}_{n-1} \\ &= d_n - \mathbf{u}_n^T . \hat{\mathbf{w}}_{n-1} \end{aligned} \tag{4.19}$$

4.2.3 Resumo do algoritmo RNL

Podemos observar que o algoritmo RNL é dado pela seqüência cíclica da sequintes equações:

$$\mathbf{k}_{n,j} = \frac{\mathbf{P}_{n-1,j}\mathbf{u}_n}{\frac{1}{\alpha_1} + \mathbf{u}_n^T \mathbf{P}_{n-1,j}\mathbf{u}_n} \tag{4.20}$$

sendo $\alpha_1 = \sum_{j=1}^{M} \{j.\,(2j-1)\,d_n^{2j-2}\}$

$$\mathbf{P}_{n,j} = \mathbf{P}_{n-1,j} - \mathbf{k}_{n,j}\mathbf{u}_n^T \mathbf{P}_{n-1,j} \tag{4.21}$$

$$\hat{\mathbf{w}}_n = \hat{\mathbf{w}}_{n-1} - \mathbf{k}_n \epsilon_n \tag{4.22}$$

sendo ϵ_n a estimativa do erro definida por:

$$\epsilon_n = \alpha_2 . \frac{1}{\alpha_1} - \mathbf{u}_n^T . \hat{\mathbf{w}}_{n-1} \tag{4.23}$$

sendo $\alpha_2 = \sum_{j=1}^{M} \{j d_n^{2j-1}\}$

4.3 Convergência do algoritmo RNL

Estudaremos nesta seção a convergencia do algoritmo RNL no contexto de um problema de modelagem de sistema, da mesma forma que estudamos a convergência do algoritmo *RLS* no capítulo anterior.

4.3.1 O comportamento médio do vetor peso no algoritmo RNL

Consideramos um regressor multiplo linear caracterizado pela equação

$$d_n = e_n + \mathbf{w}_*^T \mathbf{u}_n \tag{4.24}$$

sendo $\mathbf{w}_*$ o vetor peso do regressor, $\mathbf{u}_n$ o vetor entrada, e_n é o ruido da planta e d_n a saída da planta. O erro de medição e_n do processo é branco com média zero e variância σ^2

De 4.4 e 4.6 obtemos a seguinte equação:

$$\hat{\mathbf{w}}_n = \left[\varphi_{n,j}\right]^{-1} . \left\{ \sum_{j=1}^{M} j. \sum_{i=1}^{n} \left[\lambda^{n-i} . \mathbf{u}_i . d_i^{2j-1} \right] \right\} \tag{4.25}$$

Substituindo 4.24 em 4.25, temos

$$\hat{\mathbf{w}}_n = \left[\varphi_{n,j}\right]^{-1} . \left\{ \sum_{j=1}^{M} j. \sum_{i=1}^{n} \left[\lambda^{n-i} . \mathbf{u}_i . \left(\mathbf{w}_*^T \mathbf{u}_i + e_i \right)^{2j-1} \right] \right\} \tag{4.26}$$

Utilizando desenvoldimento Binomial, podemos reescrever a Equação 4.26 como

$$\hat{\mathbf{w}}_n \ = \mathbf{w}_* + \left[\varphi_{n,j}\right]^{-1} . \left\{ \sum_{j=1}^{M} j. \sum_{i=1}^{n} \left[\lambda^{n-i} . \mathbf{u}_i . e_i^{2j-1} \right] \right\} \tag{4.27}$$

Aplicando o operador esperança em ambos os membros da Equação 4.27 e reconhecemos do princípio de ortogonalidade que todos os elementos do vetor $\mathbf{u}_n$ são ortogonais ao erro e_n, obtemos

$$E\left[\hat{\mathbf{w}}_n\right] \ = \mathbf{w}_* \tag{4.28}$$

4.3.2 Matriz de correlação do vetor desvio

Definimos o vetor desvio como

$$\mathbf{v}_n = \hat{\mathbf{w}}_n - \mathbf{w}_* \tag{4.29}$$

De 4.27 temos

$$\mathbf{v}_n = [\varphi_{n,j}]^{-1} \cdot \left\{ \sum_{j=1}^{M} j \cdot \sum_{i=1}^{n} \left[\lambda^{n-i} \cdot \mathbf{u}_i \cdot e_i^{2j-1} \right] \right\} \tag{4.30}$$

Definimos a matriz de correlação do vetor desvio da seguinte forma

$$\mathbf{K}_n = E[\mathbf{v}_n \mathbf{v}_n^T] \tag{4.31}$$

Substituindo 4.30 em 4.31 e notando que $\left([\varphi_{n,j}]^{-1}\right)^T = [\varphi_{n,j}]^{-1}$ e $(\lambda^{n-i})^T = \lambda^{n-i}$, obtemos

$$\begin{aligned} \mathbf{K}_n = E \Bigg[& [\varphi_{n,j}]^{-1} \cdot \left\{ \sum_{j=1}^{M} j \cdot \sum_{i=1}^{n} \left[\lambda^{n-i} \cdot \mathbf{u}_i \cdot e_i^{2j-1} \right] \right\} \\ & \cdot \left\{ \sum_{j=1}^{M} j \cdot \sum_{i=1}^{n} \left[\lambda^{n-i} \cdot \left[e_i^{2j-1} \right]^T \mathbf{u}_i^T \right] \right\} \cdot [\varphi_{n,j}]^{-1} \Bigg] \end{aligned} \tag{4.32}$$

Expandindo os somatórios da Equação 4.32 e notando que o erro de medição neste processo é um ruido branco, segue-se

$$\mathbf{K}_n = \sigma^2 \cdot \; E \left[[\varphi_{n,j}]^{-1} \cdot (\mathbf{U}_n \mathbf{\Lambda}_n^2 \mathbf{U}_n^T) \cdot [\varphi_{n,j}]^{-1} \right] \tag{4.33}$$

sendo $\mathbf{\Lambda}_n$ uma matriz diagonal formada pelos fatores exponencias $1, \lambda, \lambda^2, ...,$. Definimos também a matriz das amostras de entrada como

$$\mathbf{U}_n = [\mathbf{u}_1 \;\; \mathbf{u}_2 \;\; ... \;\; \mathbf{u}_n] \tag{4.34}$$

Podemos observar em [2] que para o rigoroso cálculo da Equação 4.33 devemos fazer as seguintes hipóteses:

1. O vetor de entrada $\mathbf{u}_i$ constitui amostras de um processo estatístico. Assim, podemos usar as médias do tempo ao invés do conjunto de médias.

2. O fator de esquecimento λ é muito próximo de 1.

3. O tempo n o qual $\mathbf{K}_n$ é calculado é grande.

Assim, considerando as hipóteses acima, encontramos que,

$$\mathbf{U}_n \mathbf{\Lambda}_n \mathbf{U}_n^T \approx \frac{1-\lambda^n}{1-\lambda}\mathbf{R} \tag{4.35}$$

Analogamente, podemos reescrever a Equação 4.5 como

$$\varphi_{n,j} \approx \frac{1-\lambda^n}{1-\lambda}\mathbf{R} \cdot \beta \tag{4.36}$$

sendo $\beta = \sum_{j=1}^{M} [j(2j-1) \cdot d_n^{2j-2}]$

Substituindo 4.35 e 4.36 em 4.33 obtemos

$$\begin{aligned} \mathbf{K}_n &= \sigma^2 \cdot \left\{ E\left[\left(\frac{1-\lambda}{1-\lambda^n}\mathbf{R}^{-1} \cdot \frac{1}{\beta} \right) \left(\frac{1-\lambda^{2n}}{1-\lambda^2}\mathbf{R} \cdot \beta \right) \right] \right\} \left(\frac{1-\lambda}{1-\lambda^n}\mathbf{R}^{-1} \cdot \frac{1}{\beta} \right) \\ &= \sigma^2 \cdot \left\{ \frac{1}{\beta} \cdot \frac{1-\lambda}{1+\lambda} \cdot \frac{1+\lambda^n}{1-\lambda^n}\mathbf{R}^{-1} \right\} \end{aligned} \tag{4.37}$$

4.3.3 Curva de apredizagem do algoritmo RNL

Para algoritmo RNL podemos expressar a curva de apendizagem em termos do erro ϵ_n como

$$\xi_n = \sum_{j=1}^{M} E\left[|\epsilon_n|^{2j} \right] \tag{4.38}$$

sendo M inteiro positivo.

Usando 4.29 e 4.24, obtemos

$$\epsilon_n = e_n - \mathbf{v}_{n-1}^T \mathbf{u}_n \tag{4.39}$$

sendo $\mathbf{v_{n-1}}$ o vetor desvio no instante $n-1$.

Substituindo a Equação 4.39 em 4.38, temos

$$\begin{aligned} \xi_n &= \sum_{j=1}^{M} E\left[\left|e_n - \mathbf{v}_{n-1}^T \mathbf{u}_n\right|^{2j}\right] \\ &= \sum_{j=1}^{M} E\left[\left|-\mathbf{v}_{n-1}^T \mathbf{u}_n + e_n\right|^{2j}\right] \end{aligned} \tag{4.40}$$

Utilizando desenvolvimento Binomial e notando que $\mathbf{v}$ é próximo de zero podemos desconsiderar os termos de alta potência de $\mathbf{v}$. Assim, reescrevemos a Equação 4.40 da seguinte forma:

$$\begin{aligned} \xi_n \approx \sum_{j=1}^{M} \Big\{ E\left[|e_n|^{2j}\right] + j(2j-1)E\left[|e_n|^{2j-2}\right] .E\left[\mathbf{u}_n^T \mathbf{v}_{n-1} \mathbf{v}_{n-1}^T \mathbf{u}_n\right] - \\ -jE\left[(\mathbf{v}_{n-1}^T \mathbf{u}_n)(e_n)^{2j-1}\right] - jE\left[(\mathbf{u}_n^T \mathbf{v}_{n-1})(e_n)^{2j-1}\right] \Big\} \end{aligned} \tag{4.41}$$

O primeiro valor esperado do lado direito da equação 4.41 definimos como sendo o momento $2j$ de e_n ou simplesmente a variância de e_n. Para os demais valores esperados, relembrado o princípio de ortogonalidade, podemos fazer as mesmas observações utilizadas pro algoritmo *RLS*. Desta forma, obtemos da 4.41

$$\xi_n = \sigma^2 + \sum_{j=1}^{M} \left[j(2j-1)\sigma^2 . tr\left\{\mathbf{R}\mathbf{K}_{n-1}\right\}\right] \tag{4.42}$$

Substituindo 4.37 em 4.42, obtemos

$$\xi_n = \sigma^2 + \sum_{j=1}^{M} \left\{ j(2j-1)\sigma^2 \cdot \frac{1-\lambda}{1+\lambda} \cdot \frac{1+\lambda^{n-1}}{1-\lambda^{n-1}} \cdot L \cdot \frac{1}{\beta} \cdot \sigma^2 \right\} \tag{4.43}$$

Fazendo a expanção do somatório e notando que $E[e_n^4] = 0$, reescrevemos 4.43

$$\xi_n \quad = \quad \sigma^2 + \frac{1-\lambda}{1+\lambda} \cdot \frac{1+\lambda^{n-1}}{1-\lambda^{n-1}} \cdot L \cdot \frac{1}{\beta} \cdot \sigma^2 \tag{4.44}$$

Assim, reconhecendo que $\sigma^2 = E[e_n^2] = \xi_{mim}$ é o mínimo MSE do filtro encontrado quando uma estimativa perfeita de $\mathbf{w}_*$ é calculada, obtemos a seguinte fórmula para o erro quadrático médio no algoritmo RNL:

$$\xi_n \quad = \quad \xi_{mim} + \frac{1-\lambda}{1+\lambda} \cdot \frac{1+\lambda^{n-1}}{1-\lambda^{n-1}} \cdot L \cdot \frac{1}{\beta} \cdot \xi_{mim} \tag{4.45}$$

sendo $\beta = \sum_{j=1}^{M} [j(2j-1) \cdot d_n^{2j-2}]$

Este resultado descreve a curva de aprendizagem do algoritmo RNL. Na figura 4.3 podemos ver uma curva típica de aprendizagem resultante do uso deste algoritmo.

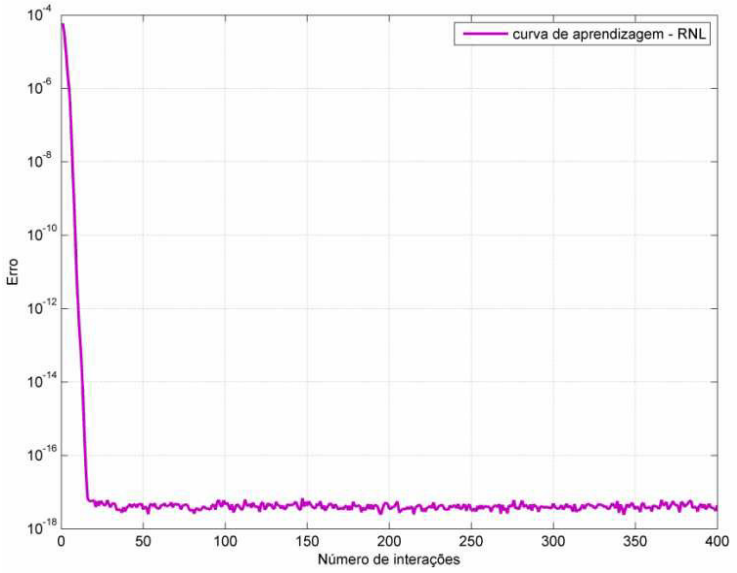

Figura 4.3: Curva de aprendizagem do algoritmo RNL. Na horizontal temos o número de iterações e na vertical o erro.

4.3.4 Análise do tempo de aprendizagem

Nesta seção faremos uma análise comparativa entre as constantes de tempo associadas ao algoritmo *RLS* e ao RNL.

Analogamente à análise feita na dedução da constante de tempo do *RLS*, no capítulo anterior, podemos verificar que a velocidade na qual o segundo termo termo do lado direito da Equação 4.45 é um valor positivo que indica o desvio de ξ_n de ξ_{min}. Notamos também que a velocidade a qual este termo converge é determinada pelo termo exponencial $\frac{1}{\beta} \cdot \lambda^{n-1}$, ou equivalentemente $\frac{1}{\beta} \cdot \lambda^n$. Desta forma, definimos a constante de tempo $\tau_{\mathcal{RNL}}$ associado com o algoritmo RNL usando a seguinte equação:

$$e^{\frac{-n}{\tau_{\mathcal{RNL}}}} = \frac{1}{\beta} \cdot \lambda^n \tag{4.46}$$

Resolvendo 4.46 para $\tau_{\mathcal{RNL}}$ obtemos

$$\frac{1}{\tau_{\mathcal{RNL}}} = -\ln \lambda + \frac{\ln \beta}{n} \tag{4.47}$$

Infelizmente a expressão dada em 4.47 não nos premite definir explicitamente o valor da constante de tempo $\tau_{\mathcal{RNL}}$. Entretanto, ela é útil para fazermos uma análise comparativa entre $\tau_{\mathcal{RNL}}$ e $\tau_{\mathcal{RLS}}$. De fato, lembrando que

$$\tau_{\mathcal{RLS}} = -\frac{1}{\ln \lambda} \tag{4.48}$$

então

$$\frac{1}{\tau_{\mathcal{RLS}}} = -\ln \lambda \tag{4.49}$$

Daí a Equação 4.47 pode ser reescrita na forma

$$\frac{1}{\tau_{\mathcal{RNL}}} = \frac{1}{\tau_{\mathcal{RLS}}} + \frac{\ln \beta}{n} \tag{4.50}$$

sendo $\beta = \sum_{j=1}^{M} [j(2j-1) \cdot d_n^{2j-2}]$ positivo,
então a expressão $\frac{\ln \beta}{n}$ também é positiva. Desta forma podemos afirmar que

$$\tau_{\mathcal{RNL}} < \tau_{\mathcal{RLS}} \tag{4.51}$$

4.3.5 Excesso do erro quadrático médio e o desajuste

Na figura 4.3, podemos ver que quando os pesos não são iguais a $\mathbf{w}_*$, o erro quadrático médio (ξ_n) é maior que o erro quadrático médio mínimo (ξ_{min}). Temos, assim, um excesso no erro final

Definimos, então, o excesso do erro quadrático médio, $ExcessoMSE$, como a diferença entre o erro quadrático médio atual e o erro quadrático médio mínimo [2]:

$$ExcessoMSE = \lim_{n\to\infty} \xi_n - \xi_{min} \tag{4.52}$$

Usando 4.45 e 4.52 obtemos

$$ExcessoMSE = \left(\frac{1-\lambda}{1+\lambda}\right) \cdot \frac{1}{\beta} \cdot L\xi_{min} \tag{4.53}$$

Definimos, também, o ExcessoMSE normalizado pelo erro quadrático médio mínimo, da Equação 4.45 como o desajuste $\mathcal{M}_{\mathcal{RNL}}$.

$$\mathcal{M}_{\mathcal{RNL}} = \left(\frac{1-\lambda}{1+\lambda}\right) \cdot \frac{1}{\beta} \cdot L \tag{4.54}$$

sendo $\beta = \sum_{j=1}^{M} [j(2j-1) \cdot d_n^{2j-2}]$ e L o tamanho do filtro.

4.4 Conclusões do Capítulo

Neste capítulo, descrevemos a idéia básica da utilização de estatística de alta ordem como uma forma de obtenção de mais informações sobre os sinais envolvidos em um processo adaptativo. Desenvolvimento de um novo algoritmo inspirado no algoritmo *RLS*, que utiliza como critério aplicado sobre o erro uma função não linear, a qual queremos minimizar. Isto origina o algoritmo Recursivo Não Linear - RNL. Deduzimos equações que determinam sua condição de convergência. O tempo de aprendizagem e o desajuste também são enfatizados.

Capítulo 5

Resultados e Discussões

5.1 Introdução

Objetivando verificar a exatidão das equações deduzidas no capítulo anterior, fizemos simulações, onde comparamos os desempenhos dos algoritmos *RLS* e *RNL*.

5.2 Simulações com o algoritmo RNL

Em nossa simulação verificamos que o novo algoritmo trabalha corretamente na identificação de um pequeno filtro FIR, como mostra a Figura 5.1. Este filtro é caracterizado pela resposta impulso h. O sinal de entrada, u_i, foi simulado como um sinal aleatório uniformemente distribuído, limitado no intervalo $[-1, 1]$. Filtramos este sinal por h obtendo o sinal desejado d_i.

Na figura 5.2 podemos visualizar a curva de aprendizagem do algoritmo RLS e as curvas de aprendizagens do algoritmo RNL com 2 termos e com 4 termos.

5.3 Discussões

Um novo algoritmo foi introduzido para ajustar os pesos de um filtro adaptativo tal que o valo esperado do erro de grau $2j$, sendo j inteiro positivo deve ser minimizado. O desenvolvimento deste algoritmo adaptativo foi baseado em funções não lineares inspirado no algoritmo *Recursive Lest Square (RLS)* proposto por Haykin [1]. Derivamos à Equação 4.1 e desemvolvemos a Equação 4.4 para obter o valor ótimo do vetor peso w_n. A estrutura de atualização do algoritmo RNL é dada pela Equação 4.18. A partir das expressões 3.47 e 4.54 podemos observar que

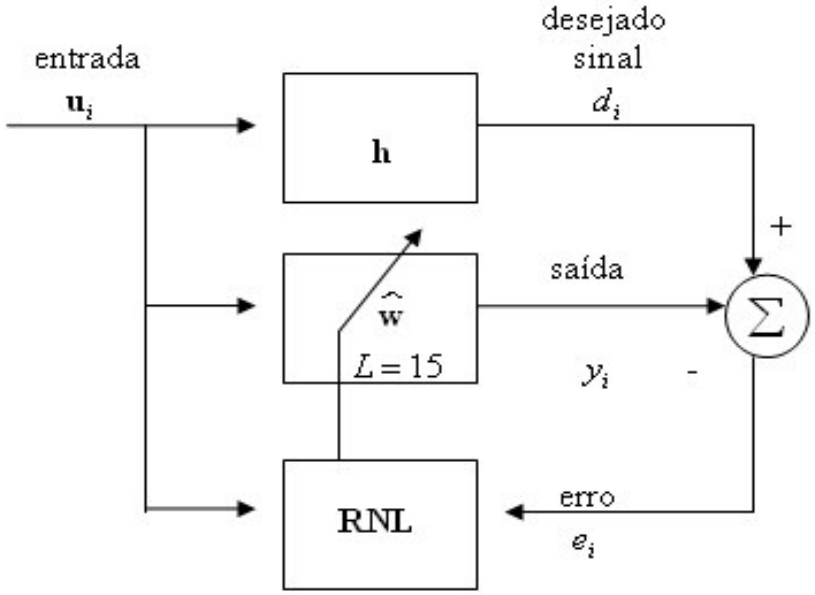

Figura 5.1: Diagrama de blocos de um filtro FIR de tamnho $L = 15$, usando o algoritmo RNL para adaptação dos pesos

o desajuste do novo algoritmo é basicamente o mesmo do *RLS* padrão, com mérito de apresentar um tempo de aprendizagem menor. Vale observar que quanto maior for o valor de M na equação 4.1 esta contante de tempo diminui.

5.4 Conclusões do Capítulo

O algoritmo RNL, conforme proposto, mostrou uma melhora no desempenho quando comparado com o RLS. Melhora esta que mostrou-se dependente da quantidade de termos M, ou seja, ao aumentarmos a quantidade de termos, conseguentemente, aumentando a inclinação da superfície de desempenho, isto pode ser observado na figura 5.2. O algoritmo RNL aumenta a velocidade de convergência dos pesos com basicamente o mesmo desajuste, porém com um tempo de aprendizagem menor.

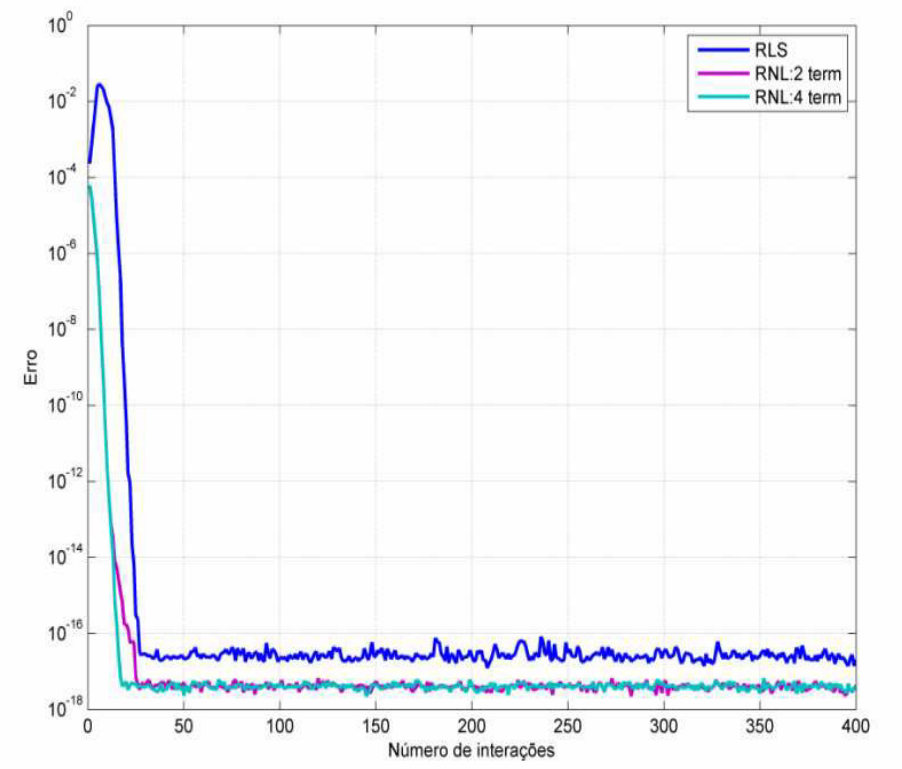

Figura 5.2: Curvas de aprendizagem dos algoritmos RLS e o algoritmo proposto RNL.

Capítulo 6

Conclusões e Proposta de Continuidade

6.1 Conclusões

A utilização de estatística de alta ordem, como uma forma de obtenção de mais informações sobre sinais, tem-se demostrado de grande valia em sistemas adaptativos. Neste trabalho desenvolvemos um filtro adaptativo, inspirado no RLS, onde mostramos uma análise matemática para descrever a aplicação de funções não lineares, pares e contínuas, como critério aplicado sobre o erro. As equações obtidas mostraram-se adequadas e através de simulações obtemos a indicação de sua veracidade.

Nas simulações, o algoritmo RNL mostrou-se mais eficiente quando comparado com o RLS padrão definido em [1]. Esta eficiência acentua-se ao aumentarmos a inclinação da superfície de desempenho.

6.2 Proposta de Continuidade

O desenvolvimento matemático aqui apresentado, foi baseado nas características das superfícies de desempenho geradas pelas não linearidades aplicadas sobre o erro. Baseado nesta idéia alguns tópicos de pesquisa podem ser identificados, tais como:

- Utilização de processos geométricos na determinação de funções não lineares a serem aplicadas como critério sobre o erro;
- Desenvolvimento de equações mais adequadas para o tempo de aprendizagem

e o desajuste;

- Estudos mais aprofundados sobre o tempo de aprendizagem.
- Evolução para o filtro de Kalman.
- Aplicações do algoritmo proposto.

Referências Bibliográficas

[1] S. Haykin, "Adaptive filter theory". Englewood Cliffs, NJ: Pentice-Hall, 1991.

[2] B. Farhang-Boroujeny, "Adaptive Filter Theory and Application". John Wiley e Sons, 1998.

[3] Ljung L., Morf M. and Falconer D., "Fast calculation of gain matrices for recursive estimation schemes", International Journal of Control, Vol 27, No 1, pp 1-19, Jan 1978.

[4] Carayannis G., Manolakis D. and Kalouptsidis N. "A fast sequential algorithm for Least-Square filter and prediction", IEEE Transactions on ASSP, Vol ASSP-31, pp 1392-1402, Dec 1983.

[5] Chansarkar M., Desai U. and Rao B. "Comparison of Approximate RLS algorithm with LMS and RLS algorithms", Proceedings of IEEE Region 10 Conference TENCON-89, Bombay, 1989.

[6] Cioffi J. and Kailath T. "Fast recursive least squares filter for adaptive filtering", IEEE Transactions on ASSP, Vol ASSP-32, pp 304-337, 1984.

[7] A. K. Barros, J. Principe, Y. Takeuchi, C. H. Sales, and N. Ohnishi, "An algorithm based on the even moments of the error,"in Proc. 8th Workshop on Neural Networks for Signal Processing, Toulouse, France, 2003, pp. 879-885

[8] Ewaldo E. C. Santana, Y. Yasuda, Y. Takeuchi, A.K. Barros. "Adaptive Estimation of Impedance Cardiographic Signal by the Sigmoidal Algorithm". Proceedings of the Fifth International Workshop on Biosignal Interpretation, September 6-8, Tokyo Japan. 2005.

[9] E. Walach, B. Widrow, "The Lest Mean Fourth (LMF) Adaptive Algorithm and Its Family". IEEE Transactions on Information Theory No. 2, 1984.