Universidade Federal do Maranhão
Centro de Ciências Exatas e Tecnológica
Programa de Pós-Graduação em Engenharia de Eletricidade

TESE

Um algoritmo não quadrático baseado no RLS estendido

LUÍS FERNANDO COELHO AMARAL

São Luis - MA, Brasil

15 de outubro de 2018

Um algoritmo não quadrático baseado no RLS estendido

Tese de Doutorado submetida à Coordenação do Programa de Pós-Graduação em Engenharia de Eletricidade da Universidade Federal do Maranhão como parte dos requisitos necessários para obtenção do grau de Doutor em Engenharia de Eletricidade na área de Automação e Controle.

LUÍS FERNANDO COELHO AMARAL

Setembro, 2018

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

Coelho Amaral, Luís Fernando.
Um algoritmo não quadrático baseado no RLS estendido / Luís Fernando Coelho Amaral. - 2018.
87 f.

Coorientador(a): Ewaldo Eder Carvalho Santana.
Orientador(a): Allan Kardec D. Barros Filho.
Tese (Doutorado) - Programa de Pós-graduação em Engenharia de Eletricidade/ccet, Universidade Federal do Maranhão, São Luís, 2018.

1. Filtragem adaptativa. 2. Função não quadrática. 3. Taxa de convergência. I. Barros Filho, Allan Kardec D. II. Carvalho Santana, Ewaldo Eder. III. Título.

UM ALGORITMO NÃO QUADRÁTICO BASEADO NO RLS ESTENDIDO

LUÍS FERNANDO COELHO AMARAL

Prof. Dr. Allan Kardec D. Barros Filho - UFMA
(Orientador)

Prof. Dr. Ewaldo Eder Carvalho Santana - UFMA
(Co-Orientador)

Profa. Dra. Áurea Celeste Ribeiro - UEMA
(Membro da Banca Examinadora)

Prof. Dr. Antônio da Silva Silveira - UFPA
(Membro da Banca Examinadora)

Prof. Dr. Fausto Lucena de Oliveira - UNICEUMA
(Membro da Banca Examinadora)

Prof. Dr. João Viana da Fonseca Neto - UFMA
(Membro da Banca Examinadora)

Prof. Dr. Francisco das Chagas Souza - UFMA
(Membro da Banca Examinadora)

AGRADECIMENTOS

Ao professor Allan Kardec D. Barros Filho pela motivação, apoio e dedicação, fundamentais para a orientação deste trabalho. Pelo grande cientista que representa para esta Universidade e pela oportunidade de conhecer aplicações até então desconhecidas no meu mundo acadêmico.

Aos professores Ewaldo Eder Carvalho Santana (Co-Orientador), João Viana Fonseca Neto, Francisco das Chagas Souza, Fausto Lucena de Oliveira, Áurea Celeste Ribeiro e Antônio da Silva Silveira pela participação na banca da Tese.

A todos os meus amigos do PIB pelo companheirismo e amizade, em especial ao colega Luís Cláudio de Oliveira Silva e mais especial ainda, a Marcus Vinícius de Sousa Lopes pelas orientações em programas no MatLab.

A Deus pelas oportunidades que me foram dadas.

A meu Pai (In memorian), sempre.

À toda a minha família pelo carinho, apoio e compreensão ao longo dos anos em que sempre busquei conquistar um projeto acadêmico.

À Cigana Lena, Zé Pelintra, João da Mata e Maria Mulambo.

Resumo

Filtros são utilizados com o objetivo geral de separar elementos diferentes. Quando esses elementos formam sinais elétricos, os filtros são dispositivos que alteram o conteúdo de frequências do sinal de entrada. A fim de remover partes indesejadas (ruídos, interferências) ou separar um sinal de outro, os filtros restringem a passagem de frequências específicas.

Um filtro adaptativo é um filtro cujos coeficientes são ajustados de forma adaptativa, em função de objetivos ou condições variáveis no tempo e traduzidos num sinal de erro. O critério típico prático para a adaptação dos coeficientes do filtro e otimização do seu desempenho é a minimização do valor médio quadrático do sinal de erro, fazendo necessário um algoritmo adaptativo para reger o comportamento do sinal de entrada ou o conhecimento prévio do sinal desejado.

As aplicações de algoritmos adaptativos são importantes em diversas áreas, como telecomunicações, sistemas de controle e outras. O modo como a função objetivo de um algoritmo adaptativo é apresentada pode oferecer infomações importantes sobre o desempenho ou o comportamento do algoritmo.

Em filtragem adaptativa, frequentemente são propostas novas estruturas e novos algoritmos de adaptação que visam acelerar a convergência do erro médio quadrático (MSE, do inglês: *mean squares error*) e/ou diminuir a complexidade computacional, principalmente em aplicações que requerem o uso de um número elevado de coeficientes adaptativos.

Diversos algoritmos para atualização dos coeficientes do filtro adaptativo foram desenvolvidos nos últimos anos. Pode-se citar alguns: o algoritmo LMS (do inglês: *least-mean-squares*) convencional, que possui baixa complexidade computacional, mas seu comportamnto durante a convergência varia de acordo com as características do sinal de entrada, acarretando uma convergência lenta para sinais de entrada correlacionados; o algoritmo RLS (do inglês: *recursive-least-squares*), que possui alta velocidade de convergência, porém elevada complexidade computacional e, em certos casos, instabilidade numérica; o algoritmo LMF (do inglês: *least-mean-fourth*)

que procura minimizar o erro quarto médio, que é uma função do vetor peso convexa.

Existem vários métodos para se derivar algoritmos de filtragem adaptativa, que podem se basear em conceitos estocásticos ou determinísticos, ou até mesmo na formulação matemática de um sistema em um problema de otimização. Apesar da grande diversidade dos algoritmos iterativos que podem resultar da solução de um problema utilizando como função de custo o MSE , a maioria leva a uma resposta que tem relação direta com a resposta dada pelo filtro de Wiener.

Neste trabalho, apresenta-se um algoritmo baseado nas potências pares do erro como função de custo, motivado pelo algoritmo EX-RLS (do inlgês: *extended recursive least squares* exponencialmente ponderado. Simulações foram mostradas, baseadas na convergência e no desajuste comparando alguns do principais algoritmos com o algoritmo proposto.

Palavras-chave: Filtragem adaptativa, função não-quadrática, taxa de convergência, rastreamento no canal de Rayleigh.

Abstract

Filters are used for the general purpose of separating different elements. When these elements form electrical signals, Filters are devices that change the frequency content of the input signal. In order to remove unwanted parts (Noise, interference) or separate one signal from another, the filters restrict the passage of specific frequencies.

An adaptive filter is a filter whose coefficients are adjusted adaptively, in function of objectives or conditions in time and translated into an error signal. The typical practical criterion for adapting coefficients of the filter and optimization of its performance is the minimization of the mean square value of the error signal.

The applications of adaptive algorithms are important in several areas, such as telecommunications, control systems and others. How the objective function of an adaptive algorithm is presented can provide important information about performance or behavior of the algorithm.

In adaptive filtering, new structures and new adaptation algorithms Accelerate the convergence of the mean square error (MSE) and / or decrease the computational complexity, Especially in applications that require the use of a large number of adaptive coefficients.

Several algorithms for updating the adaptive filter shape coefficients developed in recent years. We can mention some: the conventional LMS (Least-Mean-Square) algorithm, which has low complexity But its behavior during convergence varies according to the characteristics of the signal Leading to slow convergence for correlated input signals; The algorithm RLS (Recursive-Least-Square), which has high convergence speed but high complexity computational and, in certain cases, numerical instability; The Least-Mean-Fourth (LMF) algorithm Minimize the average fourth error, which is a function of the convex weight vector.

There are several methods to derive adaptive filtering algorithms, which can be based on concepts stochastic or deterministic, or even in the mathematical formulation of a system in a problem of optimization. In spite of the great diversity

of the iterative algorithms that can result from the solution of a problem using the MSE as a cost function, most lead to a response that has a direct relation with the given response by the Wiener filter.

In this work, we present an algorithm based on the even error, motivated by the exponentially weighted EX-RLS (Extended Recursive Least Squares) algorithm. We will show simulations based on convergence and mismatch comparing the algorithms cited with the proposed algorithm.

Keywords: Adaptive filtering, non-quadratic function, convergence rate, Rayleigh channel tracking.

Lista de Figuras

Lista de Tabelas

Lista de Abreviaturas

MSE (*Mean squares error*) - Erro quadrádito médio.

CLA Combinador linear adaptativo.

FIR (*Finite impulse response*) - Resposta ao impulso finita.

LMS (*Least mean squares*) - Mínimo quadrado médio.

RLS (*Recursive least squares*) - Mínimo quadrado recursivo.

RNQ (*Recursive Non-Quadratic*)- Recursivo não-quadrático.

LMF (*Least mean fourth*) - Mínimo quarto médio.

LLMS (*Linear least mean squares*) - Mínimo quadrado médio linear

EX-RLS (*Extended recursive least squares*) - Mínimo quadrado recursivo estendido.

EX-RNQ (*Extended recursive non quadratic*) - Recursivo não quadrático estendido.

KRLS (*Kernel recursive least squares*) - Mínimo quadrado recursivo kernelizado.

EX-KRLS (*Extended recursive least squares*) - Mínimo quadrado recursivo kernelizado estendido.

Lista de Símbolos

$\mathbf{u}$ vetor de entrada.

$\mathbf{w}$ vetor peso.

L ordem do filtro.

$\mathbf{y}$ vetor de saida.

d sinal desejado.

e sinal de erro.

∇ operador gradiente.

ϵ função de custo no RLS.

$diag\{a, b\}$ matriz diagonal com entradas a e b na diagonal.

$a \oplus b$ $diag\{a, b\}$.

$\mathbf{A}$ matriz da equação de estado no modelo de espaço de estado

φ matriz de auto-correlação do sinal de entrada no RLS.

$\mathbf{n}(i)$ ruido da equação de estado.

$v(i)$ ruido da equação de medida

q_1 elemento da diagonal da matriz de correlação do ruido de estado

q_2 variância do ruído de medida.

$\mathbf{z}$ vetor de correlação cruzada do sinal de entrada com o sinal desejado.

μ passo de adaptação.

ρ fator de ponderação.

λ fator de esquecmento.

tr traço.

J função de custo no RNQ.

ϕ matriz de auto-correlação do vetor de entrada no RNQ.

$\mathbf{g}$ vetor ganho.

$\mathbf{P}$ matriz inversa da matriz de auto-correlação do vetor de entrada.

ξ estimação a priori do erro.

$\mathbf{v}$ vetor desvio.

$\mathbf{K}$ matriz de auto-correlação do vetor desvio.

$\mathbf{\Lambda}$ matriz diagonal.

$\mathbf{U}$ matriz dos dados de entrada.

$\tau_{\mathcal{RNQ}_1}$ tempo de convergência do RNQ nos instantes iniciais.

$\tau_{\mathcal{RNQ}_1}$ tempo de convergência do RNQ após os instantes iniciais.

$\mathcal{M}_{\mathcal{RLS}}$ o desajuste do RLS.

$\mathcal{M}_{\mathcal{RNQ}}$ o desajuste do RNQ.

$\|.\|$ norma euclidiana.

Sumário

Capítulo 1

Introdução

O algoritmo RLS estendido (EX-RLS - Extended Recursive Least Squares) é um caso especial de um algoritmo mais geral chamado Filtro de Kalman [1]. Comparado com RLS padrão e LMS, uma característica distintiva do algoritmo dos mínimos quadrados recursivo estendido é que sua formulação matemática é descrita em termos do conceito de Espaço de Estado. Embora o RLS também represente o conceito de estado, o estado no RLS é invariante no tempo. Considera-se o modelo na forma de espaço de estado com duas equações envolvendo o vetor de estado, conforme modelado por Kalman em 1960. Ao fazê-lo, chega-se ao algoritmo RLS estendido que é o mais adequado para o rastreamento do vetor de estado do modelo geral de espaço de estado linear.

1.1 Modelo matemático de espaço de estado

O vetor de estado, denotado por $\mathbf{x}(i)$, é definido como o conjunto mínimo de dados que é suficiente para descrever unicamente o comportamento dinâmico não forçado do sistema. Em outras palavras, o estado compreende o menor número de dados sobre o passado do sistema que são necessários para prever seu comportamento futuro. Tipicamente, o estado $\mathbf{x}(i)$ é desconhecido e é necessário usar um conjunto de observações para estimá-lo. Em termos matemáticos,

1. **Uma equação de processo (ou de estado)**

$$\mathbf{x}(i+1) = \mathbf{A}\mathbf{x}(i) + \mathbf{n}(i). \tag{1.1}$$

Nesta equação, o vetor $\mathbf{n}(i)$, $L \times 1$, representa o ruído do processo modelado como um ruído branco e média zero cuja matriz de correlação é definida por

$$\mathbb{E}[\mathbf{n}(i)\mathbf{n}^T(j)] = \begin{cases} q_1\mathbf{I}, & i = j \\ 0, & i \neq j \end{cases} . \tag{1.2}$$

A equação (1.1) modela um fenômeno físico estocástico desconhecido denotado pelo vetor de estado $\mathbf{x}(i)$, $L \times 1$, como a saída de um sistema dinâmico linear excitado pelo ruído branco $\mathbf{n}(i)$. O sistema dinâmico é caracterizado unicamente pela conexão de realimentação de duas unidades: a matriz de estado, $L \times L$, denotada por $\mathbf{A}$, e a unidade de memória, $z^{-1}\mathbf{I}$, onde z^{-1} é o atraso de tempo discreto da unidade e $\mathbf{I}$ é a matriz identidade $L \times L$. Em geral $\mathbf{A}$ pode ser variante no tempo.

2. **Uma equação de medida**: Descreve as observações

$$d(i) = \mathbf{u}(i)^T\mathbf{x}(i) + v(i), \tag{1.3}$$

onde $\mathbf{u}(i)$ é conhecido como vetor de medida. O ruído de medida $v(i)$ é modelado como um processo de ruído branco, média zero com variância q_2. A equação de medida (1.3) relaciona a saída observável do sistema $d(i)$ com o estado $\mathbf{x}(i)$. Esta equação tem alguma semelhança com o ajuste de regressão linear em RLS com $\mathbf{u}(i)$ como a entrada e $d(i)$ como a saída. A diferença é que a aplicação entrada-saída $\mathbf{x}(i)$ na equação (1.1) é variante no tempo e governada pela equação de processo (1.3), enquanto que no RLS, é invariante no tempo [2].

É assumido que $\mathbf{x}(1)$, que é o valor inicial do estado, é descorrelacionado tanto com $\mathbf{n}(i)$ quanto $v(i)$ para $i \geq 1$. Os dois ruídos $\mathbf{n}(i)$ e $v(i)$ são estatisticamente independentes. Os parâmetros $\mathbf{A}, q_1$ e q_2 são conhecidos *a priori*. Com todas estas premissas, o algoritmo EX-RLS é requerido para resolver o seguinte problema:

Usa-se as observações inteiras $\{\mathbf{u}(1), d(1)\}, \{\mathbf{u}(2), d(2)\}, ..., \{\mathbf{u}(i), d(i)\}$, para achar o estimador mínimo médio quadrado do estado $\mathbf{x}(i+1)$. Usa-se $\mathbf{w}(i-1)$ para denotar a estimativa ótima de $\mathbf{x}(i)$ dadas as observações começando no tempo $j = 1$ até o tempo $j = i-1$, isto é, $\{\mathbf{u}(1), d(1)\}, \{\mathbf{u}(2), d(2)\}, ..., \{\mathbf{u}(i-1), d(i-1)\}$, e $\mathbf{w}(i)$ para denotar a estimativa ótima de $\mathbf{x}(i+1)$ dadas as observações começando no tempo $j = 1$ até $j = i$. Como no RLS, definimos o processo de inovação associado com $d(i)$ como

$$e(i) = d(i) - \mathbf{w}^T(i-1)\mathbf{u}(i), \tag{1.4}$$

onde $e(i)$ é o erro de predição para o $d(i)$ observado e representa a nova informação em $d(i)$. A variância do processo de inovação $e(i)$ é definida por

$$r(i) = \mathbb{E}[e(i)^2] = \mathbf{u}(i)^T\mathbf{P}(i-1)\mathbf{u}(i) + q_2, \tag{1.5}$$

onde $\mathbf{P}(i-1)$ é a matriz de correlação do erro de estado. Para prosseguir, precisa-se introduzir um conceito importante chamado Ganho de Kalman ou simplesmente vetor ganho, e uma relação fundamental entre a estimativa ótima atual e a estimativa ótima anterior

$$\underbrace{\mathbf{w}(i)}_{estimativa\ atual} = \mathbf{A} \cdot \underbrace{\mathbf{w}(i-1)}_{estimativa\ anterior} + \underbrace{\mathbf{K}(i)e(i)}_{ajuste}. \tag{1.6}$$

Esta equação mostra que pode-se calcular a estimativa mínima média quadrática $\mathbf{w}(i)$, pré-multiplicado pela matriz de estado $\mathbf{A}$, um termo de correção $\mathbf{K}(i)e(i)$. O termo de correção é igual ao erro de predição $e(i)$ pré-multiplicado pelo vetor de ganho $\mathbf{K}(i)$. Além disso, podemos expressar o vetor de ganho $\mathbf{K}(i)$ como

$$\mathbf{K}(i) = \mathbf{A}\mathbf{P}(i-1)\mathbf{u}(i)/r(i). \tag{1.7}$$

Portanto, inicializando $\mathbf{w}(0) = 0$ e $\mathbf{P}(0) = \lambda\mathbf{I}$, pode-se calcular $r(1), \mathbf{K}(1)$ e então atualizar $\mathbf{w}(1), \mathbf{P}(1)$ com $\{\mathbf{u}(1), d(1)\}$ fornecidos. Aqui, λ é um número positivo chamado parâmetro de regularização.

Como mencionado, o EX-RLS é um caso especial de um filtro de Kalman que é usado numa ampla gama de aplicações de engenharia aeroespacial, por exemplo, e um dos fundamentos na teoria de controle moderno e sistemas de controle. Além disso, EX-RLS fornece um quadro unificador para a derivação da família de filtros

RLS usando modelo de espaço de estado. Por exemplo, tomando $\mathbf{A} = \beta^{-1/2}\mathbf{I}$ e ignorando $\mathbf{n}(i)$, temos o algoritmo RLS exponencialmente ponderado. Além disso, tomando $\beta = 1$, temos o próprio algoritmo RLS. Apesar da estrutura simples dos filtros adaptativos lineares (e provavelmente por causa deles), eles gozam de ampla aplicabilidade e sucessos em diversos campos como comunicação, controle, radar, sonar, sismologia e engenharia biomédica, entre outros. A teoria dos filtros adaptativos lineares atingiu um estágio de desenvolvimento altamente maduro. No entanto, o mesmo não pode ser dito sobre filtros adaptativos não-lineares.

1.2 Objetivo geral

A formulação de um sistema adaptativo como um problema de otimização oferece uma nova visão sobre a função de custo dos algoritmos adaptativos, além de possibilitar a inclusão de restrições para incluir alguma caracteristica desejada, como, por exemplo, tornar o algoritmo mais robusto. Este trabalho de Tese tem como objetivo apresentar um algoritmo adaptativo capaz de acompanhar as variações das características estatísticas dos sinais em um meio não estacionário tomando como base os algoritmos EX-RLS [3] e o RNQ (Recursive Non Quadratic) [4]. A combinação destes dois algoritmos será, o que chamaremos de algoritmo EX-RNQ (Extended Recursive Non Quadratic). A maneira como a função custo é utilizada para derivar o algoritmo proposto, as formas de aproximar as estatísticas dos sinais envolvidos e o método do desenvolvimento do algoritmo, entre outras coisas, irão definir o algoritmo e determinar seu comportamento para diferentes tipos de sinais de entrada. Além disso, realizamos um estudo dos algoritmos existentes, como o RLS que é padrão para o desenvolvimento inicial e depois comparamos com os algoritmos recursivos nas formas estendida e kernelizada [2, 3], para efeito de comparação, em termos de convergência e desajuste.

1.2.1 Objetivos específicos

- Definir equações recursivas do algoritmo RNQ, para gerar um novo algoritmo tipo recursivo estendido;
- Verificar o desempenho dos algoritmos estendidos recursivos e kernelizados

com o algoritmo proposto em ambientes não estacionários.

1.3 Estado da arte

A maior parte dos sinais de interesse na Engenharia é não estacionária; em outras palavras, o conteúdo espectral de sinais determinísticos e as estatísticas de processos estocásticos são variantes no tempo, e novas estruturas e algoritmos são frequentemente propostos para acelerar a convergência do erro quadrático médio (MSE).

Alguns algoritmos já foram propostos para melhorar o desempenho das variações das características estatísticas dos sinais em ambientes não estacionários, como é o caso dos algoritmos recursivos estendidos, incluindo suas versões do kernel.

Liu et al. [3] apresentaram uma versão kernelizada do algoritmo recursivo dos mínimos quadrados estendidos (EX-RLS - extended recursive least squares), que implementa um modelo de referência não linear no espaço de entrada. Zhu et al. [18] propuseram um novo algoritmo de mínimos quadrados recursivos estendidos do kernel (EX-KRLS) combinando o algoritmo de mínimos quadrados recursivos do kernel (KRLS) e o filtro de Kalman ou suas extensões para estimar ou prever sinais. Li e Príncipe [19] apresentaram um novo algoritmo de filtragem adaptativa recorrente ao kernel baseado no modelo ARMA (autoregressive-moving-average), que é treinado com gradiente descendente estocástico recorrente no kernel reprodutivo de Hilbert.

Outros algoritmos recursivos estendidos vêm ganhando atenção na literatura especializada com base no gradiente estocástico, por exemplo, a teoria de identificação multi-inovação é muito importante na identificação do sistema, a fim de melhorar a estimação de parâmetros de algoritmos baseados em gradientes. Ding et al. [20] apresentaram um algoritmo de gradiente estocástico estendido multi-inovação baseado em modelo auxiliar. Pan et al. [21] propuseram um algoritmo de gradiente estocástico estendido baseado em filtragem e com o objetivo de melhorar a precisão da estimativa de parâmetros. Liu et al. [22] derivaram o algoritmo de gradiente estocástico estendido de múltiplas inovações para modelos ARMA controlados, expandindo a inovação escalar para um vetor de inovação. Wang e Ding

[23] enfocaram o problema de estimação de parâmetros dos sistemas Box-Jenkins, usando a multi-inovação na teoria de identificação. Recentemente, estudos baseados no algoritmo do gradiente estocástico de multi-inovação têm sido propostos para resolver problemas de estimação de parâmetros não-lineares de Hammerstein [25,26] com ruído colorido [24]. Todas essas abordagens são relevantes no processamento de sinais, identificação de sistemas e controle adaptativo.

1.4 Organização do texto

Este trabalho está organizado em cinco capítulos.

No Capítulo 2, faremos o desenvolvimento dos algoritmos RLS, por exemplo, [1,5] e RNQ [4] como motivação para o desenvolvimento do algoritmo proposto, que será visto no Capítulo 4, e uma análise comparativa, *a posteriori*, do desempenho dos algoritmos que serão comparados, foram baseados no livro [2] e no artigo [3].

No Capítulo 3, apresentamos o algoritmo RLS Estendido na sua forma mais geral [7]. Faremos um desenvolvimento determinístico do tema, porém trataremos também o assunto de forma estocástica, que é a forma equivalente [7] do problema determinístico.

No Capítulo 4, iremos desenvolver o algoritmo RLS Estendido em uma forma particular, [7] usando o modelo de espaço de estado, citado no Capítulo 1, como um caso também particular do filtro de Kalman [2]. Mostra-se o algoritmo proposto (algoritmo não quadrático baseado no RLS Estendido) baseado nas potências pares do erro [4]. Apresentam-se também as simulações feitas, baseadas, principalmente, na aplicação ao rastreamento de canais de Rayleigh.

Finalmente no Capítulo 5, apresentamos a conclusão e os trabalhos futuros.

Capítulo 2

O algoritmo dos mínimos quadrados recursivos

2.1 Introdução

Para os algoritmos baseados no gradiente estocástico, a função custo é definida como sendo o valor esperado da diferença ao quadrado entre o sinal desejado e a saída do filtro adaptativo. Tais algoritmos têm como principal vantagem a baixa complexidade computacional. Talvez, o algoritmo mais popular nessa linha seja o algoritmo LMS (Least Mean Squares). No caso dos algoritmos de mínimos quadrados, a função custo é definida como sendo a soma dos quadrados dos erros ponderados. Estes algoritmos têm como vantagem a baixa sensibilidade do sinal de entrada e uma maior velocidade de convergência comparado aos algoritmos de gradiente estocástico. Tem como desvantagem uma complexidade computacional elevada. O algoritmo mais popular nessa linha é o RLS (Recursive Least Squares). Faremos neste capítulo o desenvolvimento do algoritmo RLS baseado na Tese [4], que por sua vez tem como fonte principal o livro [5].

2.2 Dedução do algoritmo RLS

No algoritmo RLS o fator ponderação ρ_i é escolhido como sendo

$$\rho_i = \lambda^{n-i}, \tag{2.1}$$

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

O método dos mínimos quadrados padrão anteriormente há Equação (2.1) corresponde ao caso em que $\lambda = 1$. O parâmetro λ é denominado fator de esquecimento. Claramente quando $\lambda < 1$, o fator de ponderação definido em (2.1) dá um peso maior às amostras recentes das estimativas do erro (e assim, às 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.

A função de custo a ser minimizada na dedução do algoritmo RLS será

$$\varepsilon_n = \sum_{i=1}^{n} \lambda^{n-i}. \left|e_i\right|^2. \tag{2.2}$$

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

$$\hat{\mathbf{w}}_n = \varphi_n^{-1}\mathbf{z}_n, \tag{2.3}$$

sendo

$$\varphi_n = \sum_{i=1}^{n} \lambda^{n-i}\mathbf{u}_i\mathbf{u}_i^T, \tag{2.4}$$

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

$$\mathbf{z}_n = \sum_{i=1}^{n} \lambda^{n-i}\mathbf{u}_i d_i, \tag{2.5}$$

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

Expandindo a equação (2.4) e isolando o termo $i = n$, temos

$$\begin{aligned} \varphi_n &= \lambda\left[\sum_{i=1}^{n-1} \lambda^{n-1-i}.\mathbf{u}_i\mathbf{u}_i^T\right] + \mathbf{u}_n\mathbf{u}_n^T \\ \varphi_n &= \lambda\left[\varphi_{n-1}\right] + \mathbf{u}_n\mathbf{u}_n^T. \end{aligned} \tag{2.6}$$

Analogamente podemos escrever a equação (2.5) da seguinte forma

$$\mathbf{z}_n = \lambda\left[\mathbf{z}_{n-1}\right] + \mathbf{u}_n d_n. \tag{2.7}$$

2.2.1 O lema de inversão de matrizes

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

$$\mathbf{A} = \mathbf{B}^{-1} + \mathbf{C}\mathbf{D}^{-1}\mathbf{C}^{T}. \tag{2.8}$$

De (2.8) obtemos

$$\mathbf{A}^{-1} = \mathbf{B} - \mathbf{B}\mathbf{C}\left(\mathbf{D} + \mathbf{C}^{T}\mathbf{B}\mathbf{C}\right)^{-1}\mathbf{C}^{T}\mathbf{B}, \tag{2.9}$$

sendo $\mathbf{D}$ uma 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 (2.8) por (2.9). Na próxima seção mostraremos como o lema de inversão de matrizes pode ser aplicado para obter uma equação recursiva.

2.2.2 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 (2.4). Primeiramente faremos as seguintes identificações:

$$\begin{cases} \mathbf{A} = \varphi_n \\ \mathbf{B}^{-1} = \lambda\varphi_{n-1} \rightarrow \mathbf{B} = \lambda^{-1}\varphi_{n-1}^{-1} \\ \mathbf{C} = \mathbf{u}_n \\ \mathbf{D} = \mathbf{I} \end{cases}. \tag{2.10}$$

Substituindo as definições (2.10) na equação (2.9), obtemos a relação de recursividade da matriz φ_n , ou seja:

$$\varphi_n^{-1} = \lambda^{-1}\varphi_{n-1}^{-1} - \frac{\lambda^{-2}\varphi_{n-1}^{-1}\mathbf{u}_n\mathbf{u}_n^{T}\varphi_{n-1}^{-1}}{1 + \lambda^{-1}\mathbf{u}_n^{T}\varphi_{n-1}^{-1}\mathbf{u}_n}. \tag{2.11}$$

Por conveniência computacional podemos escrever as seguintes igualdades:

$$\mathbf{P}_n = \varphi_n^{-1}, \tag{2.12}$$

$$\mathbf{k}_n = \frac{\lambda^{-1}\mathbf{P}_{n-1}\mathbf{u}_n}{1 + \lambda^{-1}\mathbf{u}_n^{T}\mathbf{P}_{n-1}\mathbf{u}_n}. \tag{2.13}$$

Podemos reescrever a equação (2.11) como segue:

$$\mathbf{P}_n = \lambda^{-1}\mathbf{P}_{n-1} - \lambda^{-1}\mathbf{k}_n\mathbf{u}_n^T\mathbf{P}_{n-1}, \tag{2.14}$$

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 (2.13), temos:

$$\begin{aligned} \mathbf{k}_n + \mathbf{k}_n\lambda^{-1}\mathbf{u}_n^T\mathbf{P}_{n-1}\mathbf{u}_n &= \lambda^{-1}\mathbf{P}_{n-1}\mathbf{u}_n \\ \mathbf{k}_n &= \left[\lambda^{-1}\mathbf{P}_{n-1} - \mathbf{k}\lambda^{-1}\mathbf{u}_n^T\mathbf{P}_{n-1}\right]\mathbf{u}_n. \end{aligned} \tag{2.15}$$

Substituindo a equação (2.14) em (2.15) seque-se:

$$\begin{aligned} \mathbf{k}_n &= \mathbf{P}_n\mathbf{u}_n \\ \mathbf{k}_n &= \varphi_n^{-1}\mathbf{u}_n. \end{aligned} \tag{2.16}$$

2.2.3 Atualização do vetor peso

No desenvolvimento de uma equação recursiva, utilizaremos as equações (2.3), (2.7) e (2.12) para expressar a estimativa do mínimo quadrado do vetor peso, $\hat{\mathbf{w}}_n$, no instante n, como segue:

$$\begin{aligned} \hat{\mathbf{w}}_n &= \mathbf{P}_n\left[\lambda\mathbf{z}_{n-1} + \mathbf{u}_n\right] \\ \hat{\mathbf{w}}_n &= \lambda\mathbf{P}_n\mathbf{z}_{n-1} + \mathbf{P}_n\mathbf{u}_n. \end{aligned} \tag{2.17}$$

Substituindo a equação (2.14) na equação (2.17), temos:

$$\hat{\mathbf{w}}_n = \mathbf{P}_{n-1}\mathbf{z}_{n-1} - \mathbf{k}_n\mathbf{u}_n\mathbf{P}_{n-1}\mathbf{z}_{n-1} + \mathbf{P}_n\mathbf{u}_n d_n. \tag{2.18}$$

Das equações (2.16) e (2.18), segue-se:

$$\hat{\mathbf{w}}_n = \hat{\mathbf{w}}_{n-1} - \mathbf{k}_n\epsilon_n, \tag{2.19}$$

sendo ϵ_n a estimativa do erro a priori definida por:

$$\epsilon_n = d_n - \mathbf{u}_n^T \hat{\mathbf{w}}_{n-1}. \tag{2.20}$$

2.3 Convergência do algoritmo *RLS*

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

$$d_n = \mathbf{w}_*^{\mathbf{T}} \mathbf{u}_n + e_n, \tag{2.21}$$

sendo $\mathbf{w}_*$ o vetor peso do regressor, $\mathbf{u}_n$ o vetor entrada, e_n o ruído 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

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

De (2.3) e (2.5) obtemos

$$\hat{\mathbf{w}}_n = \varphi_n^{-1} \sum_{i=1}^{n} \lambda^{n-i} \mathbf{u}_i d_i. \tag{2.22}$$

Substituindo (2.21) em (2.22) e usando (2.4), temos

$$\hat{\mathbf{w}}_n = \hat{\mathbf{w}}_* + \varphi_n^{-1} \sum_{i=1}^{n} \lambda^{n-i} \mathbf{u}_i e_i. \tag{2.23}$$

Aplicando o operador esperança em ambos os membros da equação (2.23) e reconhecemos do princípio de ortogonalidade que todos os elementos do vetor $\mathbf{u}_n$ são ortogonais ao erro e_n, obtemos

$$E[\hat{\mathbf{w}}_n] = \hat{\mathbf{w}}_*. \tag{2.24}$$

2.3.2 Matriz de correlação do vetor desvio

Definimos o vetor de desvio como

$$\mathbf{v}_n = \hat{\mathbf{w}}_n - \mathbf{w}_*. \tag{2.25}$$

De (2.23), temos,

$$\mathbf{v}_n = \varphi_n^{-1} \sum_{i=1}^{n} \lambda^{n-i} \mathbf{u}_i e_i. \tag{2.26}$$

Definimos a matriz de correlação do vetor desvio da seguinte forma,

$$\mathbf{K}_n = E[\mathbf{v}_n \mathbf{v}_n^T]. \tag{2.27}$$

Substituindo (2.26) em (2.27) e notando que $[\varphi_n^{-1}]^T = \varphi_n^{-1}$ e $[\lambda^{n-i}]^T = \lambda^{n-i}$, obtemos,

$$\mathbf{K}_n = E\left[\left(\varphi_n^{-1} \sum_{i=1}^{n} \lambda^{n-i}.\mathbf{u}_i e_i\right)\left(\sum_{i=1}^{n} \lambda^{n-i} e_i^T \mathbf{u}_i^T\right)\varphi_n^{-1}\right]. \tag{2.28}$$

Conforme [5], para o rigoroso cálculo da equação (2.28) devemos fazer as seguintes hipóteses:

1. O vetor de entrada $\mathbf{u}_i$ possui amostras de um processo estocástico. 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 (2.4) 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,

$$\varphi_n \approx \frac{1-\lambda^n}{1-\lambda}\mathbf{R}, \tag{2.29}$$

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

Substituindo (2.29) em (2.28) e notando que $E[e_n e_n^T] = \sigma^2$, obtemos,

$$\begin{aligned}\mathbf{K}_n &= \sigma^2\left(\frac{1-\lambda^n}{1-\lambda}\right)^2 \cdot \left(\frac{1-\lambda^{2n}}{1-\lambda^2}\right)\mathbf{R}^{-1} \\ &= \sigma^2\left(\frac{1-\lambda}{1+\lambda}\right)\cdot\left(\frac{1+\lambda^n}{1-\lambda^n}\right)\mathbf{R}^{-1}.\end{aligned} \tag{2.30}$$

2.3.3 Curva de apredizado do algoritmo *RLS*

Para algoritmo RLS, é conveniente usar o erro ϵ_n para definir o MSE, assim podemos expressar a curva de aprendizagem do algoritmo RLS em termos do erro a priori como,

$$\xi_n = E[\epsilon_n^2], \tag{2.31}$$

De (2.25) e (2.21) podemos escrever o erro a priori da seguinte forma:

$$\epsilon_n = e_n - \mathbf{v}_{n-1}^T\mathbf{u}_n. \tag{2.32}$$

Substituindo (2.32) em (2.31), obtemos,

$$\begin{aligned}\xi_n &= E\left[e_n^2\right] + E\left[\mathbf{u}_n^T\mathbf{v}_{n-1}\mathbf{v}_{n-1}^T\mathbf{u}_n\right] \\ &- E\left[\mathbf{v}_{n-1}^T\mathbf{u}_n e_n\right] - E\left[e_n\mathbf{u}_n^T\mathbf{v}_{n-1}\right].\end{aligned} \tag{2.33}$$

O primeiro valor esperado do lado direito da equação (2.33) 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 assumido como sendo derivado de um processo

estacionário de amplo sentido de média zero. Consequentemente, podemos usar esta independência estatística junto com os resultados conhecidos de álgebra matricial para expressar o segundo valor esperado do lado direito da equação (2.33), como segue-se,

$$\begin{aligned} E\left[\mathbf{u}_n^T \mathbf{v}_{n-1} \mathbf{v}_{n-1}^T \mathbf{u}_n\right] &= E\left[tr\left\{\mathbf{u}_n^T \mathbf{v}_{n-1} \mathbf{v}_{n-1}^T \mathbf{u}_n\right\}\right] \\ &= E\left[tr\left\{\mathbf{u}_n^T \mathbf{u}_n \mathbf{v}_{n-1} \mathbf{v}_{n-1}^T\right\}\right] \\ &= tr\left\{E\left[\mathbf{u}_n^T \mathbf{u}_n\right] E\left[\mathbf{v}_{n-1} \mathbf{v}_{n-1}^T\right]\right\} \\ &= tr\left\{\mathbf{R}\mathbf{K}_{n-1}\right\}, \end{aligned} \tag{2.34}$$

sendo que na última linha utilizamos as definições de médias da matriz de correlação $\mathbf{R} = E[\mathbf{u}_n \mathbf{u}_n^T]$ 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 simples manipulação da equação (2.21). 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 (2.33) é zero. Assim,

$$E\left[\mathbf{v}_{n-1}^T \mathbf{u}_n e_n\right] = E\left[\mathbf{v}_{n-1}^T\right] . E\left[\mathbf{u}_n e_n\right]. \tag{2.35}$$

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

$$E\left[\mathbf{v}_{n-1}^T \mathbf{u}_n e_n\right] = 0. \tag{2.36}$$

3. O quarto valor esperado do lado direito da equação (2.33) tem a mesma forma matemática considerada no item 2. Entretanto podemos dizer que este valor esperado é igual a zero,

$$E\left[e_n \mathbf{u}_n^T \mathbf{v}_{n-1}\right] = 0. \tag{2.37}$$

Assim, reconhecendo que $E[e_n^2] = \xi_{mim}$, e usando os resultados das equações (2.34) até (2.37) em (2.33), obtemos a seguinte fórmula para o erro quadrático médio no algoritmo *RLS*:

$$\xi_n = \xi_{min} + tr\left\{\mathbf{R}\mathbf{K}_{n-1}\right\}. \tag{2.38}$$

sendo ξ_{min} o mínimo MSE do filtro encontrado quando uma estimativa perfeita de $\mathbf{w}_*$ é calculada. Substituindo (2.30) em (2.38) obtemos

$$\xi_n = \xi_{min} + \left(\frac{1-\lambda}{1+\lambda}\right) \cdot \left(\frac{1+\lambda^{n-1}}{1-\lambda^{n-1}}\right) \cdot L\xi_{min}. \tag{2.39}$$

Este resultado descreve a curva de aprendizagem do algoritmo *RLS*.

2.3.4 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 (2.39) 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,

$$\lambda^n = e^{\frac{-n}{\tau_{\mathcal{RLS}}}}. \tag{2.40}$$

Resolvendo (2.40) para $\tau_{\mathcal{RLS}}$ obtemos,

$$\tau_{\mathcal{RLS}} = -\frac{1}{\ln \lambda}. \tag{2.41}$$

Para simplificar (2.41) usaremos a seguinte aproximação,

$$\ln \lambda = \ln(1 - (1 - \lambda)) \approx -(1 - \lambda). \tag{2.42}$$

Substituindo (2.42) em (2.41) temos,

$$\tau_{\mathcal{RLS}} \approx \frac{1}{(1 - \lambda)}. \tag{2.43}$$

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

2.3.5 Excesso do erro quadrático médio e o desajuste

Sabendo que o erro quadrático médio (ξ_n) é maior que o erro quadrático médio mínimo (ξ_{min}) temos, então, um excesso no erro final e podemos definir o excesso do erro quadrático médio, $ExcessoMSE$, como a diferença entre o erro quadrático médio atual e o erro quadrático médio mínimo [5].

Usando (2.39) obtemos,

$$ExcessoMSE = \left(\frac{1 - \lambda}{1 + \lambda}\right) \cdot L\xi_{min}. \tag{2.44}$$

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

$$\mathcal{M}_{\mathcal{RLS}} = \frac{ExcessoMSE}{\xi_{min}}. \tag{2.45}$$

Substituido (2.44) em (2.45) temos

$$\mathcal{M}_{\mathcal{RLS}} = \left(\frac{1 - \lambda}{1 + \lambda}\right) \cdot L, \tag{2.46}$$

sendo L o tamanho do filtro.

2.4 Algoritmo RNQ baseado em uma única potência par do erro

Nesta seção desenvolveremos um algoritmo adaptativo baseado em uma potência par do erro, inspirado no algoritmo RLS padrão. Chamaremos esse novo algoritmo

de **recursivo não-quadrático** (RNQ), para o qual escolhemos uma função baseada em uma única potência par do erro como critério a ser minimizado. O nosso objetivo é determinar um algoritmo que ajuste os pesos do combinador linear adaptativo 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 ótimo quando comparado com o algoritmo RLS.

2.5 Dedução do algoritmo

A estrutura básica de um filtro adaptativo é composta por um sinal desejado d_i, um vetor de entrada $\mathbf{u}_i = [u_i, u_{i-1}, ..., u_{i-L+1}]^T$, e o erro e_i, usado para atualizar o vetor peso $\mathbf{w}_n = [w_{0,n}, w_{1,n}, ..., w_{L-1,n}]^T$. Devemos recuperar d_i estimando o sinal de saída $y_i = \mathbf{w}_n^T \mathbf{u}_i$, depois de calcular o erro $e_i = d_i - y_i$, sendo n o número de iteração, $1 \leq i \leq n$ e L a ordem do filtro.

Para desenvolver o algoritmo RNQ baseado em uma potência par do erro, utilizamos como função de custo uma função par, contínua e simétrica, representada por

$$J_n = \sum_{i=1}^{n} \Big\{ \lambda^{n-i} \left[e_i\right]^{2j} \Big\}, \tag{2.47}$$

sendo j e n inteiros positivos, com $j > 1$ e $0 << \lambda < 1$ fator peso exponencial ou fator de esquecimento. Podemos observar que para $j = 1$ obtemos o mesmo critério (função de custo) utilizado no algoritmo RLS padrão, deduzido no capítulo anterior.

Objetivando encontrar o peso ótimo, devemos calcular o gradiente da função J_n. Assim, derivando a Equação (2.47) em relação a $\mathbf{w}$, obtemos,

$$\begin{aligned} \nabla J_n &= -\sum_{i=1}^{n} \Big\{ \lambda^{n-i} \mathbf{u}_i 2j \left[e_i\right]^{2j-1} \Big\} \\ &= -\sum_{i=1}^{n} \Big\{ \lambda^{n-i} \mathbf{u}_i 2j \left[d_i - \mathbf{w}_n^T \mathbf{u}_i\right]^{2j-1} \Big\}. \end{aligned} \tag{2.48}$$

Desenvolvendo o binômio $\left[d_i - \mathbf{w}_n^T \mathbf{u}_i\right]^{2j-1}$, e negligenciando as potências de alta ordem, podemos reescrever a Equação (2.48) como,

$$\begin{aligned}\nabla J_n &\approx -\sum_{i=1}^{n} \lambda^{n-i}\mathbf{u}_i 2j \left\{d_i^{2j-1} - (2j-1)\, d_i^{2j-2}\left(\mathbf{w}_n^T\mathbf{u}_i\right)\right\} \\ &\approx -2j\left[\sum_{i=1}^{n}\left[\lambda^{n-i}d_i^{2j-1}\mathbf{u}_i\right] - (2j-1)\sum_{i=1}^{n}\left[\lambda^{n-i}d_i^{2j-2}\mathbf{u}_i\mathbf{u}_i^T\right]\mathbf{w_n}\right], \quad (2.49)\end{aligned}$$

sendo o vetor de correlação cruzada, $\mathbf{z}_n$, entre as entradas e a resposta desejada de dimensão $L \times L$, definido pelo primeiro termo do lado direito da Equação (2.49), isto é,

$$\mathbf{z}_n = \sum_{i=1}^{n}\left[\lambda^{n-i}d_i^{2j-1}.\mathbf{u}_i\right]. \quad (2.50)$$

E a matriz de correlação do vetor de entrada de dimensão $L \times L$ definida pelo segundo termo da Equação (2.49),

$$\mathbf{\Phi}_n = (2j-1)\sum_{i=1}^{n}\left[\lambda^{n-i}d_i^{2j-2}\mathbf{u}_i\mathbf{u}_i^T\right]. \quad (2.51)$$

Dessa forma, podemos reescrever a Equação (2.49) como,

$$\nabla J_n \approx -2j\left[\mathbf{z_n} - \mathbf{\Phi_n}.\mathbf{w_n}\right]. \quad (2.52)$$

Assim, o valor ótimo do vetor peso, $\hat{\mathbf{w}}_n$, para que a função da Equação (2.47) atinja o valor mínimo é definido pela equação normal escrita em forma matricial

$$\hat{\mathbf{w}}_n = \mathbf{\Phi_n^{-1}}\mathbf{z_n}. \quad (2.53)$$

Agora, para encontrarmos uma regra de atualização para o peso $\mathbf{w}$, faremos o seguinte desenvolvimento:

Primeiramente, isolando o termo correspondente a $i = n$ da Equação (2.51), podemos escrever

$$\mathbf{\Phi_n} = \lambda\mathbf{\Phi}_{n-1} + \left[(2j-1)\,.d_n^{2j-2}\right]\mathbf{u}_n.\mathbf{u}_n^T. \quad (2.54)$$

Analogamente podemos reescrever a Equação (2.50) como,

$$\mathbf{z}_n = \lambda \mathbf{z}_{n-1} + \left[d_n^{2j-1}\right] \mathbf{u}_n. \tag{2.55}$$

A dedução das Equações (2.54) e (2.55) foi feita com o intuito de facilitar o cálculo da inversa da matriz de correlação $\mathbf{\Phi_n}$, que é necessário de acordo com a Equação (2.53).

Como a matriz de correlação $\mathbf{\Phi_n}$, é positivamente definida e não singular, podemos aplicar o lema de inversão de matrizes para equação recursiva (2.51), onde faremos as seguintes identificações,

$$\begin{aligned} \mathbf{A} &= \mathbf{\Phi}_n \\ \mathbf{B}^{-1} &= \lambda \mathbf{\Phi}_{n-1} \\ \mathbf{C} &= \mathbf{u}_n \\ \mathbf{D}^{-1} &= (2j-1) d_n^{2j-2}. \end{aligned} \tag{2.56}$$

Substituindo as definições (2.56) na Equação (2.9) do lema de inversão de matrizes, obtemos a relação de recursividade da matriz $\mathbf{\Phi}_n$, ou seja,

$$\mathbf{\Phi}_n^{-1} = \lambda^{-1}\mathbf{\Phi}_{n-1}^{-1} - \frac{\lambda^{-1}\mathbf{\Phi}_{n-1}^{-1}\mathbf{u}_n\mathbf{u}_n^T\lambda^{-1}\mathbf{\Phi}_{n-1}^{-1}}{((2j-1)d_n^{2j-2})^{-1} + \lambda^{-1}\mathbf{u}_n^T\mathbf{\Phi}_{n-1}^{-1}\mathbf{u}_n}. \tag{2.57}$$

Por conveniência computacional podemos escrever as seguintes igualdades,

$$\mathbf{P}_n = \mathbf{\Phi}_n^{-1} \tag{2.58}$$

e

$$\mathbf{g}_n = \frac{\lambda^{-1}\mathbf{P}_{n-1}\mathbf{u}_n}{((2j-1)d_n^{2j-2})^{-1} + \lambda^{-1}\mathbf{u}_n^T\mathbf{P}_{n-1}\mathbf{u}_n}. \tag{2.59}$$

Assim, podemos reescrever a Equação (2.57) como,

$$\mathbf{P_n} = \lambda^{-1}\mathbf{P_{n-1}} - \lambda^{-1}\mathbf{g_n}\mathbf{u_n^T}\mathbf{P_{n-1}}, \tag{2.60}$$

sendo $\mathbf{P_n}$ a inversa da matriz de auto-correlação de dimensão $L \times L$ e $\mathbf{g}_n$ o vetor ganho de dimensão $L \times 1$.

Para facilitar futuras manipulações matemáticas, nós reorganizamos a Equação (2.59) como,

$$((2j-1)d_n^{2j-2})^{-1}\mathbf{g}_n = \mathbf{P}_n\mathbf{u}_n. \tag{2.61}$$

2.5.1 Atualização do vetor peso

No desenvolvimento de uma equação recursiva utilizaremos as Equações (2.53), (2.55) e (2.58) para determinar a regra de atualização do vetor peso $\mathbf{w}_n$. Assim,

$$\begin{aligned} \mathbf{w}_n &= \mathbf{\Phi}_n^{-1}\mathbf{z}_n \\ &= \lambda\mathbf{P}_n\mathbf{z}_{n-1} + d_n^{2j-1}\mathbf{P}_n\mathbf{u}_n. \end{aligned} \tag{2.62}$$

Substituindo a Equação (2.60) no primeiro termo do lado direito da Equação (2.62) obtemos,

$$\begin{aligned} \mathbf{w}_n &= \mathbf{P}_{n-1}\mathbf{z}_{n-1} - \mathbf{g}_n\mathbf{u}_n^T\mathbf{P}_{n-1}\mathbf{z}_{n-1} + d_n^{2j-1}\mathbf{P}_n\mathbf{u}_n \\ &= \mathbf{w}_{n-1} - \mathbf{g}_n\mathbf{u}_n^T\mathbf{w}_{n-1} + d_n^{2j-1}\mathbf{P}_n\mathbf{u}_n. \end{aligned} \tag{2.63}$$

Usando o fato de que $\mathbf{P}_n\mathbf{u}_n$ é uma fatoração de $\mathbf{g}_n$, como mostrado na Equação (2.61), obtemos a seguinte equação,

$$\begin{aligned} \mathbf{w}_n &= \mathbf{w}_{n-1} - \mathbf{g}_n\mathbf{u}_n^T\mathbf{w}_{n-1} + \frac{d_n}{(2j-1)}\mathbf{g}_n \\ &= \mathbf{w}_{n-1} + \mathbf{g}_n\left[\frac{d_n}{(2j-1)} - \mathbf{u}_n^T\mathbf{w}_{n-1}\right]. \end{aligned} \tag{2.64}$$

Podemos ver que o termo entre cochetes na Equação (2.64) é o *erro de estimação a priori do erro*, o qual denotamos por,

$$\epsilon_n = \frac{d_n}{(2j-1)} - \mathbf{u}_n^T.\hat{\mathbf{w}}_{n-1}. \tag{2.65}$$

Assim, a estrutura de atualização do algoritmo proposto é dada por,

$$\hat{\mathbf{w}}_n = \hat{\mathbf{w}}_{n-1} + \mathbf{g}_n \cdot \epsilon_n. \tag{2.66}$$

2.5.2 Resumo do algoritmo

Na Tabela 2.1 podemos observar o resumo do algoritmo proposto.

Tabela 2.1: Resumo do algoritmo RNQ para uma potência par do erro

Inicializar o algoritmo, definindo

$\mathbf{P}_0 = \delta^{-1}\mathbf{I}$, δ é uma constante positiva pequena

$\hat{\mathbf{w}}_0 = \mathbf{0}$

Para cada instante de tempo, $n = 1, 2, ...$, computar

$$\mathbf{g}_n = \frac{\lambda^{-1}\mathbf{P}_{n-1}\mathbf{u}_n}{\left[(2j-1)\mathbf{d}_n^{2j-2}\right]^{-1} + \lambda^{-1}\mathbf{u}_n^T\mathbf{P}_{n-1}\mathbf{u}_n}$$

$$\epsilon_n = \frac{d_n}{(2j-1)} - \hat{\mathbf{w}}_{n-1}^T\mathbf{u}_n$$

$$\hat{\mathbf{w}}_n = \hat{\mathbf{w}}_{n-1} + \mathbf{g}_n\epsilon_n$$

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

2.6 Convergência do algoritmo

Estudaremos nesta seção a convergência do algoritmo RNQ para uma potência par do erro no contexto de um problema de identificação de sistemas adaptativos, da mesma forma que estudamos a convergência do algoritmo RLS no capítulo anterior. A primeira análise será encontrar as condições de convergência do algoritmo e descrever como ele se comporta até atingir o estado estacionário. Isso pode ser realizado através do estudo do comportamento dos pesos. Em seguida analisaremos o desajuste e o tempo de aprendizagem do algoritmo proposto.

Como planta, consideramos um regressor múltiplo linear caracterizado pela equação,

$$d_n = e_n + \mathbf{w}_*^T\mathbf{u}_n, \tag{2.67}$$

sendo $\mathbf{w}_*$ o vetor peso do regressor, $\mathbf{u}_n$ o vetor entrada, e_n é o ruído da planta e d_n a saída da planta.

2.6.1 O comportamento médio do vetor peso no algoritmo

De (2.50), (2.53) e (2.67) obtemos,

$$\hat{\mathbf{w}}_n = \mathbf{\Phi}_n^{-1}.\left\{\sum_{i=1}^{n}\left[\lambda^{n-i}.\mathbf{u}_i.\left(\mathbf{w}_*^T\mathbf{u}_i + e_i\right)^{2j-1}\right]\right\}. \tag{2.68}$$

Utilizando desenvolvimento binomial do termo $\left(\mathbf{w}_*^T\mathbf{u}_i + e_i\right)^{2j-1}$, e desprezando os termos de alta potência (assumindo esses termos próximo de zero), podemos reescrever a Equação (2.68) como,

$$\hat{\mathbf{w}}_n \cong \mathbf{w}_* + \mathbf{\Phi}_n^{-1}.\sum_{i=1}^{n}\left[\lambda^{n-i}.\mathbf{u}_i.e_i^{2j-1}\right]. \tag{2.69}$$

Aplicando o operador esperança em ambos os membros da Equação (2.69) e lembrando que $\mathbf{u_i}$ e e_i são independentes para todo i, essa e que e_i tem média zero, obtemos,

$$E\left[\hat{\mathbf{w}}_n\right] = \mathbf{w}_*. \tag{2.70}$$

Desse resultado temos que $\hat{\mathbf{w}}_n$ é uma estimação não enviesada de $\mathbf{w}_*$. Em outras palavras, podemos afirmar que o algoritmo proposto converge em torno da média.

2.6.2 Matriz de correlação do vetor desvio

Para descrever a curva de aprendizagem do algoritmo nós estudamos a matriz de correlação do vetor desvio. Assim, iremos fazer uma mudança de variável para definir o vetor desvio como,

$$\mathbf{v}_n = \hat{\mathbf{w}}_n - \mathbf{w}_*, \tag{2.71}$$

sendo $\mathbf{w}_*$ a solução ótima.

De (2.69) e (2.71), obtemos,

$$\mathbf{v}_n = \mathbf{\Phi_n}^{-1} \cdot \sum_{i=1}^{n}\left[\lambda^{n-i} \cdot \mathbf{u}_i \cdot e_i^{2j-1}\right]. \tag{2.72}$$

Agora, definimos a matriz de correlação do vetor desvio como,

$$\mathbf{K}_n = E\left[\mathbf{v}_n \mathbf{v}_n^T\right]. \tag{2.73}$$

Substituindo (2.72) em (2.73) e notando que $\left(\mathbf{\Phi}_n^{-1}\right)^T = \mathbf{\Phi}_n^{-1}$, sabendo que o erro de medição neste processo é um ruído branco com variância σ^2 e que o vetor de entrada e o vetor desvio são independentes, essa última afirmação tem sido uma prática comum na análise de algoritmos em filtragem adaptativa, obtemos,

$$\mathbf{K}_n = \sigma^2 \cdot E\left[{\mathbf{\Phi}_n}^{-1}(\mathbf{U}_n \mathbf{\Lambda}_n \mathbf{\Lambda}_n \mathbf{U}_n^T){\mathbf{\Phi}_n}^{-1}\right], \tag{2.74}$$

sendo $\mathbf{\Lambda}_n$ uma matriz diagonal formadas pelos fatores exponenciais
$\lambda^{n-1}, \ \lambda^{n-2}, \ \ldots$, e $\mathbf{U}_n$ é a matriz dos dados de entrada, isto é:

$$\mathbf{U}_n = [\mathbf{u}_1 \ \ \mathbf{u}_2 \ \ \ldots \ \ \mathbf{u}_n]. \tag{2.75}$$

Para o cálculo rigoroso da Equação (2.74) devemos fazer as seguintes hipóteses [5] (p.427):

1. O vetor de entrada $\mathbf{u}_i$ constitui amostras de um processo ergódico. 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.

Podemos notar da Equação (2.51) que $\mathbf{\Phi}$ é uma soma ponderada dos produtos internos $[\mathbf{u}_n \cdot \mathbf{u}_n^T, \mathbf{u}_{n-1} \cdot \mathbf{u}_{n-1}^T, ...]$. Assim, considerando as hipóteses acima, temos que,

$$\mathbf{\Phi}_n \approx (2j-1)\, d_n^{2j-2} \frac{1-\lambda^n}{1-\lambda}\mathbf{R}, \tag{2.76}$$

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

Analogamente, obtemos,

$$\mathbf{U}_n \mathbf{\Lambda}_n \mathbf{\Lambda}_n \mathbf{U}_n^T \approx \frac{1-\lambda^{2n}}{1-\lambda^2}\mathbf{R}, \tag{2.77}$$

Substituindo (2.77) e (2.76) em (2.74) temos,

$$\mathbf{K}_n = \sigma^2 \cdot \left\{ \frac{1}{\left[(2j-1)\, d_n^{2j-2}\right]^2} \cdot \frac{1-\lambda}{1+\lambda} \cdot \frac{1+\lambda^n}{1-\lambda^n} \mathbf{R}^{-1} \right\}. \tag{2.78}$$

No estado estacionário, isto é, quando $n \to \infty$, e usando a Equação (2.78), obtemos,

$$\mathbf{K}_\infty = \sigma^2 \cdot \left\{ \frac{1}{\left[(2j-1)\, d^{2j-2}\right]^2} \cdot \frac{1-\lambda}{1+\lambda} \cdot \mathbf{R}^{-1} \right\}. \tag{2.79}$$

2.6.3 Curva de aprendizagem

De (2.65), (2.67) e (2.71) obtemos

$$\epsilon_n = \frac{1}{(2j-1)} \cdot (e_n - \mathbf{v}_{n-1}^T \mathbf{u}_n), \tag{2.80}$$

sendo$\mathbf{v}_{n-1}$ o vetor desvio no tempo $n-1$.

Como índice de desempenho estatístico para o algoritmo proposto, é conveniente usar um erro de estimativa a priori ϵ_n para definir o valor esperado do erro de grau $2j$,

$$\xi_n = E\left[|\epsilon_n|^{2j}\right], \tag{2.81}$$

sendo j um inteiro positivo. Substituindo (2.80) em (2.81), obtemos,

$$\xi_n = \frac{1}{(2j-1)^{2j}} \cdot E\left[\left|-\mathbf{v}_{n-1}^T \mathbf{u}_n + e_n\right|^{2j}\right]. \tag{2.82}$$

Utilizando desenvolvimento binomial e notando que $\mathbf{v}_{n-1}$ é próximo de zero, após o estado transitório, podemos desconsiderar alguns termos do lado direito da Equação (2.82) que incluem potências altas de $\mathbf{v}_{n-1}$. Entretanto, supondo que a distribuição de densidade do sinal erro, e_n, é simétrica e que $E\left[|e_n|^{2j}\right]$ é o 2j-ésima potência de e_n, temos,

$$\xi_n \approx \frac{1}{(2j-1)^{2j}} \cdot E\left[{e_n}^{2j}\right] + \frac{1}{(2j-1)^{2j}} \cdot j \cdot (2j-1) \cdot E\left[{e_n}^{2j-2}\right] tr\left\{\mathbf{RK}_{n-1}\right\}$$

$$+ \frac{1}{(2j-1)^{2j}} \cdot tr\left\{(\mathbf{RK}_{n-1})^j\right\}. \tag{2.83}$$

Substituindo (2.78) em (2.83) e definindo $\xi_{mim} = \frac{1}{(2j-1)^{2j}} \cdot E\left[\left|e_n\right|^{2j}\right]$ como o mínimo de e_n, quando $\mathbf{w} = \mathbf{w}_*$, obtemos a seguinte expressão para o algoritmo proposto,

$$\xi_n \approx \xi_{mim} + \frac{j}{(2j-1)^{2j+1}.d_n^{4j-4}} \cdot \frac{1-\lambda}{1+\lambda} \cdot \frac{1+\lambda^{n-1}}{1-\lambda^{n-1}} \cdot L \cdot \sigma^2 \cdot E\left[\left|e_n\right|^{2j-2}\right]$$

$$+ \frac{1}{(2j-1)^{4j}.d_n^{4j-4}} \cdot \frac{(1-\lambda)^j}{(1+\lambda)^j} \cdot \frac{(1+\lambda^{n-1})^j}{(1-\lambda^{n-1})^j} \cdot L \cdot (\sigma^2)^j, \tag{2.84}$$

sendo L a ordem do filtro.

A Equação (2.84) descreve a curva de aprendizagem do algoritmo RNQ para uma potência par do erro.

2.6.4 Análise do tempo de aprendizagem

Analogamente à análise feita na dedução da constante de tempo do RLS, no capítulo anterior, podemos verificar que a velocidade na qual o segundo e o terceiro termo do lado direito da Equação (2.84) convergem é determinada pelos termos exponenciais λ^n e λ^{nj}, respectivamente. Assim, podemos obter o tempo de convergência associado ao algoritmo proposto definido por $\tau_{\mathcal{RNQ}_1}$ e $\tau_{\mathcal{RNQ}_2}$, da seguinte forma: primeiro, para o segundo termo no lado direito Equação (2.84) temos,

$$e^{\frac{-n}{\tau_{\mathcal{RNQ}_1}}} = \lambda^n, \tag{2.85}$$

que nos dá,

$$\tau_{\mathcal{RNQ}_1} = \frac{-1}{\ln \lambda} = \tau_{\mathcal{RLS}}. \tag{2.86}$$

Analogamente obtemos para o último termo no lado direito da Equação (2.84),

$$e^{\frac{-n}{\tau_{\mathcal{RNQ}_2}}} = \lambda^{nj}, \tag{2.87}$$

que nos dá,

$$\tau_{\mathcal{RNQ}_2} = \frac{-1}{j \cdot \ln \lambda}. \tag{2.88}$$

2.6.5 Desajuste

O desajuste, $\mathcal{M}_{\mathcal{RNQ}}$, é definido como uma diferença dimensional entre o erro quadrático médio (MSE), e o MSE mínimo. Em outras palavras, este é uma medida normalizada do custo de adaptação [6].

O desajuste do algoritmo RLS é dado por [5],

$$\mathcal{M}_{\mathcal{RLS}} = \left(\frac{1-\lambda}{1+\lambda}\right) \cdot L. \tag{2.89}$$

Para o algoritmo proposto, encontramos,

$$\mathcal{M}_{\mathcal{RNQ}} = \frac{\lim_{n\to\infty}(\xi_n - \xi_{min})}{\xi_{min}} \cong \left(\frac{1-\lambda}{1+\lambda}\right) \frac{j}{(2j-1)^{2j+1}.d_n^{4j-4}} L. \tag{2.90}$$

2.6.6 Análise comparativa

Nesta seção faremos uma análise comparativa da velocidade de convergência e dos desajustes associados aos algoritmos RLS e RNQ para uma potência par do erro.

A curva de aprendizagem do algoritmo RLS é dada por [1],

$$\xi_{n_{RLS}} = \sigma^2 + \frac{1-\lambda}{1+\lambda} \cdot \frac{1+\lambda^{n-1}}{1-\lambda^{n-1}} \cdot L \cdot (\sigma^2). \tag{2.91}$$

Podemos observar da Equação (2.91) que o erro mínimo do algoritmo RLS é igual a σ^2 e que o estado transitório é governado pelo segundo termo da Equação (2.91).

Por outro lado, da Equação (2.84), observamos que a curva de aprendizagem do algoritmo proposto é composta por três estágios durante o aprendizado. O estado

estacionário é dado por ξ_{mim}, que tende para valores menores que σ^2 quando n tende para o infinito. Os outros dois estão associados a diferentes estatísticas do erro. Um está associado com σ^{2j} e o outro está associado com $|e_n|^{2j-2}$. A soma desses dois termos tende para zero mais rápido que o segundo termo na Equação (2.91) quando n tende para o infinito.

Para comparar a velocidade de adaptação dos algoritmos dividimos $\tau_{\mathcal{RLS}}$ por $\tau_{\mathcal{RNQ}_2}$,

$$\frac{\tau_{\mathcal{RLS}}}{\tau_{\mathcal{RNQ}_2}} = j, \tag{2.92}$$

sendo j um inteiro positivo.

Da Equação (2.92), podemos observar que $\tau_{\mathcal{RNQ}_2} = \tau_{\mathcal{RLS}}$ for $j = 1$ e, $\tau_{\mathcal{RNQ}_2} < \tau_{\mathcal{RLS}}$ for $j > 1$.

Analisando as Equações (2.86) e (2.88), podemos observar claramente que a velocidade de convergência do algoritmo RNQ para uma potência par do erro é mais rápida que a do algoritmo RLS padrão.

Das Equações (2.89) e (2.90), temos que o desajuste $\mathcal{M}_{\mathcal{RNQ}} = \mathcal{M}_{\mathcal{RLS}}$ para $j = 1$ e $\mathcal{M}_{\mathcal{RNQ}} < \mathcal{M}_{\mathcal{RLS}}$, para $j > 1$.

2.6.7 Complexidade computacional

Podemos extrair da Equação (2.66) a complexidade computacional do algoritmo proposto. O algoritmo RNQ para uma potência par do erro requer duas divisões, $L^2 + 5L + 2j$ multiplicações, $L^2 + 3L$ adições por iteração, para um filtro de ordem L. Assim, esse algoritmo requer $2j - 1$ multiplicações e uma divisão escalar extras por iteração quando comparado com algoritmo RLS padrão [7]. A complexidade computacional é da ordem de $O(N^2)$ e é comparável com a do algoritmo RLS.

Capítulo 3

O Algoritmo RLS estendido

Descreveremos o critério determinístico do algoritmo que é equivalente a um filtro de Kalman [7]. O objetivo é descrever o algoritmo RLS estendido que são os mais adequados para o rastreamento do vetor de estado do modelo de espaço de estado definido nas equações (1.1) e (1.3).

3.1 Critério do mínimo quadrado

Nesta seção, mostraremos que o problema determinístico dos mínimos quadrados é equivalente ao modelo estocástico desempenhando um papel central na teoria de estimação, uma vez que permite obter soluções ótimas em qualquer dos dois modelos.

3.1.1 Problema de estimação linear

Seja $\mathbf{d}$ um processo aleatório escalar de média zero com variância σ_d^2,

$$\mathbb{E}[\mathbf{d}] = 0, \quad \sigma_d^2 = \mathbb{E}[|\mathbf{d}|^2],$$

e seja $\mathbf{u}^T$ um vetor aleatório de média zero, $M \times 1$, com matriz de covariância definida positiva denotada por R_u,

$$R_u = \mathbb{E}[\mathbf{u}^T\mathbf{u}].$$

As variáveis $\{\mathbf{d}, \mathbf{u}\}$ podem ser, em geral, complexas.

O vetor de covariância cruzada, $M \times 1$ de $\{\mathbf{d}, \mathbf{u}\}$ é denotado por

$$R_{du} \triangleq \mathbb{E}[\mathbf{d}\mathbf{u}^T]. \quad \text{(um vetor coluna)}.$$

Então consideremos o problema de estimar $\mathbf{d}$ de $\mathbf{u}$ no sentido dos mínimos médio quadrado linear como segue:

$$\min_{w} \mathbb{E}[|\mathbf{d} - \mathbf{u}w|^2], \tag{3.1}$$

onde w é um vetor peso $M \times 1$.

3.1.2 Problema dos mínimos quadrados

Suponhamos que temos disponíveis N realizações dos processos estocásticos $\{\mathbf{d}, \mathbf{u}\}$, digamos

$$\{d(0), d(1), ..., d(N-1)\}, \quad \{u_0, u_1, ..., u_{N-1}\},$$

onde os $\{d(i)\}$ são escalares e os $\{u_i\}$ são vetores $1 \times M$. Dados $\{d(i), u_i\}$, podemos aproximar o erro médio quadrático em (3.1) pela sua média amostral como

$$\mathbb{E}[|\mathbf{d} - \mathbf{u}w|^2] \approx \frac{1}{N} \sum_{i=0}^{N-1} |d(i) - u_i w|^2. \tag{3.2}$$

Neste modo, o problema de otimização (3.1) pode ser trocado por

$$\min_{w} \left(\sum_{i=0}^{N-1} |d(i) - u_i w|^2 \right), \tag{3.3}$$

onde removemos o fator de escala $1/N$.

3.1.3 Formulação vetorial

A função de custo (3.3) pode ser reformulada em notação vetorial como segue. Agrupamos as observações $\{d(i)\}$ em um vetor y, $N \times 1$ e os vetores linhas $\{u_i\}$ em uma matriz H, $N \times M$

$$y \triangleq \begin{bmatrix} d(0) \\ d(1) \\ d(2) \\ \vdots \\ d(N-1) \end{bmatrix}, H \triangleq \begin{bmatrix} u_0 \\ u_1 \\ u_2 \\ \vdots \\ u_{N-1} \end{bmatrix},$$

então (3.3) pode ser reescrito como

$$\min_{w} \|y - Hw\|^2, \tag{3.4}$$

onde $\|.\|^2$ denota a norma euclidiana, ou seja, $\|a\|^2 = a^T a$, para um vetor coluna a. O problema (3.4) é conhecido como problema dos mínimos quadrados padrão.

Definição 3.1.1 (Problema do mínimo quadrático) *Dado um vetor $N \times 1$, y, e uma $N \times M$ matriz H, o problema do mínimo quadrado busca um $M \times 1$ vetor w que resolve*

$$\min_w \|y - Hw\|^2.$$

Note que, para um w, o vetor Hw está no espaço coluna (ou espaço range) da matriz H, escrita como $Hw \in R(H)$. Portanto, o critério do mínimo quadrado (3.4) é tal que ele busca um vetor coluna no espaço range de H que está mais próximo de y na norma euclidiana. Especificamente, o problema do mínimo quadrado busca um $\widehat{w}$ tal que $H\widehat{w}$ está mais próximo de y.

Agora sabemos da geometria euclidiana que o vetor mais próximo de y em $R(H)$ é tal que o vetor residual, $y - H\widehat{w}$, é ortogonal a todos os vetores em $R(H)$.

Portanto, deve considerar que um candidato solução $\widehat{w}$ resultará em um residual $(y - H\widehat{w})$ que é ortogonal a Hp, para todo vetor p ou, equivalentemente,

$$p^T H^T (y - H\widehat{w}) = 0.$$

Claramente, o único vetor que é ortonormal a um vetor p é o vetor nulo, tal que devemos ter

$$H^T (y - H\widehat{w}) = 0. \tag{3.5}$$

Concluímos que uma solução $\widehat{w}$ do problema mínimo quadrático (3.4) deve satisfazer a chamada equação normal

$$H^T H\widehat{w} = H^T y. \tag{3.6}$$

Estas equações são sempre consistente, isto é, uma solução $\widehat{w}$ sempre existe. Isto é devido as matrizes $H^T H$ e H^T terem o mesmo gerador coluna, isto é, $R(H^T) = R(H^T H)$.

Teorema 3.1.1 (A equação normal) *Um vetor $\widehat{w}$ resolve o problema do mínimo quadrático (3.4) se e somente se, satisfaz a equação normal*

$$H^T H\widehat{w} = H^T y,$$

ou equivalentemente, se e somente e, satisfaz a condição de ortogonalidade $y - H\widehat{w} \perp R(H)$.

■

3.1.4 Mínimo quadrático ponderado

É frequente o caso em que uma ponderação é incorporada à função de custo do problema mínimo quadrático, tal que (3.4) é trocado por

$$\min_w (y - Hw)^T W (y - Hw), \quad W > 0, \tag{3.7}$$

onde H é uma matriz simétrica definida positiva.

Um modo de resolver (3.7) é mostrar que ela se reduz a forma padrão (3.4). Para isso, usamos a decomposição de W como

$$W = V \Lambda V^T,$$

onde Λ é diagonal com entradas positivas e V é ortogonal, ou seja,

$$VV^T = V^T V = I.$$

Dessa forma a equação normal (3.6) agora passará a ser

$$H^T W H \widehat{w} = H^T W y. \tag{3.8}$$

Teorema 3.1.2 (mínimo quadrático ponderado) *Um vetor $\widehat{w}$ é uma solução do problema mínimo quadrático ponderado (3.7) se, e somente se, satisfaz a equação normal*

$$H^T W H \widehat{w} = H^T W y.$$

■

3.1.5 Mínimo quadrático regularizado

Uma segunda variação do problema mínimo quadrático padrão (3.4) é o mínimo quadrático regularizado. Nesta formulação, buscamos um vetor $\widehat{w}$ que resolve

$$\min_w [(w - \bar{w})^T \Pi (w - \bar{w}) + \|y - Hw\|^2], \tag{3.9}$$

onde, comparado com (3.9), estamos incorporando um termo chamado regularização $\|w - \bar{w}\|_W^2$. Aqui, Π é uma matriz definida positiva, geralmente um múltiplo da matriz identidade, e $\bar{w}$ é um vetor coluna dado, geralmente $\bar{w} = 0$. Pode ser mostrado que

Teorema 3.1.3 (Mínimo quadrático regularizado) *A solução do problema mínimo quadrático regularizado (3.9) é sempre única e dada por*

$$\widehat{w} = \bar{w} + [\Pi + H^T H]^{-1} H^T (y - H\bar{w}).$$

O custo mínimo resultante é dado pela expressão

$$\begin{aligned} \xi &= (y - H\bar{w})^T \tilde{y} \\ &= (y - H\bar{w})^T [I + H\Pi^{-1} H^T]^{-1} (y - H\bar{w})^T, \end{aligned}$$

onde $\tilde{y} = y - \widehat{y}$ *e* $\widehat{y} = H\widehat{w}$. *Além disso,* $\widehat{w}$ *satisfaz a condição de ortogonalidade*

$$H^T \tilde{y} = \Pi(\widehat{w} - \bar{w})$$

■

3.1.6 Mínimo quadrático ponderado regularizado

Podemos combinar as formulações vistas anteriormente e introduzir um problema mínimo quadrático ponderado regularizado. A versão ponderada de (3.9) deve ter a forma

$$\min_w [(w - \bar{w})^T \Pi (w - \bar{w}) + (y - Hw)^T W (y - Hw)]. \tag{3.10}$$

onde, como antes, W é definida positiva. Para isso, decomponha a matriz W, como $W = V\Lambda V^T$, e defina as quantidades normalizadas

$$a \triangleq \Lambda^{1/2} V^T y, \quad A \triangleq \Lambda^{1/2} V^T H,$$

então o problema ponderado regularizado (3.10) torna-se

$$\min_w [(w - \bar{w})^T \Pi (w - \bar{w}) + \|y - Hw\|_W^2], \tag{3.11}$$

que é da mesma forma como no problema mínimo quadrático regularizado não ponderado (3.9). Com isso teremos

Teorema 3.1.4 (Mínimo quadrático ponderado regularizado) *A solução do problema mínimo quadrático ponderado regularizado (3.10) é sempre única e dada por*

$$\widehat{w} = \bar{w} + [\Pi + H^T W H]^{-1} H^T W (y - H\bar{w}),$$

e o custo mínimo resultante é dado por

$$\begin{aligned}\xi &= (y - H\bar{w})^T W\tilde{y} \\ &= (y - H\bar{w})^T[W^{-1} + H\Pi^{-1}H^T]^{-1}(y - H\bar{w})^T,\end{aligned}$$

onde $\tilde{y} = y - \widehat{y}$ *e* $\widehat{y} = H\widehat{w}$. *Além disso,* $\widehat{w}$ *satisfaz a condição de ortogonalidade*

$$H^T W\tilde{y} = \Pi(\widehat{w} - \bar{w}).$$

■

3.1.7 Resultado de equivalência em estimação linear

Problema estocástico

Sejam $\mathbf{x}$ e $\mathbf{y}$ variáveis aleatórias de média zero que estão relacionadas via modelo linear da forma

$$\mathbf{y} = H\mathbf{x} + \mathbf{v}. \tag{3.12}$$

Para uma matriz H onde $\mathbf{v}$ denota um ruído aleatório de média zero com matriz de covariância conhecida, ou seja, $R_v = \mathbb{E}[\mathbf{v}\mathbf{v}^T]$. A matriz de covariância de $\mathbf{x}$ é também conhecida e denotada por $\mathbb{E}[\mathbf{x}\mathbf{x}^T] = R_x$. Ambos $\{\mathbf{x}, \mathbf{v}\}$ são descorrelacionados, isto é, $\mathbb{E}[\mathbf{x}\mathbf{v}^T] = 0$, e assumimos que $R_x > 0$ e $R_v > 0$.

É mostrado que o estimador mínimo médio quadrático linear de $\mathbf{x}$ dado $\mathbf{y}$ é

$$\widehat{\mathbf{x}} = [R_x^{-1} + H^T R_v^{-1} H]^{-1} H^T R_v^{-1}\mathbf{y}, \tag{3.13}$$

e que a matriz de erro mínimo médio quadrático resultante é

$$m.m.s.e. = [R_x^{-1} + H^T R_v^{-1} H]^{-1}. \tag{3.14}$$

Problema determinístico

Consideremos agora variáveis determinísticas $\{x, y\}$ e uma matriz H relacionados entre si como

$$y = Hx + v, \tag{3.15}$$

onde v denota um ruído de medida. Suponha também que colocamos o problema de estimar x resolvendo o problema mínimo quadrático ponderado regularizado

$$\min_y [x^T \Pi x + \|y - Hx\|_W^2], \tag{3.16}$$

onde $\Pi > 0$ é uma matriz de regularização e $W > 0$ é uma matriz de ponderação.

Pode ser mostrado que a solução $\widehat{x}$ é dada por

$$\widehat{x} = [\Pi + H^T W H]^{-1} H^T W y, \tag{3.17}$$

e que o custo mínimo resultante é

$$\xi = y^T [W^{-1} + H \Pi^{-1} H^T]^{-1} y. \tag{3.18}$$

Equivalência

A expressão (3.13) fornece um estimador mínimo médio quadrático linear $\mathbf{x}$ numa estrutura estocástica, enquanto que a expressão (3.17) fornece o estimador mínimo quadrático de x na estrutura determinística (3.16). E ainda, é claro que se trocarmos as quantidades em (3.13) por

$$R_x \longleftarrow \Pi^{-1}, \quad R_v \longleftarrow W^{-1},$$

então a solução estocástica (3.13) deve coincidir com a solução determinística (3.17). Portanto dizemos que ambos os problemas são equivalentes. Tal equivalência desempenha um papel central em teoria de estimação, uma vez que nos permite combinar as formulações determinísticas e estocásticas e para determinar a solução para o contexto de uma solução do outro.

3.2 Estimação determinística

Considere uma coleção de $(N+1)$ medições $\{d_i\}$, possivelmente vetores colunas, que satisfazem

$$d_i = U_i x_i + v_i, \tag{3.19}$$

onde os $\{x_i\}$ envolvem o tempo de acordo com a recursão de estado

$$x_{i+1} = F_i x_i + G_i n_i, \tag{3.20}$$

aqui as matrizes $\{F_i, G_i, U_i\}$ são matrizes conhecidas e $\{n_i, v_i\}$ são os ruídos.

Seja ainda Π_0 uma *matriz de regularização definida positiva*, e sejam $\{Q_i, R_i\}$ matrizes definidas positivas ponderadas. Dados $\{d_i\}$, colocamos o problema de

estimar o vetor de estado inicial x_0 e os sinais $\{n_0, n_1, ..., n_N\}$ na maneira dos mínimos quadrados regularizados resolvendo

$$\min_{\{x_0, n_0, ..., n_N\}} \left[x_0^T \Pi_0^{-1} x_0 + \sum_{i=0}^{N} (d_i - U_i x_i)^T R_i^{-1} (d_i - U_i x_i) + \sum_{i=0}^{N} n_i^T Q_i^{-1} n_i \right], \quad (3.21)$$

sujeito à restrição (3.20). Denotamos a solução por

$$\{\hat{x}_{0|N}, \hat{n}_{j|N},\ 0 \leq j \leq N\},$$

e nos referimos a elas como estimativas suaves pois elas são baseadas em observações além do tempo de ocorrência das respectivas variáveis $\{x_0, n_j\}$.

Em princípio, poderíamos resolver (3.21) usando argumento variacional (otimização), isto é, usando um argumento de multiplicadores de Lagrange. Em vez disso, vamos resolver apelando para o problema de equivalência . Em outras palavras, primeiro determinaremos o problema estocástico equivalente e então resolveremos este último problema para chegar à solução de (3.21).

Este método de resolução em (3.21) não serve apenas como uma ilustração da conveniência nos resultados da equivalência na teoria da estimação, mas também mostra que às vezes é mais fácil resolver um problema determinístico no domínio estocástico (ou vice-versa). No nosso caso, o problema em questão é mais conveniente resolvido no domínio estocástico.

Defina os vetores colunas

$$z = col\{x_0, n_0, n_1, ..., n_N\}$$

e

$$y = col\{d_0, d_1, ..., d_N\},$$

bem como as matrizes diagonais em bloco

$$W^{-1} \triangleq (R_0 \oplus R_1 \oplus ... \oplus R_N),\ \Pi^{-1} \triangleq (\Pi_0 \oplus Q_0 \oplus ... \oplus Q_N), \quad (3.22)$$

então o termo

$$x_0^T \Pi^{-1} x_0 + \sum_{i=1}^{N} n_i^T Q_i^{-1} n_i,$$

que aparece em (3.21) é igual a $z^T \Pi z$. Além disso, usando a equação de estado (3.20) para expressar cada termo $U_i x_i$ em termos de combinações das entradas de z, podemos verificar que

$$\sum_{i=0}^{N} (d_i - U_i x_i)^T R^{-1} (d_i - U_i x_i) = (y - Hz)^T W (y - Hz) = \|y - Hz\|_W^2,$$

onde a matriz H é triangular inferior e é dada por

$$H \triangleq \begin{bmatrix} U_0 & & & & \\ U_1\Phi(1,0) & U_1 G_0 & & & \\ \vdots & \vdots & \vdots & & \\ U_N\Phi(N,0) & U_N\Phi(N,1)G_0 & U_N\Phi(N,2)G_1 & \dots & U_N G_{N-1} \quad 0 \end{bmatrix}, \tag{3.23}$$

e as matrizes $\Phi(i,j)$ são definidas por

$$\Phi(i,j) \triangleq \begin{cases} F_{i-1}F_{i-2}\cdots F_j, & i > j \\ \mathbf{I}, & i = j \end{cases},$$

em outras palavras, podemos reescrever a função de custo original (3.21) como uma função de custo dos mínimos quadrados regularizada

$$\min_z \left[z^T \Pi z + (y - Hz)^T W (y - Hz) \right]. \tag{3.24}$$

Seja $\hat{z}_{|N}$ denotando a solução de (3.24), isto é, $\hat{z}_{|N}$ é um vetor coluna que contém a solução desejada $\{\hat{x}_{0|N}, \hat{n}_{0|N}, \hat{n}_{1|N}, ..., \hat{n}_{N|N}\}$. Agora, em vista do resultado de equivalência, sabemos que $\hat{z}_{|N}$ pode ser obtido resolvendo um problema de estimação estocástica que é determinado como segue

Estimação estocástica

Introduzimos vetores aleatórios de média zero $\{\mathbf{z}, \mathbf{y}\}$, com mesma dimensão e partições como acima $\{z, y\}$, e assumimos que eles estão relacionados via modelo linear da forma

$$\mathbf{y} = \mathbf{H}\mathbf{z} + \mathbf{v}, \tag{3.25}$$

onde H é a mesma matriz como em (3.23), onde v denota um vetor ruído aditivo média zero, descorrelacionado com $\mathbf{z}$, e particionado como

$$\mathbf{v} = col\{\mathbf{v}_0, \mathbf{v}_1, ..., \mathbf{v}_N\}.$$

As dimensões de $\{\mathbf{v}_i\}$ são compatíveis com as de $\{\mathbf{y}_i\}$. Denotamos as matrizes de covariância de $\{\mathbf{z}, \mathbf{v}\}$ por

$$R_z \triangleq \mathbb{E}[\mathbf{z}\mathbf{z}^T],\ R_v = \mathbb{E}[\mathbf{v}\mathbf{v}^T], \tag{3.26}$$

e escolhemos elas como

$$R_z = \Pi^{-1} \quad R_v = W^{-1},$$

onde $\{\Pi, W\}$ são dadas por (3.22).

Seja $\hat{\mathbf{z}}_{|N}$ o estimador linear mínimo médio quadrático de $\mathbf{z}$ dadas as estradas $\{\mathbf{d}_0, \mathbf{d}_1, ..., \mathbf{d}_N\}$ em $\mathbf{y}$. Particionamos mais ainda $\mathbf{z}$ como

$$\mathbf{z} = col\{\mathbf{x}_0, \mathbf{n}_0, \mathbf{n}_1, ..., \mathbf{n}_N\}.$$

Então o resultado de equivalência afirma que a expressão para $\hat{\mathbf{z}}_{|N}$ em termos de $\mathbf{y}$ no problema estocástico (3.25) é idêntico para a expressão para $\hat{z}_{|N}$ em termos de y no problema determinístico (3.24).

A fim de determinar $\hat{\mathbf{z}}_{|N}$ ou, equivalentemente, $\{\hat{\mathbf{x}}_{0|N}, \hat{\mathbf{n}}_{j|N}\}$, começamos notando que o modelo linear (3.25), juntamente com as definições de $\{R_z, R_v, H\}$ em (3.23) e (3.26), mostram que as variáveis estocásticas $\{\mathbf{d}_i, \mathbf{v}_i, \mathbf{x}_0, \mathbf{n}_i\}$ como definidas satisfazem o seguinte modelo de espaço de estado.

$$\mathbf{x}_{i+1} = F_i\mathbf{x}_i + G_i\mathbf{n}_i \tag{3.27}$$

$$\mathbf{d}_i = U_i\mathbf{x}_i + \mathbf{v}_i \tag{3.28}$$

com

$$\mathbb{E}\left[\begin{bmatrix}\begin{bmatrix}\mathbf{n}_i\\ \mathbf{v}_i\\ \mathbf{x}_0\end{bmatrix}\\ 1\end{bmatrix}\begin{bmatrix}\begin{bmatrix}\mathbf{n}_j\\ \mathbf{v}_j\\ \mathbf{x}_0\end{bmatrix}^T\end{bmatrix}\right] = \begin{bmatrix}Q_i\delta_{ij} & 0 & 0\\ 0 & R_i\delta_{ij} & 0\\ 0 & 0 & \Pi_0\\ 0 & 0 & 0\end{bmatrix}. \tag{3.29}$$

Agora podemos usar este modelo para derivar recursivamente para estimação de $\mathbf{z}$ (isto é, estimando $\{\mathbf{x}_0, \mathbf{n}_1, ..., \mathbf{n}_N\}$).

Resolvendo o problema estocástico

Seja $\hat{\mathbf{z}}_{|i}$ que denota o estimador linear mínimo médio quadrático (LLMS-Linear Least Mean Squares) de $\mathbf{z}$ dadas as principais entradas $\{\mathbf{d}_0, ..., \mathbf{d}_i\}$ em $\mathbf{y}$. Para

determinar $\hat{\mathbf{z}}_{|i}$, e finalmente $\hat{\mathbf{z}}_{|N}$, procedemos recursivamente empregando inovações $\{\mathbf{e}_i\}$ das observações $\{\mathbf{y}_i\}$. Usando a fórmula básica de estimação recursiva , temos

$$\begin{aligned}\hat{\mathbf{z}}_{|i} &= \hat{\mathbf{z}}_{|i-1} + (\mathbb{E}[\mathbf{z}\mathbf{e}_i^T])R_{e,i}^{-1}\mathbf{e}_i \\ &= \hat{\mathbf{z}}_{|i-1} + \underbrace{(\mathbb{E}[\mathbf{z}\tilde{\mathbf{x}}_{i|i-1}^T])}_{\triangleq K_{z,i}} U_i^T R_{e,i}^{-1}\mathbf{e}_i, \quad \hat{\mathbf{z}}_{|-1} = 0, \end{aligned} \tag{3.30}$$

onde usamos na última igualdade a equação de inovação

$$\begin{aligned}\mathbf{e}_i &= \mathbf{d}_i - U_i\hat{\mathbf{x}}_{i|i-1} \\ &= U_i\hat{\mathbf{x}}_{i|i-1} + \mathbf{v}_i,\end{aligned}$$

e o fato que

$$\mathbb{E}[\mathbf{x}_0\mathbf{v}_i^T] = 0 \quad \text{e} \quad \mathbb{E}[\mathbf{n}_j\mathbf{v}_i^T] = 0,$$

para todo j. Temos também introduzido uma matriz de ganho definida por

$$K_{z,i} \triangleq \mathbb{E}[\mathbf{z}\tilde{\mathbf{x}}_{i|i-1}^T],$$

e, as entradas de $\hat{\mathbf{z}}_{|i}$ têm a seguinte interpretação

$$\hat{\mathbf{z}}_{|i} = col\{\hat{\mathbf{x}}_{0|i}, \hat{\mathbf{n}}_{0|i}, \hat{\mathbf{n}}_{1|i}, ..., \hat{n}_{i-1|i}, 0, ..., 0\},$$

onde as entradas à direita de $\hat{\mathbf{z}}_{|i}$ são zero pois $\hat{\mathbf{n}}_{j|i} = 0$, para $j \geq i$.

A construção recursiva acima deve ser completa, e portanto, fornece uma quantidade desejada $\hat{\mathbf{z}}_{|N}$, uma vez que mostramos como avaliar a matriz de ganho $K_{z,i}$. Para este propósito primeiro subtraem as equações

$$\mathbf{x}_{i+1} = F_i\mathbf{x}_i + G_i\mathbf{n}_i$$

e

$$\hat{\mathbf{x}}_{i+1|i} = F_i\hat{\mathbf{x}}_{i|i-1} + K_{p,i}[U_i\hat{\mathbf{x}}_{i|i-1} + \mathbf{v}_i],$$

para obter

$$\hat{\mathbf{x}}_{i+1|i} = F_{p,i}\tilde{\mathbf{x}}_{i|i-1} + G_i\mathbf{n}_i - K_{p,i}\mathbf{v}_i,$$

onde $F_{p,i} = F_i - K_{p,i}U_i$.

Usando esta recursão, verificamos que $K_{z,i}$ satisfaz a recursão

$$K_{z,i+1} = \mathbb{E}[\mathbf{z}\tilde{\mathbf{x}}_{i+1|i}^T] = K_{z,i}F_{p,i}^T + \left[\begin{array}{c} 0 \\ - - - \\ 0 \\ I \\ 0 \end{array}\right] Q_i G_i^T,\ K_{z,0} = \left[\begin{array}{c} \Pi_0 \\ 0 \end{array}\right]. \tag{3.31}$$

A matriz identidade que aparece no segundo termo da recursão para $K_{z|i+1}$ ocorre na posição que corresponde a entrada $\mathbf{n}_i$, no vetor $\mathbf{z}$, ou seja,

$$K_{z,1} = \left[\begin{array}{c} \Pi_0 F_{p,0}^T \\ Q_0 G_0^T \\ 0 \end{array}\right], \quad K_{z,2} = \left[\begin{array}{c} \Pi_0 F_{p,0}^T F_{p,1}^T \\ Q_0 G_0^T F_{p,1}^T \\ Q_1 G_1^T \\ 0 \end{array}\right], \ ...$$

Substituindo (3.31) em (3.30) achamos a seguinte recursão

$$\begin{aligned} \hat{x}_{0|i} &= \hat{x}_{0|i-1} + \Pi_0 \Phi_p^T(i,0) U_i^T R_{e,i}^{-1} e_i,\ \hat{x}_{0|-1} = 0, \\ \hat{n}_{j|i} &= \hat{n}_{j|i-1} + Q_j G_j^T \Phi_p^T(i,j+1) U_i^T R_{e,i}^{-1} e_i,\ j < i, \\ \hat{n}_{j|i} &= 0,\ j \geq i, \end{aligned} \tag{3.32}$$

onde a matriz $\Phi_p(i,j)$ é definida por

$$\Phi_p(i,j) \triangleq \left\{ \begin{array}{rl} F_{p,i-1}F_{p,i-2}...F_{p,j} & i > j \\ I & i = j \end{array} \right. .$$

Se introduzirmos a variável auxiliar

$$\rho_{i|N} \triangleq \sum_{j=1}^{N} \Phi_p^T(j,i) U_j^T R_{e,j}^{-1} \mathbf{e}_j,$$

então as recursões (3.32) produzem

$$\begin{aligned} \hat{\mathbf{x}}_{0|N} &= \Pi_0 \rho_{0|N}, \\ \hat{\mathbf{x}}_{j+1|j} &= F_{p,j}\hat{\mathbf{x}}_{j|j-1} + K_{p,j}\mathbf{d}_j,\ \hat{\mathbf{x}}_{0|-1} = 0, \\ \mathbf{e}_j &= \mathbf{d}_j - U_j \hat{\mathbf{x}}_{j|j-1}, \\ \hat{\mathbf{n}}_{j|N} &= Q_j G_j^T \rho_{j+1|N}, \\ \rho_{j|N} &= F_{p,j}^T \rho_{j+1|N} + U_i^T R_{e,j}^{-1} \mathbf{e}_j,\ \rho_{N+1|N}. \end{aligned} \tag{3.33}$$

Estas equações são conhecidas como recursões de Bryson-Frazier na literatura de filtro de Kalman [7]; as recursões em (3.33) calcula os estimadores $\{\hat{\mathbf{x}}_{0|i}, \hat{\mathbf{n}}_{j|i}\}$ para valores sucessivos de i, e não somente para $i = N$ como em (3.33). Apenas como em $\{\hat{\mathbf{x}}_{0|N}, \hat{\mathbf{n}}_{j|N}\}$, os estimadores $\{\hat{\mathbf{x}}_{0|i}, \hat{\mathbf{n}}_{j|i}\}$ podem também está relacionado com a solução de um problema de mínimo quadrado. De fato, pela equivalência, a expressão que fornece as soluções $\{\hat{\mathbf{x}}_{0|i}, \hat{\mathbf{n}}_{j|i}\}$ em (3.32) deve coincidir com aquelas que fornecem as soluções $\{\hat{x}_{0|i}, \hat{n}_{j|i}\}$ para o seguinte problema deterministico, com dados até o tempo i (em vez de N como em (3.21))

$$\min_{\{x_0, n_0, \ldots, n_N\}} \left[x_0^T \Pi^{-1} x_0 + \sum_{j=0}^{i} (d_j - U_j x_j)^T R_j^{-1} (d_j - U_j x_j) + \sum_{j=0}^{i} n_j^T Q_j^{-1} n_j \right]. \quad (3.34)$$

Resolvendo o problema determinístico

Sabemos pela equivalência que a aplicação de $\{\mathbf{d}_j\}$ a $\{\hat{\mathbf{x}}_{0|N}, \hat{\mathbf{n}}_{j|N}\}$ no problema estocástico (3.25) coincide com a aplicação de $\{d_j\}$ a $\{\hat{x}_{0|N}, \hat{n}_{j|N}\}$ no problema determinístico (3.24). Somos, portanto, levados a seguinte afirmação

Algoritmo RLS-Estendido: Considere medidas $\{d_i\}$ que satisfazem um modelo de espaço de estado da forma

$$x_{i+1} = F_i x_i + G_i n_i$$

$$d_i = U_i x_i + v_i,$$

onde as matrizes $\{F_i, G_i, U_i\}$ são conhecidas. A solução $\{\hat{x}_{0|N}, \hat{n}_{j|N}\}$ de

$$\min_{\{x_0, n_0, \ldots, n_N\}} \left[x_0^T \Pi_0^{-1} x_0 + \sum_{i=0}^{N} (d_i - U_i x_i)^T R^{-1} (d_i - U_i x_i) + \sum_{i=0}^{N} n_i^T Q_i^{-1} n_i \right],$$

pode ser determinada recursivamente como segue. Comece com $\hat{x}_{0|1} = 0, P_{0|-1} = \Pi_0$ e $\rho_{N+1|N} = 0$, e execute as seguintes equações de $i = 0$ a $i = N$,

$$R_{e,i} = R_i + U_i P_{i|i-1} U_i^T,$$

$$K_{p,i} = F_i P_{i|i-1} U_i^T R_{e,i}^{-1},$$

$$e_i = d_i - U_i \hat{x}_{i|i-1},$$

$$\hat{x}_{i+1|i} = F_i \hat{x}_{i|i-1} + K_{p,i} e_i,$$

$$P_{i+1|i} = F_i P_{i|i-1} F_i^T + G_i Q_i G_i^T - K_{p,i} R_{e,i} K_{p,i}^T,$$

então execute a recursão de $i = N$ para $i = 0$:

$$\rho_{i|N} = F_{p,i}^T \rho_{i+1|N} + U_i^T R_{e,i}^{-1} e_i,$$

e tome

$$\hat{x}_{0|n} = \Pi_0 \rho_{0|N},$$

$$\hat{n}_{i|N} = Q_i G_i^T \rho_{i+1|N},\ 0 \leq i \leq N.$$

3.2.1 Caso especial

Rastreamento: O algoritmo EX-RLS sugere extensões do RLS a fim de rastrear variações no vetor peso que pode ser capturado pela recursão linear de espaço de estado. Assim, suponha, por exemplo que a medida $\{d_i\}$ satisfaz

$$d_i = U_i w_i^0 + v_i,$$

e que o vetor de peso desconhecido w_i^0 evolui no tempo de acordo com

$$w_{i+1}^0 = \alpha w_i^0 + n_i,$$

para algum escalar $|\alpha| \leq 1$ e ruído n_i. Este modelo é um caso especial do modelo de espaço de estado usado pelo algoritmo EX-RLS se tomarmos as identificações $x_i = w_i^0$ e $F_i = \alpha I,\ G_i = I$. Então, novamente, se selecionarmos $\Pi_0^{-1} = \lambda\Pi,\ R_i = \lambda^i$ e $Q_i = q\lambda^i I$, para algum escalar positivo q e algum escalar $0 \ll \lambda < 1$ usado para introduzir o peso exponencialmente, então a recursão do algoritmo EX-RLS reduzirá as seguintes equações. Comece com $\hat{x}_{0|-1} = 0$ e $P_{0|-1} = \lambda^{-1}\Pi^{-1}$ e repita para $i \geq 1$

$$R_e(i) = \lambda^i + u_i P_{i|i-1} u_i^T,$$

$$K_{p,i} = \alpha P_{i|i-1} u_i^T / R_e(i),$$

$$e(i) = d(i) - u_i \hat{x}_{i|i-1},$$

$$P_{i+1|i} = |\alpha|^2 \left[P_{i|i+1} - \frac{P_{i|i-1} u_i^T u_i P_{i|i-1}}{\lambda^i + u_i P_{i|i-1} u_i^T} \right] + \lambda^i q I.$$

Neste caso, como $x_i = w_i^0$, a quantidade $\hat{x}_{i|i-1}$ serve como uma estimativa para w_i^0 que é baseada na medida até o tempo $i-1$. Se denotarmos $\hat{x}_{i+1|i}$ por w_i e introduzirmos a mudança de variável

$$P_i \triangleq \lambda^{-1} P_{i+1|i},$$

então multiplicando a recursão para $P_{i+1|i}$ por λ^{-i} em ambos os lados, as equações acima podem ser reescritas equivalentemente na seguinte forma.

Comece com $w_{-1} = 0$ e $P_{-1} = \Pi^{-1}$ e repita para $i \geq 0$:

$$w_i = \alpha w_{i-1} + \frac{\lambda^{-1}\alpha P_{i-1} u_i^T}{1 + \lambda^{-1} u_i P_{i-1} u_i^T}[d_i - u_i w_{i-1}],$$

$$P_i = \lambda^{-1}|\alpha|^2 \left[P_{i-1} - \frac{\lambda^{-1} P_{i-1} u_i^T u_i P_{i-1}}{1 + \lambda^{-1} u_i P_{i-1} u_i^T} \right] + qI. \tag{3.35}$$

Estas recursões reduzem para o filtro RLS exponencialmente ponderado, quando $\alpha = 1$ e $q = 0$. A função de custo associada é

$$\min_{\{x_0, n_0, \ldots, n_N\}} \left[\lambda^{N+1} w_0^T \Pi w_0^0 + \sum_{i=0}^{N} \lambda^{N-i} |d_i - U_i w_i^0|^2 + q^{-1} \sum_{i=0}^{N} \lambda^{N-i} \|n_i\|^2 \right],$$

sujeito à $w_{i+1}^0 = \alpha w_i^0 + n_i$.

3.3 Conclusão do Capítulo

Neste capítulo, estudamos o algoritmo RLS-estendido na sua forma mais geral. Iniciamos com o critério dos mínimos quadrados de forma a mostrar que o problema de estimação linear, na forma determinística ou estocástica, é equivalente. Como um caso especial do RLS-estendido consideramos um modelo particular aplicado ao rastreamento (de sinal), onde as equações do algoritmo e o modelo de espaço de estado se tornam bem mais reduzidas.

Capítulo 4

Modelo de espaço de estado para desenvolvimento do algoritmo proposto

A busca de algoritmos para filtragem adaptativa que sejam simples de implementar, robustos, rápidos, e que acompanhem eficientemente processos não estacionários continua sendo um problema atual. Os principais algoritmos derivados do LMS tendem a ser numericamente robustos e de baixa complexidade, porém são lentos. O algoritmo RLS, por outro lado, possui uma rápida taxa de convergência, mas apresenta alguns problemas de instabilidade numérica e um alto custo computacional, além disso sua capacidade de acompanhar processos não estacionários nem sempre é adequada, como mostra [8], o RLS tem em algumas situações, desempenho até pior que o do LMS.

Vários algoritmos vêm sendo propostos para acompanhar processos não estacionários. Algoritmo de passo variável, que busca variar o passo de adaptação de acordo com a situação [9, 10]; métodos que buscam descorrelacionar o sinal de entrada (por exemplo, o RLS), mas procurando minimizar problemas numéricos, como o algoritmo rápido de Newton [11], o de projeções afins [12, 13] e algumas versões do RLS rápido [14]; algoritmos com funções de custo modificadas, como o Least Mean Fourth (LMF) [15].

O estudo teórico de filtros adaptativos tradicionalmente se baseia na obtenção da curva de aprendizagem do filtro em função de seus parâmetros de projeto. A

curva de aprendizagem (isto é, o gráfico de $\mathbb{E}[e^2(n)]$ em função de n) fornece uma medida da velocidade e da precisão com que um filtro adaptativo pode reagir a variações em suas entradas. Grande parte dos trabalhos sobre análise teórica de filtros adaptativos procuram calcular a curva de aprendizagem de um determinado filtro em função das propriedades estatísticas dos sinais de entrada $\mathbf{x}(n)$ e $\mathbf{d}(n)$. As simulações são muito usadas para validar resultados teóricos, ou mesmo para extrair informações sobre a taxa de convergência ou o erro residual (em regime permanente) de um filtro adaptativo em situações para os quais não existem resultados teóricos.

Filtros adaptativos supervisionados baseados em momentos de quarta ordem foram propostos no inicio da década de 1980, com o objetivo de obter um desempenho melhor (maior velocidade de convergência e/ou menos desajuste em regime do que o tradicional algoritmo LMS, sem chegar à complexidade computacional do RLS. O primeiro exemplo de filtro baseado em momento de quarta ordem foi o Least Mean Fourth (LMF), que é um algoritmo de gradiente, bastante semelhante ao LMS, mas baseado na minimização do valor esperado da quarta potência de erro [8, 15]. Este algoritmo é motivado pela minimização da função custo $\mathbb{E}[e^4(n)]$, usando o método do gradiente estocástico para a minimização. O trabalho original [15], na verdade propõe uma familia de algoritmos, usando uma potência $p-$ésima do erro

$$sinal(e(n))|e(n)|^p, \tag{4.1}$$

para $p \geq 1$, e mostra que filtros baseados em momentos de ordem elevada podem ser vantajosos se o erro de estimação for sub-gaussiano. No entanto este trabalho já adverte para possíveis problemas de estabilidade de filtros de ordem elevada, em especial para a dependência da estabilidade com condição inicial do filtro. Além disso o artigo mostra que o LMF pode ser útil dependendo da distribuição do erro de estimação ótimo.

A prova, de fato, da estabilidade do LMF para passo suficientemente pequeno só veio em 1992, com o artigo [16] que mostra que o algoritmo LMF é estável se o passo de adaptação for suficientemente pequeno, se $d(n)$ for estritamente limitado (isto é, se existir $B > 0$ tal que $|d(n)| < B,\ \forall n$), e se os regressores produzirem uma

excitação persistente, isto é, se existirem $0 < \alpha \leq \beta < \infty$ e $N < \infty$ tais que, $\forall$,

$$\alpha \mathbf{I} \leq \sum_{k=n}^{n+N} \mathbf{x}(k)\mathbf{x}^T(k) \leq \beta \mathbf{I}$$

onde $\mathbf{I}$ é a matriz identidade $M \times M$.

4.1 O modelo de espaço de estado para o algoritmo EX-RLS

A deficiência do RLS é que ele tem um mau desempenho de rastreamento. Do ponto de vista do modelo de espaço de estado, o RLS assume implicitamente que os dados satisfazem [1]

$$\mathbf{x}(i+1) = \mathbf{x}(i) \tag{4.2}$$

$$d(i) = \mathbf{u}^T(i)\mathbf{x}(i) + v(i), \tag{4.3}$$

isto é, o estado $\mathbf{x}(i)$ é fixado sobre o tempo, e portanto, a estimativa ótima do estado $\mathbf{w}(i)$ não é possível controlar variações. Para melhorar a capacidade de controle, várias técnicas podem ser empregadas. Por exemplo, os dados podem ser exponencialmente ponderados no tempo ou uma janela truncada pode ser aplicada nos dados de treinamento (que também pode ser visto como um tipo especial de ponderação).

Como é conhecido a partir da estimativa sequencial, um modelo mais geral de espaço de estado linear é

$$\mathbf{x}(i+1) = \mathbf{A}\mathbf{x}(i) + \mathbf{n}(i) \text{ (modelo de estado)} \tag{4.4}$$

$$d(i) = \mathbf{u}^T(i)\mathbf{x}(i) + v(i) \text{ (modelo de observação)}, \tag{4.5}$$

onde $\mathbf{A}$ é a matriz de estado, $\mathbf{n}(i)$ é o ruído de estado e $v(i)$ o ruído de observação.

O modelo de estado é também chamado de equação de processo ou modelo do processo e o modelo de observação é também chamado equação de medida.

Assumimos ambos os ruídos Gaussianos, brancos, i.e.

$$\mathbb{E}[\mathbf{n}(i)\mathbf{n}(j)^T] = \begin{cases} q_1\mathbf{I}, & i = j \\ 0, & i \neq j \end{cases},$$

$$\mathbb{E}[v(i)v(j)^T] = \begin{cases} q_2, & i = j \\ 0, & i \neq j \end{cases} .$$

O modelo de estado descreve um fenômeno estocástico físico desconhecido denotado pelo vetor de estado $\mathbf{x}(i)$, $L \times 1$ como saída de um sistema dinâmico linear excitado pelo ruído branco $\mathbf{n}(i)$. O modelo de observação relaciona a saída observável $d(i)$ do sistema com o estado $\mathbf{x}(i)$. Um algoritmo de estimação para atualizar o estado estimado numa sequência de dois passos foi proposto por Kalman [17]. É assumido que $\mathbf{x}(1)$, que é o valor inicial do estado, é descorrelacionado com $\mathbf{n}(i)$ e $v(i)$, para $i \geq 1$. Os dois ruídos $\mathbf{n}(i)$ e $v(i)$ são estatisticamente independentes. Os parâmetros $\mathbf{A}, q_1$ e q_2 são conhecidos a priori.

Usa-se $\mathbf{w}(i-1)$ para denotar a estimativa ótima de $\mathbf{x}(i)$ dadas as observações $\{\mathbf{u}(1), d(1)\}, \{\mathbf{u}(2), d(2)\}, ...\{\mathbf{u}(i-1), d(i-1)\}$, e $\mathbf{w}(i)$ para denotar a estimativa ótima de $\mathbf{x}(i+1)$, dadas as observações $\{\mathbf{u}(1), d(1)\}, \{\mathbf{u}(2), d(2)\}, ..., \{\mathbf{u}(i), d(i)\}$.

Como no RLS, define-se o processo de inovação associado com $d(i)$ como

$$e(i) = d(i) - \mathbf{w}^T(i-1)\mathbf{u}(i), \tag{4.6}$$

onde $e(i)$ é o erro de predição de $d(i)$ observado e também representa a nova informação em $d(i)$. A variância do processo de inovação $e(i)$ é definida por

$$\begin{aligned} r(i) &= \mathbb{E}[e^2(i)] \\ &= \mathbb{E}[(d(i) - \mathbf{w}^T(i-1)\mathbf{u}(i))^2] \\ &= \mathbb{E}[(\mathbf{x}^T(i-1)\mathbf{u}(i) + v(i) - \mathbf{w}^T(i-1)\mathbf{u}(i))^2] \\ &= \mathbb{E}[((\mathbf{x}(i-1) - \mathbf{w}(i-1))^T\mathbf{u}(i) + v(i))^2] \\ &= \mathbf{u}^T(i)\mathbf{P}(i-1)\mathbf{u}(i) + q_2, \end{aligned}$$

onde $\mathbf{P}(i-1)$ é a matriz de correlação do erro de estado,

$$\mathbf{P}(i-1) = \mathbb{E}[(\mathbf{x}(i-1) - \mathbf{w}(i-1))(\mathbf{x}(i-1) - \mathbf{w}(i-1))^T]. \tag{4.7}$$

Para prosseguir, precisamos introduzir um conceito importante, chamado Ganho de Kalman ou simplesmente vetor ganho, e uma relação fundamental entre a estimativa ótima atual e a estimativa ótima anterior

$$\mathbf{w}(i) = \mathbf{A}\mathbf{w}(i-1) + \mathbf{k}(i)e(i), \tag{4.8}$$

que mostra que podemos calcular a estimativa média mínima quadrática $\mathbf{w}(i)$ do estado de um sistema dinâmico linear pela adição da estimativa anterior $\mathbf{w}(i-1)$, que é pré-multiplicada pela matriz de estado $\mathbf{A}$, um termo de correção igual a $\mathbf{k}(i)e(i)$.

O termo de correção é igual ao erro de predição $e(i)$ pré-multiplicado pelo vetor de ganho $\mathbf{k}(i)$. Esta equação é similar à do RLS, exceto pelo passo da pre-multiplicação da matriz de transição $\mathbf{A}$.

Além disso, pela teoria de estimação ótima de mínimos quadrados, podemos expressar o vetor de ganho $\mathbf{k}(i)$ como

$$\mathbf{k}(i) = \mathbf{A}\mathbf{P}(i-1)\mathbf{u}(i)/r(i), \tag{4.9}$$

resta apenas o problema de encontrar uma maneira recursiva de calcular a matriz $\mathbf{P}(i-1)$ de correlação do erro de estado predita. Usando a equação de Riccati, temos

$$\mathbf{P}(i) = \mathbf{A}[\mathbf{P}(i-1) - \mathbf{A}^{-1}\mathbf{k}(i)\mathbf{u}^T(i)\mathbf{P}(i-1)]\mathbf{A}^T + q_1\mathbf{I}. \tag{4.10}$$

Portanto, inicializando $\mathbf{w}(0) = 0$ e $\mathbf{P}(0) = \lambda^{-1}\mathbf{I}$, podemos calcular $r(1), \mathbf{k}(1)$ e então atualizar $\mathbf{w}(1), \mathbf{P}(1)$ com $\{\mathbf{u}(1), d(1)\}$ fornecido. A recursividade pode continuar, desde que as observações estejam disponíveis.

Na Tabela 4.1 podemos observar o resumo do algoritmo EX-RLS padrão.

Tabela 4.1: Resumo do algoritmo EX-RLS padrão

Inicializar o algoritmo, definindo

$\mathbf{P}(0) = \lambda^{-1}\mathbf{I}$

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

Para cada instante de tempo, $i = 1, 2, ...,$ computar

$r(i) = q_2 + \mathbf{u}^T(i)\mathbf{P}(i-1)\mathbf{u}(i)$

$\mathbf{k}(i) = \mathbf{A}\mathbf{P}(i-1)\mathbf{u}(i)/r(i)$

$e(i) = d(i) - \mathbf{u}^T(i)\mathbf{w}(i-1)$

$\mathbf{w}(i) = \mathbf{A}\mathbf{w}(i-1) + \mathbf{k}(i)e(i)$

$\mathbf{P}(i) = \mathbf{A}\mathbf{P}(i-1)\mathbf{A}^T - \mathbf{k}(i)\mathbf{k}^T(i)r(i) + q_1\mathbf{I}$

A derivação acima é baseada na resolução de um problema estocástico. Tem uma formulação determinística equivalente. De fato, a solução desse problema de estimação de espaço de estado equivale a resolver o seguinte problema de custo mínimo

$$\min_{\{x(1),n(1),\ldots,n(i)\}}[q_2^{-1}\sum_{j=1}^{i}|d(j)-\mathbf{u}^T(j)\mathbf{x}(j)|^2+\lambda\|\mathbf{x}(1)\|^2+q_1^{-1}\sum_{j=1}^{i}\|\mathbf{n}(j)\|^2]$$

sujeito a

$$\mathbf{x}(j+1)=\mathbf{A}\mathbf{x}(j)+\mathbf{n}(j).$$

Ou equivalentemente,

$$\min_{\{x(1),n(1),\ldots,n(i)\}}[\sum_{j=1}^{i}|d(j)-\mathbf{u}^T(j)\mathbf{x}(j)|^2+\lambda\|\mathbf{x}(1)\|^2+q^{-1}\sum_{j=1}^{i}\|\mathbf{n}(j)\|^2]$$

sujeito a

$$\mathbf{x}(j+1)=\mathbf{A}\mathbf{x}(j)+\mathbf{n}(j),$$

onde $q=\frac{q_1}{q_2}$, que indica o *trade-off* entre o erro de medida e o erro de estado. Portanto, a recursividade no algoritmo EX-RLS é equivalente à

Tabela 4.2: Resumo do algoritmo EX-RLS equivalente

Inicializar o algoritmo, definindo

$\mathbf{P}(0)=\lambda^{-1}\mathbf{I}$

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

Para cada instante de tempo, $i=1,2,...$, computar

$r(i)=1+\mathbf{u}^T(i)\mathbf{P}(i-1)\mathbf{u}(i)$

$\mathbf{k}(i)=\mathbf{A}\mathbf{P}(i-1)\mathbf{u}(i)/r(i)$

$e(i)=d(i)-\mathbf{u}^T(i)\mathbf{w}(i-1)$

$\mathbf{w}(i)=\mathbf{A}\mathbf{w}(i-1)+\mathbf{k}(i)e(i)$

$\mathbf{P}(i)=\mathbf{A}\mathbf{P}(i-1)\mathbf{A}^T-\mathbf{k}(i)\mathbf{k}^T(i)r(i)+q\mathbf{I}$

4.2 Algoritmo EX-RLS exponencialmente ponderado

Similar ao RLS, podemos introduzir um fator de esquecimento para atualizar o peso. A função de custo exponencialmente ponderado é

$$\min_{\{x(1),n(1),\ldots,n(i)\}}[\sum_{j=1}^{i}\beta^{i-j}|d(j)-\mathbf{u}^T(j)\mathbf{x}(j)|^2+\lambda\beta^i\|\mathbf{x}(1)\|^2+q^{-1}\sum_{j=1}^{i}\beta^{i-j}\|\mathbf{n}(j)\|^2]$$

sujeito a

$$\mathbf{x}(j+1) = \mathbf{A}\mathbf{x}(j) + \mathbf{n}(j),$$

onde β é um fator de esquecimento, λ é o parâmetro de regularização para controlar a norma do vetor de estado inicial, e q fornece um *trade-off* entre a variação da modelagem e a perturbação da medida. Repetindo o mesmo argumento anteriormente, temos o algoritmo RLS-estendido exponencialmente ponderado

Na Tabela 4.3 podemos observar o resumo do algoritmo EX-RLS exponencialmente ponderado.

Tabela 4.3: Resumo do algoritmo EX-RLS exponencialmente ponderado

Inicializar o algoritmo, definindo

$\mathbf{P}(0) = \lambda^{-1}\mathbf{I}$

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

Para cada instante de tempo, $i = 1, 2, ...,$ computar

$r(i) = \beta^i + \mathbf{u}^T(i)\mathbf{P}(i-1)\mathbf{u}(i)$

$\mathbf{k}(i) = \mathbf{A}\mathbf{P}(i-1)\mathbf{u}(i)/r(i)$

$e(i) = d(i) - \mathbf{u}^T(i)\mathbf{w}(i-1)$

$\mathbf{w}(i) = \mathbf{A}\mathbf{w}(i-1) + \mathbf{k}(i)e(i)$

$\mathbf{P}(i) = \mathbf{A}\mathbf{P}(i-1)\mathbf{A}^T - \mathbf{k}(i)\mathbf{k}^T(i)r(i) + \beta^i q\mathbf{I}$

É claro que o EX-RLS é um caso especial do EX-RLS exponencialmente ponderado quando $\beta = 1$. Ao falarmos do algoritmo EX-RLS, estaremos nos referindo sempre ao exponencialmente ponderado.

4.3 O algoritmo proposto

Nesta seção mostraremos o algoritmo proposto, baseado no algoritmo RNQ e no RLS-estendido, desenvolvidos no Capítulo 3.

Para o desenvolvimento do algoritmo proposto EX-RNQ, iniciaremos com a função custo dada por

$$J_n = \sum_{i=1}^{n} \{\beta^{n-i}[e(i)^{2j}]\}, \tag{4.11}$$

sendo j e n inteiros positivos, com $j \geq 1$ e $0 \ll \beta < 1$, fator de esquecimento, $e(i) = d(i) - \mathbf{w}^T(n)\mathbf{u}(i)$, $1 \leq i \leq n$, $\mathbf{u}(i) = [u_i, u_{i-1}, ..., u_{i-L+1}]$ e o vetor peso $\mathbf{w}(n) = [w_{0,n}, w_{1,n}, ..., w_{L-1,n}]^T$, sendo L a ordem do filtro.

O peso ótimo $\mathbf{w}(n)$ da função custo J_n é solução da equação

$$\nabla J_n = -\sum_{i=1}^{n} \left\{\beta^{n-i}\mathbf{u}(i)2j(d(i) - \mathbf{w}^T(n)\mathbf{u}(i))^{2j-1}\right\} = 0, \tag{4.12}$$

onde ∇ é o gradiente da função custo J_n em relação a $\mathbf{w}$.

Usando a aproximação [15]

$$(d(i) - \mathbf{w}^T(n)\mathbf{u}(i))^{2j-1} \approx d^{2j-1}(i) - (2j-1)d^{2j-2}(i)\mathbf{w}^T(n)\mathbf{u}(i),$$

escrevemos

$$\begin{aligned} \nabla J_n \quad \approx & -2j[\sum_{i=1}^{n}[\beta^{n-i}d(i)^{2j-1}\mathbf{u}(i)] \\ & -(2j-1)\sum_{i=1}^{n}[\beta^{n-1}d(i)^{2j-2}\mathbf{u}(i)\mathbf{u}^T(i)]\mathbf{w}(n)]. \end{aligned} \tag{4.13}$$

Definindo

$$\mathbf{z}_n = -2j[\sum_{i=1}^{n}[\beta^{n-i}d(i)^{2j-1}\mathbf{u}(i)], \tag{4.14}$$

$$\mathbf{\Phi}_n = (2j-1)\sum_{i=1}^{n}[\beta^{n-1}d(i)^{2j-2}\mathbf{u}(i)\mathbf{u}^T(i)]\mathbf{w}(n), \tag{4.15}$$

escrevemos as equação (4.13) como

$$\nabla J_n \approx -2j[\mathbf{z}_n - \mathbf{\Phi}_n \mathbf{w}(n)]. \tag{4.16}$$

Portanto, o valor ótimo do vetor peso $\mathbf{w}(n)$, para que a função custo (4.11) atinja o mínimo satisfaz a equação

$$\mathbf{w}(n) = \mathbf{\Phi}_n^{-1} \mathbf{z}_n. \tag{4.17}$$

Isolando o termo referente a $i = n$ na equação (4.15), escrevemos

$$\Phi_n = \beta \Phi_{n-1} + [(2j-1)d^{2j-2}(n)]\mathbf{u}(n)\mathbf{u}^T(n), \tag{4.18}$$

e aplicando o Lema de inversão de matrizes para a equação (4.18) com as identificações $\mathcal{A} = \mathbf{\Phi_n},\ \mathcal{B}^{-1} = \beta \mathbf{\Phi}_{n-1},\ \mathcal{C} = \mathbf{u}(n)$ e $\mathcal{D}^{-1} = (2j-1)d^{2j-2}(n)$, teremos

$$\Phi_n^{-1} = \beta^{-1}\Phi_{n-1}^{-1} - \frac{\beta^{-1}\Phi_{n-1}^{-1}\mathbf{u}(n)\mathbf{u}^T(n)\beta^{-1}\Phi_{n-1}^{-1}}{((2j-1)d^{2j-2}(n))^{-1} + \beta^{-1}\mathbf{u}^T(n)\Phi_{n-1}^{-1}\mathbf{u}(n)}. \tag{4.19}$$

Fazendo a identificação $\mathbf{P}(n) = \mathbf{\Phi}_n^{-1}$, escrevemos

$$\mathbf{P}(n) = \beta^{-1}\mathbf{P}(n-1) - \beta^{-1}\mathbf{g}(n)\mathbf{u}^T(n)\mathbf{P}(n-1), \tag{4.20}$$

onde

$$\mathbf{g}(n) = \frac{\beta^{-1}\mathbf{P}(n-1)\mathbf{u}(n)}{((2j-1)d^{2j-2}(n))^{-1} + \beta^{-1}\mathbf{u}^T(n)\mathbf{P}(n-1)\mathbf{u}(n)}.$$

Baseado nos desenvolvimentos dos algoritmos EX-RLS e RNQ, segue então que, para o algoritmo proposto EX-RNQ, definimos de forma equivalente, via modelo de espaço de estado, o vetor ganho

$$\mathbf{G}(i) = \beta^{-1}\mathbf{A}\mathbf{P}(i-1)\mathbf{u}(i)/R(i), \tag{4.21}$$

onde $R(i) = ((2j-1)d^{2j-2}(i))^{-1} + \beta^{-1}\mathbf{u}^T(i)\mathbf{P}(i-1)\mathbf{u}(i)$, com a matriz $\mathbf{P}(i)$ associada dada por

$$\mathbf{P}(i) = \beta^{-1}\mathbf{A}\mathbf{P}(i-1)\mathbf{A}^T - \beta^{-1}\mathbf{G}(i)\mathbf{G}^T(i)R(i) + ((2j-1)d^{2j-2}(i))^{-1}q\mathbf{I}, \tag{4.22}$$

para uma constante q (*trade-off*) e uma matriz $\mathbf{A}$ definidos em termos do espaço de estado.

Para atualizarmos o vetor peso $\mathbf{w}(i)$, da equação (4.17), escrevemos o fator $\mathbf{z}(n)$, de forma equivalente como na equação (4.18) para escrever

$$\mathbf{z}(n) = \beta \mathbf{z}_{n-1} + d^{2j-1}(n)\mathbf{u}(n). \tag{4.23}$$

Daí, a equação (4.17) resulta em

$$\mathbf{w}(n) = \beta\mathbf{P}(n)\mathbf{z}(n-1) + d^{2j-1}(n)\mathbf{P}(n)\mathbf{u}(n). \tag{4.24}$$

Substituindo $\mathbf{P}(n)$ da equação (4.20) na equação (4.24) e com algumas manipulações matemáticas, chegamos à expressão

$$\mathbf{w}(n) = \mathbf{w}(n-1) + \mathbf{g}(n)\left[\frac{d(n)}{(2j-1)} - \mathbf{u}^T(n)\mathbf{w}(n-1)\right], \tag{4.25}$$

com isso, para o algoritmo proposto EX-RNQ, definimos o processo de inovação, associado com $d(i)$ e a estimativa ótima atual, via modelo espaço de estado, respectivamente, como

$$\varepsilon(i) = \frac{d(i)}{(2j-1)} - \mathbf{u}^T(i)\mathbf{w}(i-1), \tag{4.26}$$

$$\mathbf{w}(i) = \mathbf{A}\mathbf{w}(i-1) + \mathbf{G}(i)\varepsilon(i). \tag{4.27}$$

Na Tabela 4.4 podemos observar o resumo do algoritmo proposto EX-RNQ.

Tabela 4.4: Resumo do algoritmo proposto EX-RNQ

Inicializar o algoritmo, definindo
$\mathbf{P}(0) = \lambda^{-1}\mathbf{I}$, λ fator de esquecimento
$\mathbf{w}(0) = \mathbf{0}$
Para cada instante de tempo, $i = 1, 2, ...$, computar

$R(i) = ((2j-1)d^{2j-2}(i))^{-1} + \beta^{-1}\mathbf{u}^T(i)\mathbf{P}(i-1)\mathbf{u}(i)$
$\mathbf{G}(i) = \beta^{-1}\mathbf{A}\mathbf{P}(i-1)\mathbf{u}(i)/R(i)$
$\varepsilon(i) = \frac{d(i)}{(2j-1)} - \mathbf{u}^T(i)\mathbf{w}(i-1)$
$\mathbf{w}(i) = \mathbf{A}\mathbf{w}(i-1) + \mathbf{G}(i)\varepsilon(i)$
$\mathbf{P}(i) = \beta^{-1}\mathbf{A}\mathbf{P}(i-1)\mathbf{A}^T - \beta^{-1}\mathbf{G}(i)\mathbf{G}^T(i)R(i) + ((2j-1)d^{2j-2}(i))^{-1}q\mathbf{I}$

4.4 Experimentos computacionais

4.4.1 Identificação de sistema

Nesta simulação, verificamos que nosso algoritmo funciona corretamente, identificando um filtro curto passa-baixa FIR, como mostra a Figura 4.1. Este filtro é caracterizado por uma resposta ao impulso $\mathbf{h}$, comprimento $\mathbf{L} = 15$, frequência de corte $f_c = 0,5\pi$ para as primeiras 1000 amostras e $f_c = 0,2\pi$ para as últimas

1000 amostras. Usamos u_i como um sinal Gaussiano distribuído a ser filtrado por **h**. A saída do filtro adaptativo é denotada por y_i e o sinal desejado d_i é obtido por meio da filtragem do sinal de entrada u_i por **h**. Executamos 100 simulações de Monte Carlo. O algoritmo proposto mostra uma convergência rápida e com menor erro final quando comparado com outros algoritmos. As curvas de aprendizagem são plotadas na Figura 4.2 para comparar a eficiência dos algoritmos.

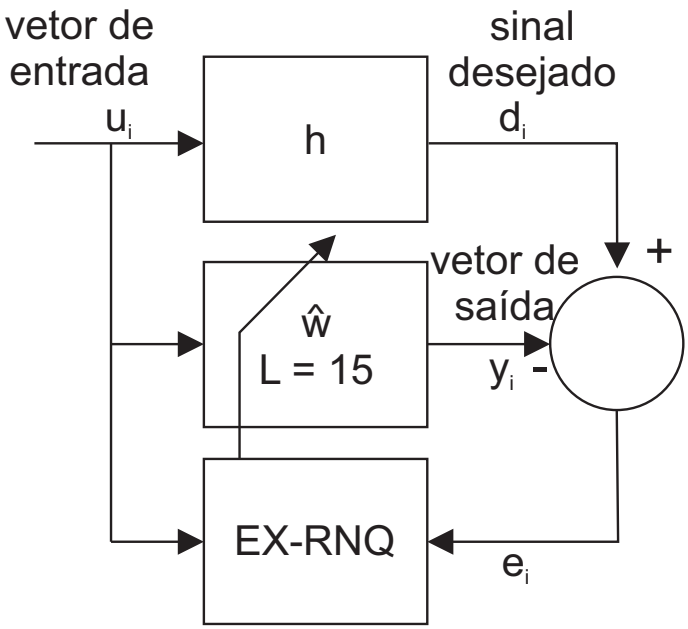

Figura 4.1: Filtro FIR com L = 15 como o comprimento do filtro adaptativo.

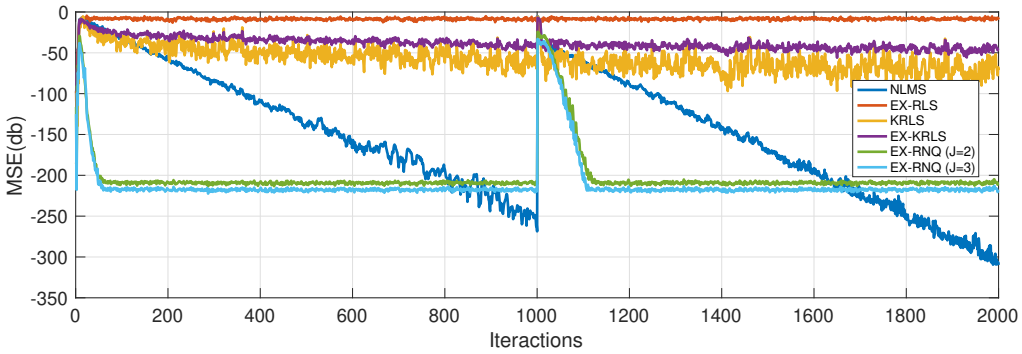

Figura 4.2: Curvas de aprendizagem dos algoritmos NLMS, EX-RLS, KRLS, EX-KRLS e EX-RNQ (EX-RNQ, $j = 2, 3$) no problema de identificação de sistema. O sistema é um filtro passa-baixa FIR L=15, frequência de corte $f_c = 0,5\pi$. Na iteração=1001, $f_c = 0,2\pi$.

4.4.2 Aplicação ao rastreamento de canais de Rayleigh

O desvanecimento de Rayleigh [2] é um modelo estatístico para o efeito de um ambiente de propagação em um sinal de rádio, tal como aquele usado por dispositivos sem fio. Em comunicações sem fio, múltiplas reflexões de sinais transmitidos chegarão a um receptor por causa de canais de comunicação complexos. As reflexões são com distorções de amplitude e fase distintas, que podem se puramente aleatórias. O sinal global recebido é a soma combinada de todas as reflexões. Com base nas fases relativas das reflexões, os sinais podem somar de forma construtiva ou destrutiva no receptor. Além disso, o canal físico pode mudar ao longo do tempo, por exemplo, o receptor está em movimento (como um telefone celular), pelo que estas interferências destrutivas e construtivas irão variar ao longo do tempo. Esse fenômeno é conhecido como desvanecimento do canal.

O desvanecimento de Rayleigh é um modelo razoável quando muitos objetos no ambiente espalham o sinal de rádio antes dele chegar ao receptor. Pelo Teorema do limite central, se houver dispersão suficiente, a resposta ao impulso do canal será bem modelada como um processo Gaussiano, independentemente da distribuição dos componentes individuais. Se não houver componente dominante na dispersão, então tal processo terá média zero e fase uniformemente distribuída entre 0 e 2π rad. O envelope da resposta do canal será, portanto, uma distribuição de Rayleigh. Em outras palavras, se a resposta ao impulso de um único canal de desvanecimento é

$$h(n) = x(n)\delta(n - \tau), \tag{4.28}$$

então a distribuição de Rayleigh afirma que

$$p(|x(n)|) = \frac{|x(n)|}{\sigma^2} \exp(\frac{-|x(n)|^2}{2\sigma^2}),$$

$$p(\angle x(n)) = \frac{1}{2\pi}, \tag{4.29}$$

onde $x(n)$ é um número complexo variante no tempo que modela as variações de tempo no canal, $|x(n)|$ denota a magnitude, $\angle x(n)$ denota a fase, τ é o atraso do canal, σ é o modo da distribuição. A média e a variância podem ser expressas por

$$\mathbb{E}[|x(n)|] = \sigma\sqrt{\pi/2},$$

$$\mathbb{E}\{[|x(n)| - \mathbb{E}[|x(n)|]]^2\} = \frac{4-\pi}{2}\sigma^2. \tag{4.30}$$

Além disso, a função de autocorrelação normalizada com movimento a uma velocidade constante é modelada por uma função de Bessel de ordem zero do primeiro tipo

$$\mathbb{E}[x(n)x(n-k)] = \mathcal{J}_0(2\pi f_D T_s k),\ k = ..., -1, 0, 1, ..., \tag{4.31}$$

onde T_s é o período amostral da sequência, f_D é a frequência Doppler máxima do canal de desvanecimento de Rayleigh, e a função $\mathcal{J}_0(.)$ é definida como

$$\mathcal{J}_0(y) = \frac{1}{\pi}\int_0^{\pi} \cos(y\sin\theta)d\theta. \tag{4.32}$$

A frequência Doppler f_D está relacionada à velocidade do receptor em relação ao transmissor v e a frequência da portadora f_c como segue:

$$f_D = \frac{v f_c}{c}, \tag{4.33}$$

onde c é a velocidade da luz, $c = 3 \times 10^8\ m/s$.

No canal de desvanecimento multicaminho de Rayleigh, o número de caminho é escolhido como $M = 5$, a frequência Doppler máxima $f_D = 100\ Hz$, e a taxa de amostragem, $T_s = 0,8\ \mu s$ (assim é um canal de desvanecimento lento com o mesmo desvanecimento para todos os caminhos). Todos os coeficientes de derivação são gerados de acordo com o modelo de Rayleigh mas apenas a parte real é utilizada nesta experiência.

Uma distribuição Gaussiana branca em série temporal (com potência unitária) é enviada através deste canal, corrompido com ruído aditivo Gaussiano branco (com variância $\sigma^2 = 0,001$) e então a não linearidade de saturação $y = \tanh(x)$ é aplicada, onde x é a saída do canal de Rayleigh. Todo o canal não linear é tratado como uma caixa-preta, e somente a entrada e a saída são conhecidas.

Para o nosso experimento, consideremos o problema de rastreamento não linear multicaminho do canal de Rayleigh e comparemos o desempenho do algoritmo EX-RNQ. Testamos o algoritmo proposto duas vezes, repetindo os mesmos parâmetros definidos pelos algoritmos EX-RLS e EX-KRLS. Além disso, o desempenho dos algoritmos NLMS, EX-RLS, KRLS e EX-KRLS [2, 3] foram incluídos para comparação.

Tabela 4.5: Parâmetros dos algoritmos no canal de rastreamento de Rayleigh

Algoritmos	Parâmetros
NLMS	$\epsilon = 10^{-3},\ \eta = 0,25$
EX-RLS	$\alpha = 0,9999999368, q = 3,26 \times 10^{-7},\ \beta = 0,995, \lambda = 10^{-3}$
KRLS	$\lambda = 0,01,\ a = 1$
EX-KRLS	$\alpha = 0,999998,\ q = 10^{-4},\ \beta = 0,995,\ \lambda = 0,01,\ a = 1$
EX-RNQ* $(j = 2,3)$	$\alpha = 0,9999999368,\ q = 3,26 \times 10^{-7},\ \lambda = 10^{-3}$
EX-RNQ** $(j = 2,3)$	$\alpha = 0,999998,\ q = 10^{-4},\ \lambda = 0,01$

Para observar o desempenho dos algoritmos, os parâmetros foram definidos na Tabela 4.5. Além disso, geramos 1000 símbolos para o experimento e 200 experimentos por Monte Carlo.

Na Tabela 4.5, os parâmetros para o algoritmo NLMS são, respectivamnete, o fator de regularização e o tamanho do passo; no algoritmo EX-RLS, temos o elemento da diagonal da matriz $\alpha\mathbf{I}$, o *trade-off* q, o fator de esquecimento e o parâmetro de regularização, assim como para os outros algoritmos. O parâmetro a é o parâmetro do kernel.

As curvas de aprendizagem são mostradas nas figuras 4.3 e 4.4 para comparar a eficiência dos algoritmos, observando sua convergência e desajuste que claramente mostra que o algoritmo EX-RNQ tem uma melhor capacidade de rastreamento que os algoritmos EX-RLS, KRLS and EX-KRLS. Os últimos 100 valores nas curvas de aprendizagem são usados para calcular o MSE, que está listado nas tabelas 4.6 e 4.7. As tabelas 4.6 e 4.7 mostram, de fato, que o algoritmo EX-RNQ funciona melhor em termos de rastreamento, onde vemos uma diferença de aproximadamente 0,32-dB com respeito ao algoritmo EX-KRLS, quando consideramos a quarta potência do erro, mas com convergência mais rápida e especialmente quando consideramos a sexta potência do erro, vemos uma diferença de aproximadamente 4,54-dB com os parâmetros definidos para EX-RNQ* e uma diferença de aproximadamente 0,46-dB com respeito ao algoritmo EX-KRLS , quando consideramos a quarta potência do error, mas com convergência mas rápida e especialmente quando consideramos a sexta potência do error, vemos uma diferença de aproximadamente 4,68-dB com os parâmetros definidos por EX-RNQ**. Os resultados mostram que o algoritmo EX-RNQ é uma escolha alternativa para os algoritmos estendidos existentes, pois

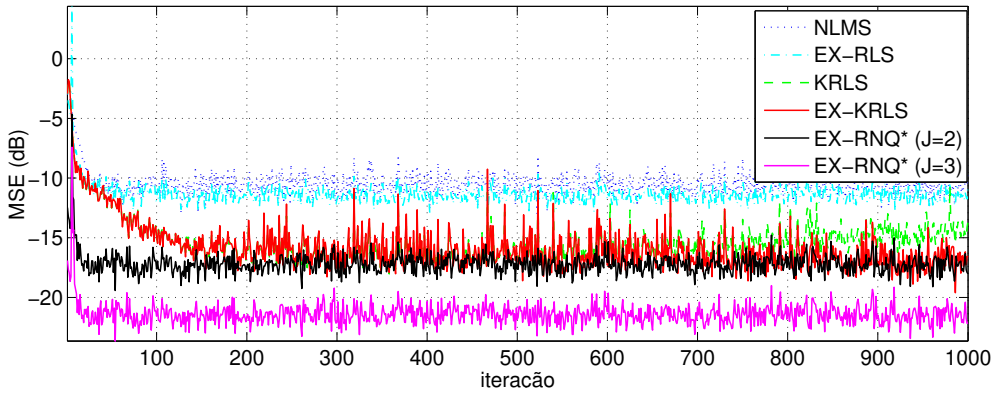

Figura 4.3: Conjunto de curvas de aprendizagem dos algoritmos NLMS, EX-RLS, KRLS, EX-KRLS and EX-RNQ (EX-RNQ*, $j = 2, 3$) no canal de rastreamento multicaminho de Rayleigh.

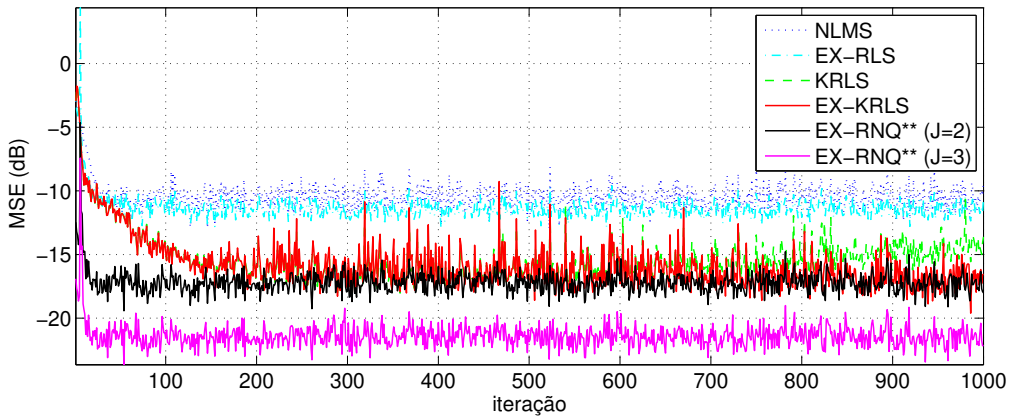

Figura 4.4: Conjunto de curvas de aprendizagem dos algoritmos NLMS, EX-RLS, KRLS, EX-KRLS and EX-RNQ (EX-RNQ**, $j = 2, 3$) no rastreamento no canal multicaminho de Rayleigh.

tem melhor desempenho em velocidade de convergência e menor erro médio.

Lembrando que a citação EX-RNQ* significa que os parâmetros α, q, λ foram os mesmos usados no algoritmo EX-RLS e EX-RLS** significa que os parâmetros α, q, λ foram os mesmo usados no algoritmo EX-KRLS.

Tabela 4.6: Desempenho de comparação no rastreamento no canal de Rayleigh (*)

Algorithm	**MSE (dB)**
NLMS	$-10,6964 \pm 0,87432$
EX-RLS	$-11,5608 \pm 0,55673$
KRLS	$-14,8808 \pm 0,72817$
EX-KRLS	$-17,098 \pm 1,266$
EX-RNQ* $(j = 2)$	$-17,4196 \pm 0,61465$
EX-RNQ* $(j = 3)$	$-21,6402 \pm 0,63135$

Tabela 4.7: Desempenho de comparação no rastreamento no canal de Rayleigh (**)

Algorithm	**MSE (dB)**
NLMS	$-10,6964 \pm 0,87432$
EX-RLS	$-11,5608 \pm 0,55673$
KRLS	$-14,8808 \pm 0,72817$
EX-KRLS	$-17,0945 \pm 1,2677$
EX-RNQ** $(j = 2)$	$-17,4196 \pm 0,61465$
EX-RNQ** $(j = 3)$	$-21,6402 \pm 0,63135$

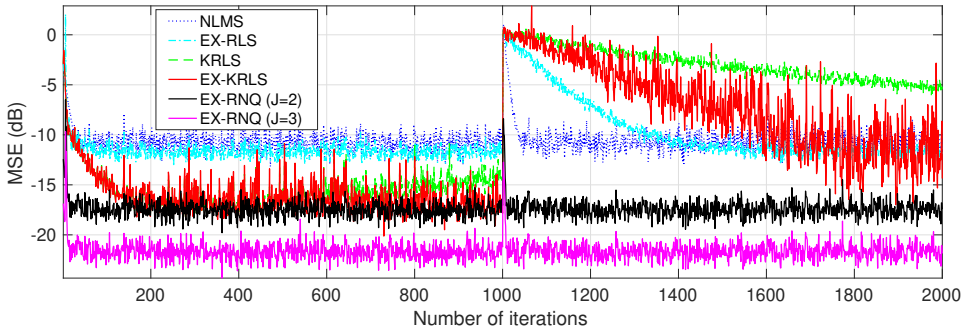

Figura 4.5: Curvas de aprendizagem dos algritmos NLMS, EX-RLS, KRLS, EX-KRLS and EX-RNQ (EX-RNQ, $j = 2, 3$) no rastreamento no canal de Rayleigh. Na iteração $n = 1001$, frequência máxima Dopller foi mudada para $f_D = 50\ Hz$.

Capítulo 5

Conclusão e trabalhos futuros

Neste trabalho, apresenta-se o desempenho de um novo algoritmo, atualizando os pesos de um filtro adaptativo usando uma função de custo baseada na potência par do erro aplicado ao modelo de espaço de estado, como um caso particular do Filtro de Kalman. Este algoritmo é adequado para um modelo não-linear com um desvanecimento lento e uma pequena variação no estado. Os resultados mostraram que podemos ter uma boa aplicabilidade em extensões não-lineares do algoritmo EX-RLS. As aplicações em comunicações digitais são provavelmente as mais apropriadas. O resultado é o algoritmo EX-RNQ, dado na Tabela 4.4, semelhante ao algoritmo EX-RLS, mas com melhor desempenho em termos de velocidade de convergência e desajuste, em comparação com os algoritmos mostrados nos experimentos, como mostrado em Figuras 4.4 e 4.3, e nas tabelas 4.6 e 4.7.

Para trabalhos futuros, já obtivemos resultados significativos quando modificamos a função de custo (4.11) para uma função de custo exponencialmente ponderada, melhorando ainda mais a convergência e o desajuste, obtendo também uma curva de aprendizagem com a variância do erro mais suave. Além disso, queremos tornar o algoritmo, de fato, um algoritmo aplicável ao rastreamento de sinal, de modo prático, fazendo uma simulação no mundo real.

Para algoritmo EX-RNL, consideramos a função custo não linear

$$J_n = \sum_{j=1}^{m} \sum_{i=1}^{n} \{\beta^{n-i}[e(i)]^{2j}\}, \tag{5.1}$$

sendo m e n inteiros positivos, e $0 \ll \beta < 1$ o fator de esquecimento, $e(i) = d(i) - \mathbf{u}^T(i)\mathbf{w}(n),\ 1 \leq i \leq n$.

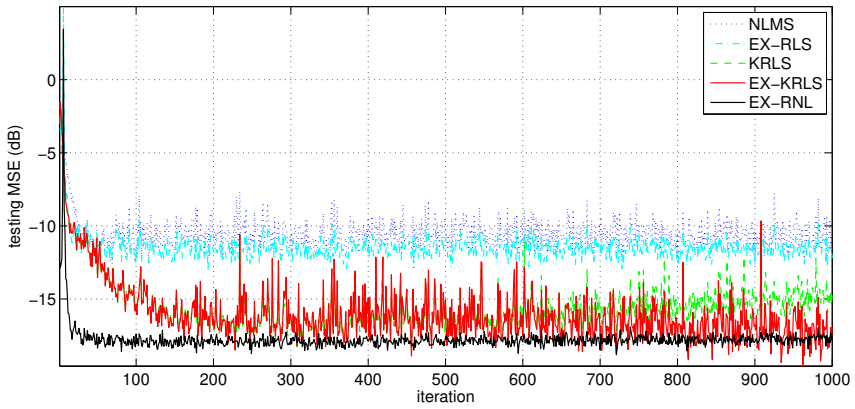

Figura 5.1: Curvas de aprendizagem dos algritmos NLMS, EX-RLS, KRLS, EX-KRLS e EX-RNL no problema de rastreamento no canal de Rayleigh.

Tabela 5.1: Comparação de desempenho no rastreamento no canal de Rayleigh

Algorithm	**MSE (dB)**
NLMS	-10.6964 ± 0.87432
EX-RLS	-11.5608 ± 0.55673
KRLS	-14.8808 ± 0.72817
EX-KRLS	-17.098 ± 1.266
EX-RNL	-17.7225 ± 0.26335

Apêndice

5.1 Código do algoritmo (Matlab)

```
 % rayleigh channel tracking
% Weifeng Liu
% Sep. 2007.
%
% Description:
% compare extended KRLS and KRLS in rayleigh channel tracking
% learning curves
%
% Outside functions used
% LMS2, RLS, EX_RLS, KRLS and EX_KRLS


close all
clear all
%clc

var_n = 1e-3;
sqn = sqrt(var_n);

sampleInterval = 0.8e-6; % sampling frequency is 1MHz
numberSymbol = 1000;
dopplerFrequency = 100; % Doppler frequency
trainSize = numberSymbol;

epsilon = 1e-3;

channelLength = 5;
channel  = zeros(channelLength,trainSize);



alpha = besselj(0,2*pi*dopplerFrequency*sampleInterval);
q = 1-alpha*alpha;


ensembleLearningCurveLms2 = zeros(trainSize,1);
%ensembleLearningCurveRls = zeros(trainSize,1);
ensembleLearningCurveExrls = zeros(trainSize,1);
ensembleLearningCurveKrls = zeros(trainSize,1);
```

```
ensembleLearningCurveExkrls = zeros(trainSize,1);
ensembleLearningCurveEX-RNQ2 = zeros(trainSize,1); %EX-RNQ($j=2$)
ensembleLearningCurveEX-RNQ3 = zeros(trainSize,1);%EX-RNQ($j=3$)
% time delay (embedding) length
inputDimension = channelLength;

%Nonlinearity parameter
typeNonlinear = 1;
paramNonlinear = 2;

%Kernel parameters
typeKernel = 'Gauss';
paramKernel = .05;


L = 500;
%L = 300;
%L = 10;

disp([num2str(L),' Monte Carlo simulations. Please wait...'])

for k = 1:L
    disp(k);

    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %
    %       Data Formatting
    %
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

    for i=1:channelLength;
      channel(i,:) = rayleigh(sampleInterval,numberSymbol,
                     dopplerFrequency);
    end

    channel = real(channel);

     % Input signal
    inputSignal = randn(1,trainSize + channelLength);
    noise = sqn*randn(1,trainSize);


    %Input training signal with data embedding
    trainInput = zeros(inputDimension,trainSize);
    for kk = 1:trainSize
        trainInput(:,kk) = inputSignal(kk:kk+inputDimension-1);
    end

    %Desired training signal
    trainTarget = zeros(trainSize,1);
    for ii=1:trainSize
        trainTarget(ii) = trainInput(:,ii)'*channel(:,ii);
    end

    trainTarget = trainTarget + noise';
```

```
    %Pass through the nonlinearity
    trainTarget = nlG(trainTarget,paramNonlinear,typeNonlinear);


     %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %
    %               Normalized LMS 2
    %
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    regularizationFactorLms2 = epsilon;
    stepSizeWeightLms2 = .25;
    stepSizeBiasLms2 = 0;

    [weightVectorLms2,biasTermLms2,learningCurveLms2]= ...
     LMS2(trainInput,trainTarget,regularizationFactorLms2,
          stepSizeWeightLms2,stepSizeBiasLms2);

    %=========end of Normalized LMS 2===============



%     %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%     %
%     %               RLS
%     %
%     %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%     pInitialRls = (1/epsilon)*eye(inputDimension);
%     forgettingFactorRls = 1;
%
%     [weightVectorRls,learningCurveRls]= ...
%      RLS(trainInput,trainTarget,pInitialRls,forgettingFactorRls);
%
%     %%=========end of RLS===============


    % %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %
    %             Ex-RLS
    %
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    pInitialExrls = (1/epsilon)*eye(inputDimension);
    forgettingFactorExrls = .995;
    alphaParameterExrls = alpha;
    qFactorExrls = q;
    % flagLearningCurve = 1;
    [weightVectorExrls,learningCurveExrls]= ...
     EX_RLS(trainInput,trainTarget,pInitialExrls,forgettingFactorExrls,
            alphaParameterExrls,qFactorExrls);
    %=========end of ex-rls===============



    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %
    %              KRLS
```

```
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

regularizationFactorKrls = 0.01;
forgettingFactorKrls = 1;

% flagLearningCurve = 1;
[expansionCoefficientKrls,learningCurveKrls] = ...
 KRLS(trainInput,trainTarget,typeKernel,paramKernel,
      regularizationFactorKrls,forgettingFactorKrls);

% =========end of KRLS================


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%               Ex-KRLS
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
alphaParameterExkrls = 0.99999;
regularizationFactorExkrls = 0.01;
forgettingFactorExkrls = .995;
qFactorExkrls = 1e-4;
% flagLearningCurve = 1;

[expansionCoefficientExkrls,learningCurveExkrls] = ...
 EX_KRLS(trainInput,trainTarget,typeKernel,paramKernel,
         alphaParameterExkrls,regularizationFactorExkrls,
         forgettingFactorExkrls,qFactorExkrls);
%=========end of Ex_KRLS================



%Adicionado em 27/05/2017
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%               Ex-RNQ2
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
lambda=0.96;
[y,e,w1]=EX-RNQ2(trainInput,trainTarget,lambda);

learningCurveEX-RNQ2 = [e.^2]';
weightVectorEX-RNQ2 = w1;
%=========end of Ex-RNQ2================


 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%               Ex-RNQ3
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
lambda=0.96;
[y,e,w1]=EX-RNQ3(trainInput,trainTarget,lambda);
```

```
    learningCurveEX-RNQ3 = [e.^2]';
    weightVectorEX-RNQ3 = w1;
    %=========end of Ex-RNQ===============

 ensembleLearningCurveLms2 = ensembleLearningCurveLms2
                             + learningCurveLms2;
 %ensembleLearningCurveRls = ensembleLearningCurveRls
    %                        + learningCurveRls;
 ensembleLearningCurveExrls = ensembleLearningCurveExrls
                              + learningCurveExrls;

 ensembleLearningCurveKrls = ensembleLearningCurveKrls
                             + learningCurveKrls;
 ensembleLearningCurveExkrls = ensembleLearningCurveExkrls
                               + learningCurveExkrls;

 ensembleLearningCurveEX-RNQ2 = ensembleLearningCurveEX-RNQ2
                                   + learningCurveEX-RNQ2;
 ensembleLearningCurveEX-RNQ3 = ensembleLearningCurveEX-RNQ3
                                     + learningCurveEX-RNQ3;
end

ensembleLearningCurveLms2 = 10*log10(ensembleLearningCurveLms2/L);
%ensembleLearningCurveRls = 10*log10(ensembleLearningCurveRls/L);
ensembleLearningCurveExrls = 10*log10(ensembleLearningCurveExrls/L);
ensembleLearningCurveKrls = 10*log10(ensembleLearningCurveKrls/L);
ensembleLearningCurveExkrls = 10*log10(ensembleLearningCurveExkrls/L);
ensembleLearningCurveEX-RNQ2 = 10*log10(ensembleLearningCurveEX-RNQ2/L);
ensembleLearningCurveEX-RNQ3 = 10*log10(ensembleLearningCurveEX-RNQ3/L);

mse_Lms2 = mean(ensembleLearningCurveLms2(end-100:end));
%mse_Rls = mean(ensembleLearningCurveRls(end-100:end));
mse_Exrls = mean(ensembleLearningCurveExrls(end-100:end));
mse_Krls = mean(ensembleLearningCurveKrls(end-100:end));
mse_Exkrls = mean(ensembleLearningCurveExkrls(end-100:end));
mse_EX-RNQ2 = mean(ensembleLearningCurveEX-RNQ(end-100:end));
mse_EX-RNQ3 = mean(ensembleLearningCurveEX-RNQ3(end-100:end));
disp([num2str(mse_Lms2),' ',num2str(mse_Exrls),' ',num2str(mse_Krls),'
     ',num2str(mse_Exkrls),' ',num2str(mse_EX-RNQ2),
     num2str(mse_EX-RNQ3),' '])

figure
plot(1:trainSize,ensembleLearningCurveLms2,'b:','LineWidth',1)
hold on
plot(1:trainSize,ensembleLearningCurveExrls,'c-.','LineWidth',1)
plot(1:trainSize,ensembleLearningCurveKrls,'g--','LineWidth',1)
plot(1:trainSize,ensembleLearningCurveExkrls,'r-','LineWidth',1)
plot(1:trainSize,ensembleLearningCurveEX-RNQ2,'k-','LineWidth',1)
plot(1:trainSize,ensembleLearningCurveEX-RNQ3,'m-','LineWidth',1)


xlabel('iteration')
ylabel('dB')
grid
axis tight
```

```
legend('NLMS','EX-RLS','KRLS','EX-KRLS', 'EX-RNQ2', 'EX-RNQ3')

set(gca, 'FontSize', 14);
set(gca, 'FontName', 'Arial');
%legend('Conditional Information')
xlabel('iteration'),ylabel('testing MSE (dB)')



disp('====================================')

mseMean = mean(ensembleLearningCurveLms2(end-100:end));
mseStd = std(ensembleLearningCurveLms2(end-100:end));

disp([num2str(mseMean),'+/-',num2str(mseStd)]);
%
% mseMean = mean(ensembleLearningCurveRls(end-100:end));
% mseStd = std(ensembleLearningCurveRls(end-100:end));
%
% disp([num2str(mseMean),'+/-',num2str(mseStd)]);

mseMean = mean(ensembleLearningCurveExrls(end-100:end));
mseStd = std(ensembleLearningCurveExrls(end-100:end));

disp([num2str(mseMean),'+/-',num2str(mseStd)]);

mseMean = mean(ensembleLearningCurveKrls(end-100:end));
mseStd = std(ensembleLearningCurveKrls(end-100:end));

disp([num2str(mseMean),'+/-',num2str(mseStd)]);

mseMean = mean(ensembleLearningCurveExkrls(end-100:end));
mseStd = std(ensembleLearningCurveExkrls(end-100:end));

disp([num2str(mseMean),'+/-',num2str(mseStd)]);


mseMean = mean(ensembleLearningCurveEX-RNQ2(end-100:end));
mseStd = std(ensembleLearningCurveEX-RNQ2(end-100:end));

disp([num2str(mseMean),'+/-',num2str(mseStd)]);

mseMean = mean(ensembleLearningCurveEX-RNQ3(end-100:end));
mseStd = std(ensembleLearningCurveEX-RNQ3(end-100:end));


disp([num2str(mseMean),'+/-',num2str(mseStd)]);
disp('====================================')
```

Referências Bibliográficas

[1] S. Haykin, Adaptive filter theory, Prentice Hall, (1991).

[2] W. Liu, J. C. Príncipe and S. Haykin, Kernel adaptive filtering, John Wiley & Sons, 2010.

[3] W. Liu, I. Park, Y. Wang and J. C. Príncipe, Extended Kernel Recursive Least Squares Algorithm, IEEE Trans.on Signal Processing, vol. 57, n0 10, 2009.

[4] C. C. S. Silva, E. C. Santana. E. Aguiar, M. Araújo and A. K. Barros, An adaptive recursive algorithm based on non-quadratic function of the error, Signal Processing, volume 92, número 4 (2012) 853 - 856.

[5] B. Farhang-Boroujeny, Adaptive Filter Theory and Application, John Wiley & Sons, (1998).

[6] B. Widrow and Samuel D. Stearns. Adaptive Signal Processing, Prentice-Hall signal processing series, (1985).

[7] A. H. Sayed, Adaptive filters, John Wiley & Sons, (2008).

[8] SAYED, A. H. Fundamentals of Adaptive Filtering. [S.l.]: Wiley-Interscience, 2003.

[9] KWONG, R. H.; JOHSTON, E. W. A variable step size LMS algorithm. IEEE Trans. Signal Processing, v. 40, n. 7, p. 1633-1642, jul. 1992.

[10] ABOULNASR, T.; MAYYAS, K. A robust variable step-size LMS-type algorithm. IEEE Trans. Signal Processing, v. 45, n. 3, p. 631-639, mar. 1997.

[11] MOUSTAKIDES, G. V.; THEODORIDES, S. Fast Newton transversal filters-a new class of adaptive estimation algorithm. IEEE Trans. Signal Processing, v. 39, n. 10, p. 2184-2193, out. 1991.

[12] GAY, S. L.; TAVATHIA, S. The fast affine projection algorithm. In: 1996 IEEE International Conference on Acoustics, Speech, and Signal Processing.

[13] ALMEIDA, S. J. M. de et al. A stochastic model for the convergence behavior of the affine projection algorithm for gaussian inputs. In: Proceedings of the 2003 IEEE International Conference on Acoustics, speech and Signal Processing (CDROM). [S.l.:s.n.], 2003.VI, p. VI 313-VI 316.

[14] MIRANDA, M. D.; GERKEN, M. A hybrid least squares QR-lattice algorithm using a priori erros. IEEE Trans. Signal Processing, v. 45, n. 12, p. 2900-2911, dez. 1997.

[15] WALACH, E.; WIDROW, B. The least mean fourth (LMF) adaptive algorithm and its family. IEEE Trans. Inform. Theory, IT-30, n. 2, p. 275-283, mar. 1984.

[16] SETHARES, W. A. Adaptive algorithms with nonlinear data and error functions. IEEE Trans. Signal Processing, v. 40, n. 9, p. 2199-2206, set. 1992.

[17] R. E. Kalman. A new approach to linear filtering and prediction problems, Transactions of the ASME- Jounal of Basic Engineering, 82:35-45, 1960.

[18] P. Zhu, B. Chen, and J. C. Príncipe, A novel extended kernel recursive least squares algorithm, *Neural Networks*, vol. 32, pp. 349 – 357, 2012.

[19] K. Li and J. C. Príncipe, The kernel adaptive autoregressive-moving-average algorithm, *IEEE Transactions on Neural Networks and Learning Systems*, vol. 27, no. 2, pp. 334–346, Feb 2016.

[20] F. Ding, P. X. Liu, and G. Liu, Auxiliary model based multi-innovation extended stochastic gradient parameter estimation with colored measurement noises, *Signal Processing*, vol. 89, no. 10, pp. 1883 – 1890, 2009.

[21] J. Pan, X. Jiang, X. Wan, and W. Ding, A filtering based multi-innovation extended stochastic gradient algorithm for multivariable control systems, *International Journal of Control, Automation and Systems*, vol. 15, no. 3, pp. 1189–1197, Jun 2017.

[22] Y. Liu, L. Yu, and F. Ding, Multi-innovation extended stochastic gradient algorithm and its performance analysis, *Circuits, Systems and Signal Processing*, vol. 29, no. 4, pp. 649–667, Aug 2010.

[23] X. Wang and F. Ding, Convergence of the auxiliary model-based multi-innovation generalized extended stochastic gradient algorithm for box-jenkins systems, *Nonlinear Dynamics*, vol. 82, no. 1, pp. 269–280, Oct 2015.

[24] Y. Mao, F. Ding, and E. Yang, Adaptive filtering based multi-innovation gradient algorithm for input nonlinear systems with autoregressive noise, *International Journal of Adaptive Control and Signal Processing*, vol. 31, no. 10, pp. 1388–1400.

[25] Y. Mao and F. Ding, A novel parameter separation based identification algorithm for hammerstein systems, *Applied Mathematics Letters*, vol. 60, pp. 21 – 27, 2016.

[26] Y. Mao, F. Ding, A. Alsaedi, and T. Hayat, Adaptive filtering parameter estimation algorithms for hammerstein nonlinear systems, *Signal Processing*, vol. 128, pp. 417 – 425, 2016.

[27] F. Ding, P. X. Liu, and G. Liu, "Auxiliary model based multi-innovation extended stochastic gradient parameter estimation with colored measurement noises,"Signal Processing, vol. 89, no. 10, pp. 1883 - 1890, 2009.

[28] J. Pan, X. Jiang, X. Wan, and W. Ding, "A filtering based multi-innovation extended stochastic gradient algorithm for multivariable control systems,"International Journal of Control, Automation and Systems, vol. 15, no. 3, pp. 1189-1197, Jun 2017.

[29] X. Wang and F. Ding, "Convergence of the auxiliary model-based multi-innovation generalized extended stochastic gradient algorithm for box-jenkins systems,"Nonlinear Dynamics, vol. 82, no. 1, pp. 269-280, Oct 2015.

[30] Y. Mao, F. Ding, and E. Yang, "Adaptive filtering-based multi-innovation gradient algorithm for input nonlinear systems with autoregressive noise,"International Journal of Adaptive Control and Signal Processing, vol. 31, no. 10, pp. 1388-1400.

[31] Y. Mao, F. Ding, A. Alsaedi, and T. Hayat, "Adaptive filtering parameter estimation algorithms for hammerstein nonlinear systems,"Signal Processing, vol. 128, pp. 417-425, 2016.

[32] Y. Mao and F. Ding, "A novel parameter separation based identification algorithm for hammerstein systems,"Applied Mathematics Letters, vol. 60, pp. 21-27, 2016.

[33] W. A. Sethares. Adaptive Algorithms with Nonlinear data and Error Functions. IEEE Trans. Signal Processing, v. 40, n. 9, p. 2199-2206, set. 1992.

[34] M. Chansarkar, U. Desai, B. Rao, A comparative study of the approxi- mate RLS with LMS and RLS algorithms, Proceedings of IEEE Region 10 Conference TENCON-89, Bombay, (1989), 255 - 258.

[35] G. Carayannis, D. Manolakis, N. Kalouptsidis, A fast sequential algo- rithm for least-squares filtering and prediction, Acoustics, Speech and Signal Processing, IEEE Transactions on 31, (1983), 1394 - 1402.

[36] J. Cioffi, T. Kailath, Fast, recursive-least-squares transversal filters for adaptive filtering, Acoustics, Speech and Signal Processing, IEEE Trans- actions on 32, (1984), 304 - 337.

[37] A. Barros, J. Principe, Y. Takeuchi, C. Sales, 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), 879-885

[38] M. Fliess, C. Join and H. S.-Ramirez, Non-linear estimation is easy, Int. J. Modelling Identification Control, 4, 1 (2008) 12-27.

[39] Anum Ali, Anis-ur-Rehman, R. Liaqat Ali, An improved gain vector to enhance convergence characteristics of recursive least squared algorithm, International Journal of Hybrid Information Technology, 4, 2, April (2011).

[40] E. Eweda, Comparison of RLS, LMS and sign algorithms for tracking randomly time-varying channels,. IEEE Transactions on Signal Processing, vol. 42, pp. 2937.2944, Nov. (1994).

[41] J. Arenas-García, M. Martínez-Ramón, A. Navia-Vázquez, and A. R. Figueiras-Vidal, Plant identication via adaptive combination of transversal filters, Signal Processing, vol. 86, pp. 2430-2438, Sep. (2006).

[42] J. Arenas-García, A. R. Figueiras-Vidal, and A. H. Sayed, Mean-square performance of a convex combination of two adaptive filters, IEEE Transactions on Signal Processing, vol. 54, pp. 1078-1090, Mar. (2006).

[43] J. Arenas-García, A. R. Figueiras-Vidal, and A. H. Sayed, Tracking properties of a convex combination of two adaptive lters, in Proc. Of IEEE 13th Workshop on Statistical Signal Processing (SSP 05), pp. 109-114 (2005).

[44] M. T. M. Silva and V. H. Nascimento, Improving the tracking capability of adaptive filters via convex combination, IEEE Transactions on signal processing, (2008).

[45] M. M. Chansarkar and U. B. Desai, A fast approximate RLS algorithm, IEEE Tencon (1993).

[46] E. Eleftheriou and D. D. Falconer, Tracking properties and steady-state performance of RLS adaptive filter algorithms, IEEE Trans. Acoust., Speech, Signal Processing, vol. ASSP-34, no. 5, pp.1097 -1109 (1986).