



UNIVERSIDADE FEDERAL DO MARANHÃO
CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

JONAS DE JESUS BARROS

**ALGORITMO RECURSIVO NÃO QUADRÁTICO
AUTORREGULADO**

SÃO LUÍS-MA

2026

Jonas de Jesus Barros

Algoritmo Recursivo Não Quadrático Autorregulado

Tese de Doutorado apresentada ao Programa de Pós-Graduação em Engenharia Elétrica (área de concentração: Automação e Controle), como parte dos requisitos necessários para a obtenção do Título de Doutor em Engenharia Elétrica.

Universidade Federal do Maranhão - UFMA

Centro de Ciências Exatas e Tecnologia

Programa de Pós-Graduação em Engenharia Elétrica

Orientador: Prof. Dr. Allan Kardec Dualibe Barros Filho

Coorientador: Prof. Dr. Marcus Vinicius de Sousa Lopes

São Luís-MA

2026

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

de Jesus Barros, Jonas.

Algoritmo Recursivo Não Quadrático Autorregulado /
Jonas de Jesus Barros. - 2026.
71 p.

Coorientador(a) 1: Marcus Vinicius de Sousa Lopes.
Orientador(a): Allan Kardec Dualibe Barros Filho.
Tese (Doutorado) - Programa de Pós-graduação em
Engenharia Elétrica/ccet, Universidade Federal do
Maranhão, São Luis - Ma, 2026.

1. Filtro Adaptativo. 2. Regularização. 3.
Velocidade de Convergência. I. Dualibe Barros Filho,
Allan Kardec. II. de Sousa Lopes, Marcus Vinicius. III.
Título.

Jonas de Jesus Barros

Algoritmo Recursivo Não Quadrático Autorregulado

Tese de Doutorado apresentada ao Programa de Pós-Graduação em Engenharia Elétrica (área de concentração: Automação e Controle), como parte dos requisitos necessários para a obtenção do Título de Doutor em Engenharia Elétrica.

Trabalho aprovado em, de de 2026:

Prof. Dr. Allan Kardec Dualibe Barros Filho
Orientador

Prof. Dr. Marcus Vinicius de Sousa Lopes
Coorientador

Prof. Dr. João Viana Fonseca Neto
Interno

**Prof. Dr. Ewaldo Eder Carvalho
Santana**
Interno

Prof. Dra. Marta de Oliveira Barreiros
Externo

Prof. Dr. Jonathan Araujo Queiroz
Externo

São Luís-MA
2026

Agradecimentos

Ao professor orientador Allan Kardec Duailibe Barros Filho , pela orientação segura, pela motivação, apoio e dedicação, fundamentais para início, desenvolvimento e conclusão dessa tese. Agradeço ao professor Marcus Vinicius de Sousa Lopes pela coorientação e participação direta, além das discussões, correções e sugestões para conclusão desse trabalho. Ao professor José Mairton Barros da Silva que me conduziu ao PIB dando início a essa empreitada.

A todos aos meus colegas do PIB pelo compartilhamento de suas experiências companheirismo e amizades. Aos colegas de trabalho do Centro de Lançamento de Alcântara que torceram por mim.

Aos meus filhos Jordan Silva Barros, João Pedro Silva Barros, Juliana Silva Barros, Jônatas Silva Barros e à mãe dos meus filhos Ana Paula dos Santos Silva Barros que sempre me incentivaram.

A Deus pela oportunidade que me foi dada.

“Se eu vi mais longe, foi por estar sobre ombros de gigantes.”
(Isaac Newton)

Resumo

A filtragem adaptativa tem sido extensivamente estudada para uma ampla gama de aplicações como economia, engenharia biomédica, equalização de canais, sistema de controle e telecomunicações. O problema de sinais não estacionários constitui um dos maiores desafios no contexto de filtros adaptativos, uma vez que exige simultaneamente alta velocidade de convergência, boa capacidade de rastreamento e estabilidade sendo motivação de propostas novos algoritmos de filtragem e variantes de algoritmos bem estabelecidos. A variabilidade das propriedades estatísticas do sinal impõe a necessidade de mecanismos de detecção para variações abruptas do sinal de entrada, aumentando a complexidade computacional dos algoritmos. Este trabalho apresenta o algoritmo Recursivo Não Quadrático Autorregulado (RNQA) que deriva de uma família de algoritmos baseados em potências pares de erro onde a regularização dinâmica da matriz de autocorrelação é feita continuamente pelo quadrado do erro instantâneo do algoritmo, caracterizando um mecanismo de ajuste adaptativo na fase de atualização da matriz, capaz de responder efetivamente em ambientes não estacionários. Uma configuração de identificação de sistemas foi realizada em um ambiente não estacionário, caracterizado por mudanças abruptas no parâmetro desconhecido, permitiu avaliar a capacidade de rastreamento do algoritmos proposto. O RNQA apresentou melhor resultado de desempenho em termos de velocidade de convergência em comparação ao RNQ padrão e algoritmo do filtro de Kalman numa estrutura de identificação de sistemas.

Palavras-chaves: Filtro adaptativo, Regularização, Velocidade de convergência.

Abstract

Adaptive filtering has been extensively studied for a wide range of applications such as economics, biomedical engineering, channel equalization, control systems, and telecommunications. The problem of non-stationary signals constitutes one of the greatest challenges in the context of adaptive filters, since it simultaneously requires high convergence speed, good tracking capability, and stability, motivating the proposal of new filtering algorithms and variants of well-established algorithms. The variability of the signal's statistical properties imposes the need for detection mechanisms for abrupt statistical variations in the input signal, increasing the computational complexity of the algorithm. This work presents the Recursive Non-Quadratic Self-Regulated (RNQA) algorithm, which derives from a family of algorithms based on even-power errors where the dynamic regularization of the autocorrelation matrix is continuously performed by the square of the instantaneous error of the algorithm, characterizing an adaptive adjustment mechanism in the matrix update phase, capable of responding effectively in non-stationary environments. A system identification configuration was performed in a non-stationary environment, characterized by abrupt changes in the unknown parameter, allowing the evaluation of the tracking capability of the proposed algorithm. RNQA showed better performance results in terms of convergence speed compared to the standard RNQ and Kalman filter algorithm in a system identification framework.

Keywords: Adaptive filter, Regularization, Convergence speed.

Lista de ilustrações

Figura 1 – Estrutura transversal do combinador linear adaptativo.	13
Figura 2 – Uma superfície quadrática tridimensional. O erro quadrático está plotado na vertical, w_0 e w_1 no intervalo -1 a 1	14
Figura 3 – Gráficos das funções $(e^2 + e^4 + e^6)$ e e^2 , onde podemos ver a maior inclinação da primeira, no intervalo $[-1; 1]$	37
Figura 4 – Estrutura do filtro adaptativo RNQA com configuração identificação de sistemas, sendo u_i o vetor de entrada, $\mathbf{h}$ um filtro FIR, d_i o sinal desejado, e_i o sinal de erro e y_i a saída do filtro.	47
Figura 5 – Comparação das curvas de aprendizagem do RNQA com $m = 5$ com base em e_n^2 , o Filtro de Kalman para identificação do sistema e o RNQ com $(m = 15)$, usando um fator de regularização fixo de $\beta = 10^{-4}$. a) As simulações foram conduzidas com um sistema livre de ruído. b) As simulações foram conduzidas com ruído gaussiano aditivo com SNR = 60 dB.	48

Lista de tabelas

Tabela 1	–	Resumo do Algoritmo RNQ para uma potência par do erro	29
Tabela 2	–	Equações do Algoritmo RNQA	42
Tabela 3	–	Número aproximado de iterações para convergência dos algoritmos RNQA, RNQ e Kalman. Sistema sem ruído	49
Tabela 4	–	Número aproximado de iterações para convergência dos algoritmos RNQA, RNQ e Kalman. Sistema com ruído	49

Siglas

ADALINE Neurônio Linear Adaptativo (*Adaptive Linear Neuron*). 4

EMSE Excesso de erro quadrático médio(*Mean squared error excess*). 8

FIR Resposta ao Impulso Finita (*Finite Impulse Response*). 9, 47

LMF quarto mínimo médio (*Least Mean Fourth*). 37

LMS Mínimos Quadrados Médios (*Least Mean Squares*). 4, 7, 8, 14, 15

MSE Erro Médio Quadrático (*Mean Square Error*). 34, 37, 43

NLMS Mínimos Quadrados Médios Normalizados (*Normalized Least Mean Square*). 8

RLS Mínimos Quadrados Recursivos (*Recursive Least Squares*). 4–9, 15, 16, 21–26, 33–35, 38, 40, 42, 51

RNQ Recursivo Não Quadrático. 8–10, 26, 29, 33–36, 49–51

RNQA Recursivo Não Quadrático Autorregulado. 9, 10, 36, 37, 40–43, 46, 48–51

Lista de símbolos

d_n	Sinal desejado
y_n	Sinal de saída do filtro adaptativo
e_n	Erro de estimação
$\mathbf{u}_n$	Vetor de entrada
$\mathbf{w}(n)$	Vetor peso
$\mathbf{w}^*$	Vetor peso ótimo
L	Ordem do filtro
μ	Passo de adaptação do algoritmo LMS
λ	Fator de esquecimento do algoritmo RLS
δ	Fator de regularização inicial
Φ_n^{-1}	Matriz inversa da autocorrelação
$\mathbf{P}_n$	Matriz inversa da autocorrelação
$\mathbf{R}$	Matriz do sinal de entrada
$\mathbf{z}_n$	Vetor de correlação cruzada
ζ_n	Quantidade a ser minimizada pelo mínimos quadrados
∇	Operador gradiente da função custo
$E[\cdot]$	Operador esperança matemática
$\mathbf{I}$	Matriz identidade
$(\cdot)^T$	Transposta de vetor ou matriz
$\ \cdot\ $	Norma Euclidiana
φ_n	Matriz de correlação do vetor de entrada do RLS
$\mathbf{k}_n$	Vetor ganho do algoritmo RLS
$\xi(n)$	Função de custo a ser minimizada pelo RLS
MSE	Erro quadrático médio (<i>Mean Square Error</i>)
$EMSE$	Excesso de erro quadrático médio
$\mathbf{v}_n$	Vetor desvio
$\mathbf{K}_n$	Matriz de correlação do vetor desvio
ϵ_n	Erro a <i>priori</i>
$\xi_{\min}$	Mínimo MSE
Tr	Traço da matriz
$\xi_{\min}$	Mínimo MSE

Sumário

1	INTRODUÇÃO	1
1.1	Um contexto histórico	1
1.2	Estado da arte	6
1.3	Motivação	9
1.4	Objetivos	9
1.4.1	Objetivo Geral	9
1.4.2	Objetivos Específicos	10
2	FILTRAGEM ADAPTATIVA	11
2.1	Introdução	11
2.2	O Combinador linear adaptativo	12
2.3	Algoritmos de gradiente estocástico	13
2.4	Algoritmos de mínimos quadrados	15
2.5	Algoritmo de mínimos quadrados recursivo	15
2.5.1	Dedução do algoritmo RLS	15
2.5.2	A Regularização	16
2.5.3	Algoritmo RLS ponderado exponencialmente	19
2.5.4	Atualização do vetor peso	20
2.5.5	Convergência do algoritmo RLS	21
2.5.6	Conclusão	25
3	ALGORITMO BASEADO EM UMA ÚNICA POTÊNCIA PAR DO ERRO (RNQ)	26
3.1	Introdução	26
3.2	Dedução do algoritmo	26
3.2.1	Atualização do vetor peso	28
3.2.2	Resumo do algoritmo	29
3.3	Convergência do algoritmo	29
3.3.1	O comportamento médio do vetor peso no algoritmo	30
3.3.2	Matriz de correlação do vetor desvio	30
3.3.3	Curva de aprendizagem	32
3.3.4	Análise do tempo de aprendizagem	33
3.3.5	Desajuste	34
3.3.6	Análise comparativa	34
3.3.7	Complexidade computacional	35

4	ALGORITMO RNQ BASEADO NA SOMA DE POTÊNCIAS PA- RES DO ERROS AUTORREGULADO (RNQA)	36
4.1	Introdução	36
4.2	Dedução do algoritmo	36
4.2.1	Atualização do vetor peso	40
4.2.2	Matriz de autocorrelação normalizada	41
4.2.3	Análise do vetor ganho do algoritmo RNQA	42
4.2.4	Interpretação do efeito da matriz de covariância normalizada pelo quadrado do erro no algoritmo RNQA	43
4.2.5	Conclusão	46
4.3	Resultados	47
4.3.1	Identificação de sistema usando um filtro FIR	47
4.3.2	Discussão	48
4.3.3	Conclusão	50
5	CONCLUSÃO E TRABALHOS FUTUROS	51
	REFERÊNCIAS	52
	ANEXOS	54
	ANEXO A – O FILTRO DE KALMAN COMO ESTIMADOR DE PARÂMETROS	55

1 Introdução

1.1 Um contexto histórico

A teoria da filtragem adaptativa emerge da convergência entre diferentes áreas do conhecimento, incluindo estatística, engenharia elétrica, controle e neurociência. Sua evolução reflete uma mudança profunda na forma de tratar problemas de estimação em ambientes incertos, passando de abordagens estáticas para métodos recursivos capazes de operar em tempo real.

O ponto de partida dessa trajetória remonta ao trabalho de Carl Friedrich Gauss, no início do século XIX, no contexto da astronomia e da geodesia. Ao estudar a determinação de órbitas celestes a partir de medições ruidosas, Gauss introduziu o método dos mínimos quadrados, baseado na minimização da soma dos erros quadráticos.

Durante grande parte do século XIX e início do século XX, esse método foi amplamente utilizado em aplicações científicas e de engenharia, sendo resolvido por meio de manipulações algébricas diretas, frequentemente com auxílio de tabelas numéricas e cálculos manuais. Nesse período, o problema era tipicamente formulado de forma estática: um conjunto completo de observações era coletado e, somente então, processado em bloco para obtenção da solução. Mesmo com os avanços promovidos por autores como Pierre-Simon Laplace, que aprofundaram a interpretação probabilística dos erros, a abordagem permanecia essencialmente manual.

Nos primórdios da computação particularmente entre as décadas de 1930 e 1940, o cenário tecnológico era marcado por limitações severas de hardware, inexistência de arquiteturas padronizadas e uma predominância quase absoluta de abordagens analíticas. Com o trabalho de Alan Turing ([TURING, 1936](#)), matemático e lógico britânico, Sobre Números Computáveis, com uma Aplicação ao problema de decisão, introduziu o conceito teórico de uma máquina universal (máquina de Turing). Reconhecido como um dos pais da computação moderna, liderou esforços no centro de Bletchley Park para decifrar mensagens criptografadas pelo sistema Enigma, utilizado pela Alemanha nazista. Esse fato antecipa conceitos fundamentais que mais tarde seriam centrais no processamento de sinais, como a automação de procedimentos algorítmicos, o tratamento sistemático de dados e a necessidade de soluções eficientes sob restrições computacionais severas.

Na década de 1940, Norbert Wiener, atuando nas áreas de engenharia elétrica, controle e comunicações, em instituições como o MIT Radiation Laboratory, desenvolveu a teoria da filtragem ótima. Seu trabalho consolidou a aplicação da estatística no processamento de sinais, formulando o problema de estimação em problemas como predição de

sinais e controle sob ruído. Seu trabalho seminal está consolidado no livro *Extrapolation, Interpolation, and Smoothing of Stationary Time Series* (WIENER, 1949), no qual o problema de estimação é formulado como a minimização do erro quadrático médio,

$$J(\mathbf{w}) = \mathbb{E}[e(n)^2]$$

Essa formulação conecta diretamente estatística e álgebra linear por meio das equações matriciais normais,

$$\mathbf{R}\mathbf{w}_o = \mathbf{p}$$

Esse ponto marca a consolidação da estrutura matemática que ainda hoje define a filtragem adaptativa e a base da estimação linear ótima. Entretanto, sua aplicação prática é limitada, pois requer o conhecimento prévio das estatísticas dos sinais, como a matriz de autocorrelação e o vetor de correlação cruzada, além de envolver a inversão de matriz, o que pode ser computacionalmente oneroso e inviável em tempo real ou em ambientes não estacionários.

A teoria da filtragem adaptativa possui raízes profundas na tentativa de modelar matematicamente os mecanismos de aprendizagem observados em sistemas biológicos. Desde as primeiras formulações de neurônios artificiais até os modernos algoritmos de estimação recursiva, observa-se uma evolução conceitual que conecta diretamente neurociência, inteligência artificial e processamento estatístico de sinais.

No campo interdisciplinar emergente da cibernética, em 1943, McCulloch e Pitts (MCCULLOCH; PITTS, 1949) e Hebb (HEBB, 1949) estabeleceram os fundamentos da aprendizagem em sistemas artificiais. Esses trabalhos introduziram conceitos que conectavam biologia, matemática e engenharia, especialmente a idealização de adaptação baseada em dados. Eles propuseram um modelo abstrato de neurônio baseado em uma combinação linear de sinais de entrada seguida por uma operação de decisão não linear. Essa formulação introduziu a concepção de que sistemas complexos poderiam emergir da interconexão de unidades simples cujos parâmetros, denominados pesos sinápticos, podem ser ajustados. Hebb introduziu um princípio de aprendizagem baseado na correlação entre atividades neurais. Nesse modelo, o neurônio realiza uma combinação linear das entradas seguida por uma função de ativação

$$y = f\left(\sum_{i=1}^L w_i x_i\right)$$

sendo que x_i representam os sinais de entrada, w_i são os pesos sinápticos e $f(\cdot)$ é uma função não linear que modela o mecanismo de disparo do neurônio. De forma simplificada, esse princípio estabelece que conexões entre neurônios devem se fortalecer quando os sinais

de entrada e saída ocorrem simultaneamente. Embora originalmente proposto no contexto da neurociência, esse princípio antecipou um conceito central da filtragem adaptativa: a atualização de parâmetros baseada em relações estatísticas observadas nos dados.

Esse modelo introduziu dois conceitos fundamentais: a representação linear ponderada das entradas, a possibilidade de adaptação dos pesos sinápticos estabelecendo a base matemática para a modelagem de sistemas capazes de aprender a partir de dados. Embora originalmente proposto no contexto da neurociência, esse princípio antecipou um conceito central da filtragem adaptativa: a atualização de parâmetros baseada em relações estatísticas observadas nos dados. Hebb propôs um princípio de aprendizagem baseado na correlação entre atividades neurais. A chamada regra de Hebb pode ser expressa de forma simplificada como

$$\Delta w_i \propto x_i y$$

sendo a representação da atividade do neurônio pré-sináptico dada por x_i e y a atividade do neurônio pós-sináptico. Esse princípio estabelece que a força de uma conexão sináptica aumenta quando os neurônios conectados são ativados simultaneamente. Do ponto de vista matemático, essa regra representa uma forma primitiva de adaptação baseada em correlação estatística.

A regra de Hebb exerceu forte influência sobre o desenvolvimento posterior de algoritmos de aprendizagem adaptativa, pois introduziu o conceito fundamental de que os parâmetros de um sistema podem ser ajustados automaticamente em função dos dados observados.

Na década de 1950, Rosenblatt ([ROSENBLATT, 1958](#)), no artigo “*A Probabilistic Model for Information Storage and Organization*”, apresentou o *perceptron*, um sistema capaz de aprender a classificar padrões por meio do ajuste iterativo de pesos. O perceptron introduziu conceitos fundamentais de aprendizagem supervisionada e mostrou que sistemas artificiais poderiam modificar seus parâmetros automaticamente em resposta ao erro observado durante o processo de aprendizagem, modelado por

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

sendo $\mathbf{x}$ o vetor de entrada e $\mathbf{w}$ representa os pesos sinápticos.

Apesar de sua simplicidade, o *perceptron* introduziu conceitos fundamentais para o desenvolvimento posterior de algoritmos adaptativos, incluindo a de atualização iterativa de parâmetros com base em um erro observado.

A história da aprendizagem adaptativa teve início no início da década de 1960, com o trabalho de Widrow e Hoff ([WIDROW; HOFF, 1960](#)). Buscando desenvolver sistemas eletrônicos capazes de aprender diretamente a partir de dados, os pesquisadores

introduziram o modelo conhecido como Neurônio Linear Adaptativo (*Adaptive Linear Element* - [ADALINE](#)). Diferentemente do perceptron, esse modelo utilizava uma saída linear, $y(n) = \mathbf{w}^T(n)\mathbf{x}(n)$, e definia o processo de aprendizagem por meio da minimização do erro quadrático médio, $J = \mathbb{E}[e^2(n)]$ entre o sinal desejado e a saída do sistema.

O [ADALINE](#) estabelece a formulação do problema como a minimização do erro quadrático médio. Uma abordagem alternativa foi proposta por Widrow e Hoff por um algoritmo baseado em um método de descida do gradiente $\nabla J(\mathbf{w})$ aplicado à função custo do erro quadrático médio.

O insight central de Widrow foi perceber que como a esperança estatística, que na prática não é conhecida, ela poderia ser aproximada utilizando apenas o erro instantâneo disponível em cada iteração. Essa observação levou ao Algoritmo dos Mínimos Quadrados Médios (*Least Mean Squares*—[LMS](#)), no qual os coeficientes do sistema são atualizados iterativamente utilizando uma estimativa estocástica do gradiente do erro médio.

Essa formulação resultou em um algoritmo simples e recursivo capaz de aprender diretamente a partir dos dados observados, permitindo adaptação contínua em ambientes dinâmicos. Isso transformou um problema de otimização estatística que não exigiria o conhecimento explícito de esperanças estatísticas e matrizes de correlação em um algoritmo simples e recursivo capaz de aprender diretamente a partir dos dados observados. Mais importante ainda, essa estrutura permite que o sistema se ajuste continuamente à medida que novos dados são observados, característica essencial em ambientes nos quais as propriedades estatísticas dos sinais não permanecem constantes ao longo do tempo.

Apesar de sua simplicidade, o [LMS](#) pode apresentar convergência lenta quando os sinais de entrada possuem forte correlação estatística ou quando o sistema a ser estimado varia rapidamente no tempo, comprometendo sua capacidade de rastreamento. Essas limitações motivaram o desenvolvimento de algoritmos mais sofisticados baseados no critério de mínimos quadrados. Entre eles destaca-se o algoritmo dos Mínimos Quadrados Recursivos (*Recursive Least Squares*—[RLS](#)) surge como a síntese natural da evolução a fundamentação estatística que implementa uma solução recursiva para o problema de mínimos quadrados ponderados utilizando um fator de esquecimento que permite dar maior peso às observações mais recentes.

A introdução desse fator de esquecimento, $0 < \lambda \leq 1$, fornece um mecanismo natural para lidar com ambientes não estacionários, permitindo que o algoritmo adapte seus parâmetros de forma mais rápida às mudanças nas propriedades do sistema. O algoritmo [RLS](#) busca minimizar a função custo

$$J_n(\mathbf{w}) = \sum_{i=1}^n \lambda^{n-i} e_i^2$$

Esse algoritmo apresenta convergência significativamente mais rápida que o [LMS](#) e

melhor capacidade de rastreamento em sistemas variantes no tempo ([HAYKIN, 1991](#)).

Relação histórica entre o filtro de Kalman e o algoritmo RLS

Um aspecto fundamental na evolução da teoria de estimação adaptativa diz respeito à relação histórica e conceitual entre o filtro de Kalman e o algoritmo [RLS](#). Embora frequentemente apresentados como métodos distintos na literatura de processamento de sinais, sua origem revela uma conexão profunda.

O filtro de Kalman, introduzido por Rudolf Emil Kalman ([KALMAN, 1960](#)), surgiu no contexto da estimação de estados em sistemas dinâmicos, motivado por aplicações em navegação, controle e rastreamento. Seu objetivo consistia em estimar de forma ótima o estado de um sistema linear sujeito a ruído, utilizando uma formulação recursiva que permitia atualizar a estimativa a cada nova observação. Kalman observou que a solução ótima de um problema de mínimos quadrados poderia ser implementada de forma sequencial, eliminando a necessidade de processamento em lote.

Paralelamente, o problema de estimação de parâmetros baseado no critério de mínimos quadrados já era conhecido desde os trabalhos clássicos de Gauss e posteriormente formalizado no contexto estatístico por Wiener. No entanto, essas abordagens eram essencialmente não recursivas e, portanto, pouco adequadas para aplicações em tempo real.

A conexão entre essas duas linhas de desenvolvimento tornou-se evidente quando se considerou o problema de estimação de parâmetros constantes dentro da estrutura do filtro de Kalman. Ao modelar o vetor de parâmetros como um sistema dinâmico trivial, descrito $\mathbf{w}(n+1) = \mathbf{w}(n)$, e a relação de observação como $d(n) = \mathbf{x}^T(n)\mathbf{w}(n) + v(n)$, obtém-se um caso particular do filtro de Kalman cuja solução resulta em equações estruturalmente idênticas às do algoritmo [RLS](#).

Sob essa perspectiva, o algoritmo [RLS](#) pode ser interpretado como uma especialização do filtro de Kalman para o problema de estimação de parâmetros constantes. Historicamente, isso implica que o desenvolvimento do [RLS](#) foi fortemente influenciado pelo surgimento da estimação recursiva proposta por Kalman, ao mesmo tempo em que incorpora diretamente o critério de mínimos quadrados herdado da tradição estatística. Essa dupla origem confere ao [RLS](#) uma posição singular na teoria de filtragem adaptativa: ele representa simultaneamente a forma recursiva exata do método dos mínimos quadrados ponderados e um caso particular de um estimador ótimo mais geral. Como consequência, o algoritmo combina elevada eficiência estatística com capacidade de adaptação em tempo real, características que o tornam particularmente adequado para aplicações em ambientes dinâmicos e não estacionários.

Do ponto de vista histórico, pode-se afirmar que o filtro de Kalman precede o [RLS](#),

tendo estabelecido o paradigma da estimação recursiva.

O [RLS](#) surge, então, como a materialização desse paradigma no contexto específico do processamento de sinais. Essa relação evidencia que o avanço da área não ocorreu por substituição de métodos, mas sim por progressiva integração, culminando em algoritmos capazes de unir optimalidade teórica e viabilidade computacional em aplicações práticas.

1.2 Estado da arte

Em processamento de sinais aleatórios quando as características estatísticas desses sinais não são conhecidas a solução é o emprego de filtros adaptativos para efetuar suas predição e estimação ([HAYKIN, 2014](#)).

O termo filtro é usado para descrever um sistema de hardware ou software, cuja finalidade pode ser tanto para recuperar informação contaminada por ruído ou simplesmente selecionar partes de interesse de um sinal, como por exemplo, componentes de uma certa banda de frequência. Filtros usados para tais propósitos podem ser de parâmetros fixos ou adaptativos.

Os filtros adaptativos operam com algoritmos recursivos, cuja principal vantagem consiste no ajuste automático dos parâmetros do filtro de acordo com as mudanças no sinal de entrada, tornando possível o funcionamento satisfatório destes filtros em ambientes não-estacionários. Dependendo da aplicação, um filtro pode executar tarefas básicas de processamento de informações ([HAYKIN, 2014](#)): a *predição*, que consiste em estimar valores futuros de uma grandeza de interesse com base nas observações disponíveis até o instante corrente; a *filtragem*, que busca estimar o estado atual utilizando medições presentes e passadas; e a *suavização*, na qual se procura reduzir a variabilidade induzida pelo ruído nas estimativas de estados passados por meio do uso adicional de observações futuras. Nesse último caso, admite-se um atraso na produção do resultado, em contrapartida a estimativas geralmente mais precisas do que aquelas obtidas apenas por filtragem.

Algoritmos adaptativos baseados em mínimos quadrados recursivos são amplamente utilizados devido à sua rápida convergência. No entanto, em certas aplicações, em que apresenta variações bruscas dos parâmetros do sinal de entrada a reação do algoritmo pode ser insuficiente para responder rápido a essas variações.

Em ambientes não estacionários o algoritmo deve ser capaz de rastrear continuamente essas variações sem introduzir flutuações excessivas na estimativa, nesse cenário algoritmos adaptativos clássicos apresentam limitações significativas. Um aspecto fundamental é o compromisso intrínseco entre precisão e capacidade de rastreamento do algoritmo. Intuitivamente, esse fenômeno pode ser compreendido por meio de uma analogia simples: o parâmetro que controla a adaptação do algoritmo atua como um "cobertor

curto”, ao melhorar uma característica do algoritmo, inevitavelmente se degrada outra. Esse fenômeno é um desafio que evidencia uma limitação operacional dos algoritmos adaptativos clássicos.

O problema de sinais não estacionários constitui um dos desafios para algoritmos adaptativos para rastrear de forma rápida e estável. Exemplos representativos de sinais não estacionários são eletrocardiograma (ECG) e eletroencefalograma (EEG).

Algoritmos com passo de adaptação reduzido apresentam excelente desempenho em regime estacionário, caracterizado por baixo erro de estimação. No entanto, tais configurações implicam baixa capacidade de rastreamento (*tracking*) em ambientes não estacionários, especialmente diante de variações abruptas dos parâmetros do sistema.

Esse comportamento é análogo ao observado em algoritmos [RLS](#) com fator de esquecimento próximo da unidade ($\lambda \rightarrow 1$), conforme discutido em ([HAYKIN, 2014](#)), nos quais a longa memória temporal reduz a sensibilidade a novas informações. De forma semelhante, no filtro de Kalman, a adoção de covariâncias de processo reduzidas implica menor ganho de Kalman, limitando a capacidade de adaptação a mudanças rápidas e degradando o desempenho de rastreamento.

A literatura propõe diversas estratégias que garantam um comportamento estável do filtro, dentre elas é adicionar um termo referenciado na literatura um “pequeno número positivo” à matriz de covariância ([HAYKIN, 1991](#)). Por outro lado, configurações para melhorar a resposta transitória aumentam a sensibilidade ao ruído aumentando a variância do erro e o risco de instabilidade.

Esse compromisso motivou propostas de extensões de algoritmos e novos algoritmos têm sido em filtragem adaptativa para aplicações diversas. Uma técnica para melhorar a taxa de convergência do algoritmo [RLS](#), é apresentada por Ali ([ALI; REHMAN; ALI, 2010](#)), onde o vetor de ganho leva em consideração o valor de $|1/\varepsilon|$, sendo ε uma estimativa a priori do erro.

Passo variável formulado como VSS-LMS ([MOMOH O. C. UBADIKE, 2020](#)) é comparados com o algoritmo [LMS](#) convencional em termos de MSE e velocidade de convergência. O mecanismo em ([CAI; LAMARE, 2013](#)) é proposto empregar um componente atualizado relacionado à média temporal da correlação de erros para ajustar automaticamente o fator de esquecimento, garantindo convergência rápida e bom rastreamento da interferência do canal.

O trabalho em ([ZHU et al., 2021](#)) considera a estimação de estado com ruído de processo e medição desconhecido, não estacionário em experimentos com dados sintéticos e dados reais de veículos inteligentes. Numa variação desse aspecto o artigo *On Regularization in Adaptive Filtering*, ([BENESTY; PALEOLOGU; CIOCHINĂ, 2011](#)), no contexto do cancelamento do eco, objetiva uma regularização constante considerando a variância do

sinal de entrada, apresentando uma versão componente de correção do algoritmo Mínimos Quadrados Médios Normalizados (*Normalized Least Mean Square NLMS*) regularizada.

Conforme analisado por Slock (SLOCK, 1996), o comportamento de convergência dos algoritmos LMS e NLMS evidencia um compromisso entre rapidez de adaptação e estabilidade, frequentemente interpretado como um *trade-off* entre velocidade de convergência e sensibilidade ao ruído. Esse compromisso motiva o desenvolvimento de estratégias adaptativas capazes de ajustar dinamicamente os parâmetros desses algoritmos em função do grau de não estacionaridade do sinal, resposta rápida versus estabilidade.

É comum na literatura a avaliar desempenhos entre algoritmos segundo algum critério para uma dada aplicação. Filtros bem conhecidos, como o RLS, o LMS e o Filtro de Kalman, utilizam funções de custo de segunda ordem, sendo seus desempenhos ótimos geralmente associados a ambientes com estatísticas Gaussianas. Em destaque, cita-se o algoritmo Recursivo Não Quadrático (RNQ) (SILVA et al., 2015), como uma classe do RLS que ajusta os pesos de um filtro adaptativo.

No artigo “*An algorithm based on non-squared sum of the errors*”, no qual os autores modificam as equações recursivas do RLS ao utilizar como superfície de desempenho uma soma ponderada de potências pares do erro que não se restringem ao quadrado tradicional. O algoritmo RNQ obteve desempenho superior em comparação com algoritmo Mínimos Quadrado Recursivo (RLS), no cenário identificação sistemas, mostrando maior velocidade de adaptação em relação ao RLS padrão.

O filtro de Kalman é conhecido por sua capacidade de rastrear parâmetros variantes no tempo com resposta assintótica adequada, porém tem alto custo computacional e requer sintonização cuidadosa das matrizes de covariância para garantir desempenho robusto no mundo real (HAYKIN, 1991).

Em (SADIKI; SLOCK, 2006) é feita comparação entre o filtro de Kalman do ponto de vista bayesiano e a filtragem adaptativa padrão em termos do excesso de erro médio quadrático (EMSE—Excess Mean Square Error) das estimativas em estado estacionário.

Recentemente, diversas estratégias de regularização para algoritmos adaptativos têm sido investigadas visando melhorar a robustez numérica e o desempenho em ambientes variantes no tempo. Essas abordagens incluem mecanismos de regularização variante no tempo, reutilização de dados e técnicas de penalização, buscando aprimorar a estabilidade, a velocidade de convergência e a capacidade de rastreamento dos algoritmos da família RLS (LI, 2022; SCHMIDT, 2022; OTOPELEANU et al., 2025). Entretanto, observa-se que tais métodos, em geral, introduzem parâmetros adicionais ou estratégias específicas de ajuste para a regularização.

Neste trabalho apresentamos algoritmo autorregularizado, onde o erro instantâneo do próprio algoritmo é usado explicitamente como parâmetro de regularização dinâmica

da sua matriz de auto covariância que conduzem a um melhor desempenho, em termos de velocidade de convergência, quando comparado com o RNQ e filtro de Kalman para identificação de sistemas apresentado em (SÖDERSTRÖM, 1989).

1.3 Motivação

Algoritmos adaptativos são geralmente avaliados segundo critérios principais de desempenho como velocidade de convergência, erro em regime permanente, capacidade de rastreamento, complexidade computacional. Em geral, esses critérios não podem ser simultaneamente otimizados dentro de uma mesma estrutura algorítmica, dando origem ao conhecido compromisso entre desempenho e estabilidade.

Em consulta a literatura constata-se que é comum resultados superiores do Filtro de Kalman sobre o RLS e de seus variantes, principalmente na precisão e convergência.

Recentemente em (SILVA et al., 2015), o algoritmo Recursivo Não Quadrático (RNQ) apresentou melhor convergência do que o RLS numa estrutura de configuração de identificação de sistemas com custo computacional igual do algoritmo RLS. Entretanto nenhuma análise comparativa de desempenho do algoritmo RNQ é considerada em relação a algoritmos clássicos considerados superior, como o algoritmo de filtro de Kalman.

Neste trabalho apresentamos o algoritmo Recursivo Não quadrático Autorregulado (RNQA) com característica de ajuste adaptativo onde o erro instantâneo é o parâmetro de regularização na atualização da sua matriz de autocovariância. Essa estratégias na regularização resultou na melhoria do desempenho da adaptação do algoritmo em regime não estacionário, na capacidade de rastreamento (*tracking ability*) que conduzem a um melhor desempenho quando comparado com outros filtros.

1.4 Objetivos

1.4.1 Objetivo Geral

Propor autorregularização da matriz de autocovariância no algoritmo RNQA, com o objetivo de acelerar a convergência na estimação de parâmetros e aprimorar a capacidade de rastreamento em ambiente não estacionário. Tal abordagem visa mitigar o compromisso entre velocidade de convergência e capacidade de rastreamento, especialmente em cenários não estacionários.

1.4.2 Objetivos Específicos

- Propor um mecanismo de autorregularização adaptativa na matriz de autocovariância do algoritmo [RNQA](#), baseado no erro instantâneo, em substituição à regularização clássica de valor constante.
- Analisar a influência da autorregularização adaptativa na dinâmica de aprendizagem do algoritmo, especialmente sobre a velocidade de convergência e a capacidade de rastreamento.
- Comparar o desempenho do algoritmo [RNQA](#) com o algoritmo [RNQ](#) convencional e com o filtro de Kalman, considerando métricas como velocidade de convergência em uma configuração de identificação de sistemas.

2 Filtragem adaptativa

2.1 Introdução

Historicamente, a abordagem paramétrica de sinais tem sido explorada com sucesso em problemas de comunicações, controle, robótica, radar, sismologia e engenharia biomédica. Os chamados problemas de filtragem podem ser identificados e caracterizados mais especificadamente pelos termos de filtragem, suavização, predição (HAYKIN, 2014).

O principal objetivo da filtragem de sinais é melhorar a qualidade do sinal de acordo com um critério de desempenho. Os sinais podem ser considerados no domínio do tempo ou no domínio da frequência.

Muitos algoritmos adaptativos se utilizam do erro quadrado médio (MSE - *mean squared error*) como função de custo que se deseja minimizar. O erro quadrático médio define uma função convexa no espaço dos parâmetros, cuja superfície de custo assume a forma de um paraboloide quadrático estritamente convexo, assegurando a existência de um único mínimo global. O ponto mínimo desta superfície representa a solução de Wiener conforme (FARHANG-BOROUJENY, 2013) representada na Figura (2). O problema consiste em determinar procedimentos de tal forma a encontrar esse mínimo, de maneira rápida e com menor erro final.

A escolha de um algoritmo em detrimento de outro é determinada por um ou mais dos seguintes fatores: taxa de convergência (*Rate of convergence*), desajuste (*Misadjustment*) Rastreamento (*Tracking*) (HAYKIN, 2014). A taxa de convergência é definido como o número de ciclos de adaptação necessários para que o algoritmo, em resposta a entradas estacionárias, convirja “suficientemente próximo” da solução Wiener ótima no sentido do erro quadrático médio. Uma rápida taxa de convergência permite que o algoritmo se adapte rapidamente a um ambiente estacionário de estatísticas desconhecidas. O desajuste é um parâmetro que fornece uma medida quantitativa do valor pelo qual o valor final do erro quadrático médio, calculado em média sobre um conjunto de filtros adaptativos, se desvia da solução de Wiener. O rastreamento refere-se à habilidade do algoritmo em acompanhar variações estatísticas quando opera em ambiente não estacionário. O desempenho do algoritmo, entretanto, é influenciado pelo compromisso entre a taxa de convergência e a flutuação em estado estacionário. Em particular, um algoritmo adaptativo com boa propriedade de convergência não necessariamente possui uma rápida capacidade de rastreamento, e vice-versa (HAYKIN, 1991)

Para que um filtro adaptativo seja robusto, pequenas perturbações (ou seja, perturbações com pequena energia) só podem resultar em pequenos erros de estimativa. As

perturbações podem surgir de uma variedade de fatores, internos ou externos ao filtro. O requisito computacional é o número de operações (ou seja, multiplicações, divisões e adições/subtrações) necessárias para fazer um ciclo completo de adaptação do algoritmo.

Os filtros são classificados como lineares ou não lineares. Um filtro é linear se a quantidade filtrada, suavizada ou prevista na saída do filtro, for uma função linear das observações aplicadas à entrada do filtro. Caso contrário, o filtro é não linear.

Na abordagem estatística para a solução do problema de filtragem linear de modo a minimizar os efeitos do ruído na saída do filtro de acordo com algum critério estatístico para minimizar o valor quadrático médio do sinal de erro definido como a diferença entre alguma resposta desejada e a saída real do filtro (HAYKIN, 2014).

Uma importante medida de desempenho para um filtro adaptativo é seu erro quadrático médio (mean square error-MSE) em estado estacionário, definido como (2.1), (YOUSEF; SAYED, 2001):

$$J_{\infty} \triangleq \lim_{n \rightarrow \infty} \mathbb{E}\{|e(n)|^2\} \quad (2.1)$$

onde $e(n) = d(n) - y(n)$ representa o erro de estimativas instantâneo entre o sinal desejado $d(n)$ e a saída do filtro $y(n)$, e $\mathbb{E}\{\cdot\}$ denota o operador esperança.

2.2 O Combinador linear adaptativo

A estrutura mais usada na implementação de filtros adaptativos é o combinador linear adaptativo, Figura 1, observa-se que o filtro adaptativo de ordem L com uma entrada única $\mathbf{u}_i$ (no tempo i), definida como:

$$\mathbf{u}_i = [u_i, u_{i-1}, \dots, u_{i-L+1}]^T \quad (2.2)$$

O vetor peso $\mathbf{w}_n$, dado por

$$\mathbf{w}_n = [w_{n0}, w_{n1}, \dots, w_{n(L-1)}]^T \quad (2.3)$$

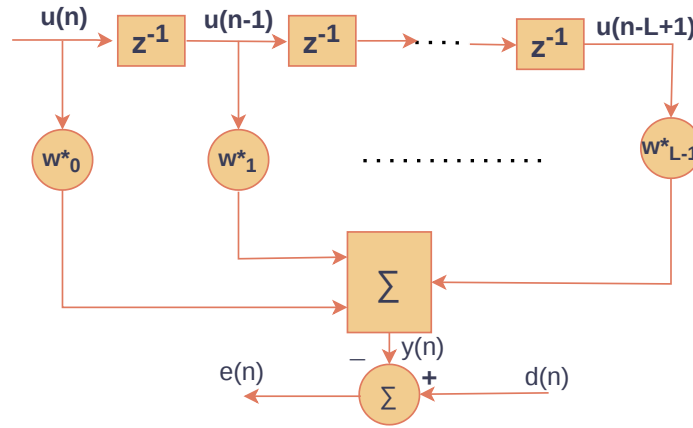
e a saída y_n , é dada pelo produto interno de $\mathbf{u}_i$ por $\mathbf{w}_i$

$$y_i = \mathbf{u}_i^T \mathbf{w}_{n-1} = \mathbf{w}_{n-1}^T \mathbf{u}_i \quad (2.4)$$

Conforme a Figura 1, o sinal de erro no instante i é dado por:

$$e_i = d_i - y_i \quad (2.5)$$

Figura 1 – Estrutura transversal do combinador linear adaptativo.



Fonte: Elaborado pelo autor.

Substituindo (2.4) nesta expressão, temos:

$$e_i = d_i - \mathbf{u}_i^T \mathbf{w}_{n-1} = d_i - \mathbf{w}_{n-1}^T \mathbf{u}_i \quad (2.6)$$

Existem vários algoritmos e abordagens que podem ser utilizados em filtragem adaptativa, dependendo dos requisitos do problema. No entanto, existem duas abordagens principais para o desenvolvimento de algoritmo de filtros adaptativos, discutidos nas próximas seções.

2.3 Algoritmos de gradiente estocástico

O MSE é uma função convexa dos componentes do vetor peso e gera uma superfície paraboloide que garante a existência de um mínimo global, o que representa uma solução de Wiener (WIENER, 1949). Essa solução pode ser encontrada pelo método de descida mais íngreme, que utiliza o vetor gradiente para descer passo a passo para um mínimo da função de erro. Para determinar a solução ótima a partir do critério do erro quadrático médio, conforme apresentado em (WIDROW; STEARNS, 1985),

$$\xi = E[e_i^2] = E(d_i - \mathbf{w}_n^T \mathbf{u}_i)^2 \quad (2.7)$$

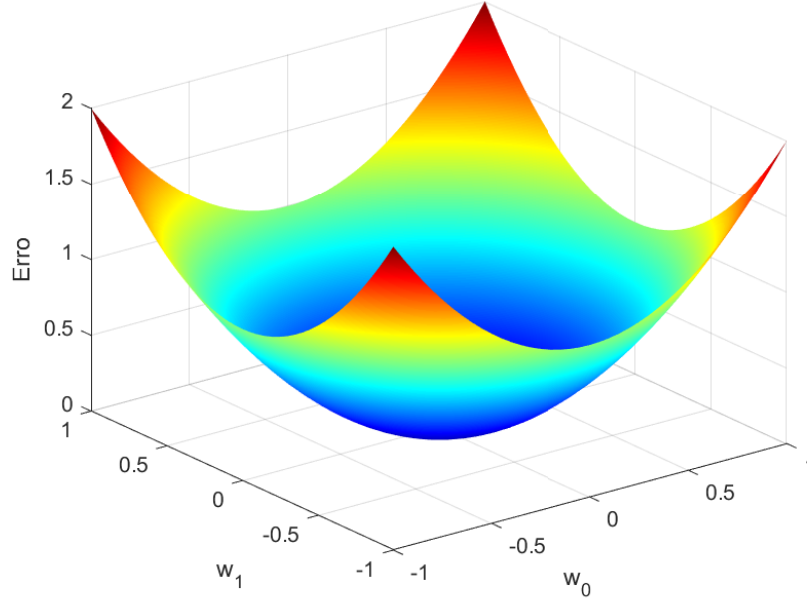
considerando um ambiente estacionário, podemos desenvolver a eq. (2.7) como

$$\xi = d_i - \mathbf{w}^T \boldsymbol{\varphi} \mathbf{w}_n^T - 2\mathbf{z}^T \mathbf{w} \quad (2.8)$$

sendo $\boldsymbol{\varphi}$ a matriz de autocorrelação do sinal de entrada $\mathbf{u}_i$ e $\mathbf{z}$ o vetor de correlação cruzada do sinal de entrada $\mathbf{u}_i$ e o sinal desejado d_i .

Pode-se observar que o MSE é uma função quadrática dos pesos, cujo gráfico é uma superfície cônica parabolóide, da forma representada na Figura, (2), onde são considerados apenas dois pesos w_0 , w_1 . Esta função, obviamente, nunca pode ser negativa.

Figura 2 – Uma superfície quadrática tridimensional. O erro quadrático está plotado na vertical, w_0 e w_1 no intervalo -1 a 1



Fonte: Elaborado pelo autor.

O gradiente da superfície de desempenho do erro quadrado médio, designado por ∇ , pode ser obtido derivando a eq. (2.8) para obter o erro quadrado médio mínimo:

Para desenvolver o algoritmo [LMS](#), é usado o próprio erro e_i^2 como estimativa de ξ_i . Para cada interação no processo iterativo é obtido uma estimativa do gradiente na forma :

$$\hat{\nabla}_n = \begin{bmatrix} \frac{\partial e_n^2}{\partial w_0} \\ \vdots \\ \frac{\partial e_n^2}{\partial w_L} \end{bmatrix} = 2\varphi \mathbf{w} - 2\mathbf{z}_n \quad (2.9)$$

O vetor peso é ajustado para seu valor ótimo, $\mathbf{w}_*$, onde o gradiente é zero:

$$\mathbf{w}_* = \varphi^{-1} \mathbf{z} \quad (2.10)$$

Esta equação é a forma matricial da equação Wiener-Hopf.

Um sistema modificado da equação Wiener-Hopf, usado para adaptar os pesos do filtro em direção ao mínimo é o algoritmo chamado de mínimo quadrático médio ([LMS](#)-Least Mean Squares). Este algoritmo calcula o gradiente da função de erro de valores

instantâneos da matriz de correlação das entradas e do vetor de correlação cruzada entre as entradas e a resposta desejada.

As derivadas de e_i , em relação aos pesos, seguem diretamente da eq. (2.5). A partir da eq.(2.9), temos o algoritmo **LMS** (**WIDROW; HOFF, 1960**).

$$\mathbf{w}_n = \mathbf{w}_{n-1} + \mu e_n \mathbf{u}_n \quad (2.11)$$

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

2.4 Algoritmos de mínimos quadrados

No método dos mínimos quadrados os pesos do algoritmo adaptativos são calculados, nos instantes $n > 0$, tal que a quantidade

$$\zeta = \sum_{i=1}^n \rho_i |e_i|^2, \quad (2.12)$$

é minimizada, sendo ρ_i um fator de ponderação, daí o nome de mínimos quadrados.

Um algoritmo muito popular da família dos mínimos quadrados recursivo (**Recursive Least Square-RLS**). Tal algoritmo tem vantagem de baixa sensibilidade à natureza do sinal de entrada e uma maior velocidade de convergência quando comparado com os algoritmos de gradiente estocástico.

2.5 Algoritmo de mínimos quadrados recursivo

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

2.5.1 Dedução do algoritmo RLS

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

$$\rho_i = \lambda^{n-1} \quad (2.13)$$

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

O método dos mínimos quadrados padrão visto anteriormente corresponde ao caso em que $\lambda = 1$. O parâmetro λ é denominado fator de esquecimento. Quando $\lambda < 1$, os

fatores de ponderação definido em (2.13) dão um peso maior às amostras mais recentes das estimativas do erro (e assim, as amostras recentes do dado observado) comparado com as amostras mais antigas.

Substituindo (2.13) em (2.12) obtém-se a equação de custo a ser minimizada na dedução do RLS:

$$\varepsilon_n = \sum_{i=1}^n \lambda^{n-i} |e_i|^2, \quad (2.14)$$

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 na forma matricial

$$\mathbf{w}_* = \varphi^{-1} \mathbf{z} \quad (2.15)$$

sendo

$$\varphi_n = \sum_{i=1}^n \lambda^{n-1} \mathbf{u}_i \mathbf{u}_i^T, \quad (2.16)$$

a matriz de autocorrelação do sinal de entrada $\mathbf{u}_i$ e

$$\mathbf{z}_n = \sum_{i=1}^n \lambda^{n-1} \mathbf{u}_i d_i \quad (2.17)$$

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

Expandindo a eq. (2.16) e isolando o termo correspondente a $i = n$, temos:

$$\varphi_n = \lambda \left[\sum_{i=1}^n \lambda^{n-1} \mathbf{u}_i \mathbf{u}_i^T \right] + \mathbf{u}_n \mathbf{u}_n^T \quad (2.18)$$

$$\varphi_n = \lambda [\varphi_{n-1}] + \mathbf{u}_n \mathbf{u}_n^T \quad (2.19)$$

e da mesma forma em (2.17),

$$\mathbf{z}_n = \lambda [\mathbf{z}_{n-1}] + d_n \mathbf{u}_n \quad (2.20)$$

2.5.2 A Regularização

A estimação por mínimos quadrados, no contexto de algoritmos adaptativos, pode ser interpretada como um problema inverso no qual se busca estimar o vetor de parâmetros desconhecido a partir das observações do vetor de entrada $\mathbf{u}_n$ e da resposta desejada d_n . O desempenho desses métodos está diretamente relacionado às propriedades estatísticas do sinal de entrada, em particular à matriz de autocorrelação $\mathbf{R} = E[\mathbf{u}_n \mathbf{u}_n^T]$ (HAYKIN, 2014; LJUNG, 1999).

O mal condicionamento desse problema está associado à elevada correlação entre os componentes do vetor de entrada, frequentemente decorrente da autocorrelação do sinal de

excitação, o que implica redundância de informação nas observações. Como consequência, a matriz $\mathbf{R}$ apresenta autovalores altamente dispersos, caracterizados por um elevado número de condição (GOLUB; LOAN, 2013; BJÖRCK, 1996).

No contexto de algoritmos adaptativos, como o LMS e o RLS, essa dispersão dos autovalores impacta diretamente a velocidade de convergência. Em particular, no algoritmo LMS, a taxa de convergência é limitada pelo espalhamento dos autovalores de $\mathbf{R}$, de modo que direções associadas a autovalores maiores convergem mais rapidamente, enquanto direções associadas a autovalores pequenos apresentam convergência significativamente mais lenta (HAYKIN, 2014). Esse fenômeno resulta em convergência não uniforme entre os modos associados aos autovalores, aumentando o tempo necessário para atingir o regime permanente.

No caso do algoritmo RLS, embora a convergência seja, em geral, mais rápida e menos sensível ao espalhamento espectral, o mal condicionamento ainda pode afetar a estabilidade numérica e a robustez da estimação, especialmente em presença de ruído e limitações de precisão computacional (LJUNG, 1999; HAYKIN, 2014).

Dessa forma, a elevada correlação entre os regressores — frequentemente induzida pela autocorrelação do sinal de entrada constitui um fator crítico que degrada tanto a estabilidade numérica quanto a velocidade de convergência dos algoritmos adaptativos, justificando o uso de técnicas de regularização e estratégias de pré-condicionamento (TIKHONOV; ARSENIN, 1977).

A natureza mal condicionada da estimação por mínimos quadrados se deve aos seguintes motivos:

- Há informações insuficientes nos dados de entrada para reconstruir o mapeamento entrada-saída de forma única.
- A presença inevitável de ruído ou imprecisão nos dados de entrada adiciona incerteza ao mapeamento entrada-saída reconstruído.

Para tornar o problema de estimação “bem condicionado”, alguma forma de informação prévia sobre o mapeamento entrada-saída é necessária. Isso, por sua vez, significa que a formulação da função de custo deve ser expandida para levar em consideração as informações prévias.

Para atingir esse objetivo, é expandida a função de custo a ser minimizada como a soma de dois componentes,

$$\varepsilon_n = \sum_{i=1}^n \lambda^{n-i} |e_i|^2 + \delta \lambda^n \|\mathbf{w}_n\|^2 \quad (2.21)$$

1. A soma dos quadrados dos erros ponderados,

$$\sum_{i=1}^n \lambda^{n-i} |e_i|^2 = \sum_{i=1}^n \lambda^{n-i} |d_i - \mathbf{w}_n^T \mathbf{u}_i|^2$$

que depende dos dados. Este componente mede o erro ponderado exponencialmente entre a resposta desejada d_i e a resposta real do filtro, y_i , que está relacionada ao vetor de entrada $\mathbf{u}_i$ por $e_i = d_i - \mathbf{w}_n^T \mathbf{u}_i$

2. O termo de regularização,

$$\delta \lambda^n \|\mathbf{w}_n\|^2 = \delta \lambda^n \mathbf{w}_n^T \mathbf{w}_n$$

onde δ é um número real positivo chamado parâmetro de regularização. Exceto pelo fator $\delta \lambda^n$, o termo de regularização depende exclusivamente do vetor de pesos $\mathbf{w}(\mathbf{n})$. O termo de regularização é incluído na função de custo para estabilizar a solução do problema de mínimos quadrados recursivo, suavizando a solução.

Reformulação das Equações Normais

Expandindo a equação (2.21) e agrupando os termos, mostra que o efeito da inclusão do termo de regularização $\delta \lambda^n \|\mathbf{w}_n\|^2$ na função de custo φ_n é equivalente a uma reformulação da matriz de correlação média temporal $L \times L$ do vetor de entrada $\mathbf{u}_n$, (HAYKIN, 1991):

$$\varphi_n = \sum_{i=1}^n \lambda^{n-1} \mathbf{u}_i \mathbf{u}_i^T + \delta \lambda^n \mathbf{I}, \quad (2.22)$$

onde $\mathbf{I}$ é a matriz identidade $L \times L$. A adição do termo de regularização também tem o efeito de tornar a matriz de correlação φ_n não singular em todos os estágios do cálculo, a partir de $n = 0$.

O vetor $\mathbf{z}_n$, $L \times 1$, de correlação cruzada média temporal entre as entradas do filtro de resposta ao impulso finita (FIR, do inglês *Finite Impulse Response*) e a resposta desejada não é afetado pelo uso da regularização, conforme a eq.(2.20).

Cálculos recursivos de φ_n e $\mathbf{z}_n$

Isolando o termo correspondente a $i = n$ do restante da soma no lado direito da eq.(2.22), podemos escrever,

$$\varphi_n = \lambda \left[\sum_{i=1}^{n-1} \lambda^{n-1-i} \mathbf{u}_i \mathbf{u}_i^T + \delta \lambda^{n-1} \mathbf{I} \right] + \mathbf{u}_n \mathbf{u}_n^T \quad (2.23)$$

Por definição, a expressão dentro dos colchetes no lado direito da eq.(2.23) é igual à matriz de correlação φ_{n-1} . Portanto, temos a seguinte recursão para atualizar o valor da matriz de correlação da entrada, da forma da eq.(2.24),

$$\varphi_n = \lambda [\varphi_{n-1}] + \mathbf{u}_n \mathbf{u}_n^T \quad (2.24)$$

sendo φ_{n-1} é o valor anterior da matriz de correlação, e o produto matricial $\mathbf{u}_n \mathbf{u}_n^T$ desempenha o papel de um termo de “correção” na operação de atualização. Observe que a recursão da eq.(2.24) é válida, independentemente da condição de inicialização.

Da mesma forma, podemos usar a eq.(2.20) recursiva do vetor de correlação cruzada entre a entrada e a resposta desejada:

$$\mathbf{z}_n = \lambda [\mathbf{z}_{n-1}] + d_n \mathbf{u}_n \quad (2.25)$$

Lemma de inversão de matrizes

Lema 1 - Sejam A e B matrizes definidas positivas de dimensão $L \times L$, definimos

$$A = B^{-1} + CD^{-1}C^T \quad (2.26)$$

Da eq.(2.26) obtemos :

$$A^{-1} = B - BC(D + C^T BC)^{-1}C^T B \quad (2.27)$$

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

A demonstração desse Lema é estabelecida pela multiplicação da eq. (2.26) por (2.27). Na próxima seção mostraremos como o lema da inversão de matrizes pode ser aplicada para obter uma equação recursiva.

2.5.3 Algoritmo RLS ponderado exponencialmente

Considerando que a matriz de correlação positiva definida positiva e não singular, aplica-se o lema de inversão de matrizes para equação recursiva (2.24), onde são feitas as seguintes atribuições:

$$\begin{aligned} \mathbf{A} &= \varphi_n, \\ \mathbf{B}^{-1} &= \lambda \varphi_{n-1} \Rightarrow \mathbf{B} = \lambda^{-1} \varphi_{n-1}^{-1}, \\ \mathbf{C} &= \mathbf{u}_n, \\ \mathbf{D} &= \mathbf{I}. \end{aligned} \quad (2.28)$$

Substituindo (2.28) na eq.(2.27) do lema de inversão das matrizes, obtemos a relação de recursividade da matriz φ_n ,

$$\boldsymbol{\varphi}_n^{-1} = \lambda^{-1} \boldsymbol{\varphi}_{n-1}^{-1} - \frac{\lambda^{-1} \boldsymbol{\varphi}_{n-1}^{-1} \mathbf{u}_n \mathbf{u}_n^\top \boldsymbol{\varphi}_{n-1}^{-1}}{1 + \lambda^{-1} \mathbf{u}_n^\top \boldsymbol{\varphi}_{n-1}^{-1} \mathbf{u}_n} \quad (2.29)$$

Por conveniência computacional, escreve-se as seguintes igualdades,

$$\mathbf{P}_n = \boldsymbol{\varphi}_{n-1}^{-1} \quad (2.30)$$

e

$$\mathbf{k}_n = \frac{\lambda^{-1} \mathbf{P}_{n-1} \mathbf{u}_n}{1 + \lambda^{-1} \mathbf{u}_n^\top \mathbf{P}_{n-1} \mathbf{u}_n} \quad (2.31)$$

Assim podemos escrever a eq.(2.29) como

$$\mathbf{P}_n = \lambda^{-1} \mathbf{P}_{n-1} - \lambda^{-1} \mathbf{k}_n \mathbf{u}_n^\top \mathbf{P}_{n-1} \quad (2.32)$$

sendo $\mathbf{P}_n$ a inversa da matriz de autocorrelação de dimensão $L \times L$ e $\mathbf{k}_n$ o vetor ganho de dimensão $L \times 1$. Reorganizando a eq.(2.31), temos,

$$\begin{aligned} \mathbf{k}_n + \mathbf{k}_n \lambda^{-1} \mathbf{u}_n^\top \mathbf{P}_{n-1} \mathbf{u}_n &= \lambda^{-1} \mathbf{P}_{n-1} \mathbf{u}_n \\ \mathbf{k}_n &= \left[\lambda^{-1} \mathbf{P}_{n-1} - \mathbf{k}_n \lambda^{-1} \mathbf{u}_n^\top \mathbf{P}_{n-1} \right] \mathbf{u}_n \end{aligned} \quad (2.33)$$

Substituindo eq. (2.32) na eq.(2.33), segue,

$$\mathbf{k}_n = \mathbf{P}_n \mathbf{u}_n \quad (2.34)$$

ou

$$\mathbf{k}_n = \boldsymbol{\varphi}_n^{-1} \mathbf{u}_n \quad (2.35)$$

2.5.4 Atualização do vetor peso

No desenvolvimento de uma equação recursiva, utilizaremos as eq.(2.15), (2.17) e (2.30) para expressar a estimativa do mínimo quadrado do vetor peso, $\hat{\mathbf{w}}_n$, no instante n como segue

$$\begin{aligned} \hat{\mathbf{w}}_n &= \mathbf{P}_n \left[\lambda \mathbf{z}_{n-1} + \mathbf{u}_n d_n \right] \\ \hat{\mathbf{w}}_n &= \lambda \mathbf{P}_n \mathbf{z}_{n-1} + \mathbf{P}_n \mathbf{u}_n d_n \end{aligned} \quad (2.36)$$

Substituindo a eq.(2.32) na eq.(2.37) obtemos

$$\hat{\mathbf{w}}_n = \mathbf{P}_{n-1} \mathbf{z}_{n-1} - \mathbf{k}_n \mathbf{u}_n^\top \mathbf{P}_{n-1} \mathbf{z}_{n-1} + \mathbf{P}_n \mathbf{u}_n d_n \quad (2.37)$$

Da eq.2,29 e 2.31 segue-se

$$\hat{\mathbf{w}}_n = \hat{\mathbf{w}}_{n-1} + \mathbf{k}_n \epsilon_n \quad (2.38)$$

sendo ϵ_n a estimativa do erro a priori definida por,

$$\epsilon_n = d_n - \mathbf{u}_n^T \hat{\mathbf{w}}_n \quad (2.39)$$

2.5.5 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

$$d_n = \mathbf{w}_*^T \mathbf{u}_n + e_n \quad (2.40)$$

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

O comportamento médio do vetor peso no algoritmo RLS

De [\(2.15\)](#) e [\(2.17\)](#)

$$\hat{\mathbf{w}}_n = \boldsymbol{\varphi}_n^{-1} \sum_{i=1}^n \lambda^{n-i} \mathbf{u}_i d_i \quad (2.41)$$

substituindo [\(2.40\)](#) em [\(2.41\)](#) e usando [\(2.16\)](#), temos

$$\hat{\mathbf{w}}_n = \hat{\mathbf{w}}_* + \boldsymbol{\varphi}_n^{-1} \sum_{i=1}^n \lambda^{n-i} \mathbf{u}_i d_i \quad (2.42)$$

Aplicando o operador esperança em ambos os lados da eq. [\(2.42\)](#) e reconhecendo do princípio de ortogonalidade que todos os elementos do vetor $\mathbf{u}_n$ são ortogonais ao erro e_n , obtemos

$$E[\hat{\mathbf{w}}_n] = \hat{\mathbf{w}}_* \quad (2.43)$$

Matriz de correlação do vetor desvio

Definimos vetor desvio

$$\mathbf{v}_n = \hat{\mathbf{w}}_n - \mathbf{w}_* \quad (2.44)$$

Da eq. [\(2.42\)](#) temos

$$\mathbf{v}_n = \boldsymbol{\varphi}_n^{-1} \sum_{i=1}^n \lambda^{n-i} \mathbf{u}_i d_i \quad (2.45)$$

Definimos a matriz de correlação do vetor desvio da seguinte forma

$$\mathbf{K}_n = E[\mathbf{v}_n \mathbf{v}_n^T] \quad (2.46)$$

Substituindo (2.45) em (2.46) e sendo $[\boldsymbol{\varphi}_n^{-1}]^T = \boldsymbol{\varphi}_n^{-1}$ e $[\lambda^{-1}]^T = \lambda^{-1}$ obtemos

$$\mathbf{K}_n = E\left[\left(\boldsymbol{\varphi}^{-1} \sum_{i=1}^n \lambda^{n-i} \mathbf{u}_i e_i\right) \left(\sum_{i=1}^n \lambda^{n-i} \mathbf{u}_i e_i\right) \boldsymbol{\varphi}^{-1}\right] \quad (2.47)$$

Pra rigoroso cálculo da eq.(2.47) devemos fazer as seguintes hipóteses, (FARHANG-BOROUJENY, 1999):

- O vetor de entrada $\mathbf{u}_i$ possui amostras de um processo ergótico. Assim podemos usar as médias do tempo ao invés do conjunto de médias.
- O fator de esquecimento λ é muito próximo de 1.
- O tempo n na qual $\mathbf{K}_n$ é calculado é muito grande.

Notamos que eq.(2.15) que φ é uma soma ponderada dos produtos internos.

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

Assim, considerando as hipóteses acima, encontramos

$$\boldsymbol{\varphi}_n \approx \frac{1 - \lambda^n}{1 - \lambda} \mathbf{R} \quad (2.48)$$

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

Substituindo eq.(2.48) na eq.(2.47) e sendo $E[e_n e_n^T] = \sigma^2$, obtemos

$$\begin{aligned} \mathbf{K}_n &= \sigma^2 \left(\frac{1 - \lambda^n}{1 - \lambda} \right)^2 \left(\frac{1 + \lambda^{2n}}{1 - \lambda^2} \right) \mathbf{R}^{-1} \\ &= \sigma^2 \left(\frac{1 - \lambda}{1 + \lambda} \right) \left(\frac{1 + \lambda^n}{1 - \lambda^n} \right) \mathbf{R}^{-1} \end{aligned} \quad (2.49)$$

Curva de aprendizagem do algoritmo RLS

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

$$\xi_n = \mathbb{E}[\epsilon_n^2] \quad (2.50)$$

Da eq.(2.44) e eq.(2.40) podemos escrever o erro *a priori* da seguinte forma

$$\epsilon_n = e_n - \mathbf{v}_{n-1}^T \mathbf{u}_n \quad (2.51)$$

Substituindo eq.(2.51) na eq.(2.50)

$$\xi_n = \mathbb{E}[e_n^2] + \mathbb{E}[\mathbf{u}_n \mathbf{v}_{n-1}^T \mathbf{v}_{n-1}^T \mathbf{u}_n] - \mathbb{E}[\mathbf{v}_{n-1}^T \mathbf{u}_n e_n] - \mathbb{E}[e_n \mathbf{u}_n^T \mathbf{v}_{n-1}] \quad (2.52)$$

O primeiro valor esperado do lado direito da eq.(2.52) é 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 no sentido amplo 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 eq.(2.52), como segue,

$$\begin{aligned} E[\mathbf{u}_n \mathbf{v}_{n-1}^T \mathbf{v}_{n-1}^T \mathbf{u}_n] &= E[\text{Tr}\{\mathbf{u}_n \mathbf{v}_{n-1}^T \mathbf{v}_{n-1}^T \mathbf{u}_n\}] \\ &= E[\text{Tr}\{\mathbf{u}_n^T \mathbf{u}_n \mathbf{v}_{n-1} \mathbf{v}_{n-1}^T\}] \\ &= \text{Tr}\{E[\mathbf{u}_n^T \mathbf{u}_n] E[\mathbf{v}_{n-1} \mathbf{v}_{n-1}^T]\} \\ &= \text{Tr}\{\mathbf{R} \mathbf{K}_{n-1}\} \end{aligned} \quad (2.53)$$

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 de desvio $\mathbf{K}_n = E[\mathbf{v}_n \mathbf{v}_n^T]$.

2. O vetor de medição e_n depende do vetor de entrada $\mathbf{u}_n$. Isto segue de uma simples manipulação da eq.(2.40). 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 da eq.(2.52) é zero. Assim,

$$E[\mathbf{v}_{n-1}^T \mathbf{u}_n e_n] = E[\mathbf{v}_{n-1}^T] E[\mathbf{u}_n e_n] \quad (2.54)$$

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

$$E[\mathbf{v}_{n-1}^T \mathbf{u}_n e_n] = 0 \quad (2.55)$$

3. O quarto valor esperado do lado direito da eq.(2.52) tem a mesma forma matemática considerada no item 2. Entretanto podemos dizer que este valor esperado é igual a zero,

$$E[e_n \mathbf{v}_{n-1}^T \mathbf{u}_n] = 0 \quad (2.56)$$

Assim, reconhecendo que $E[e_n^2] = \xi_{min}$, e usando os resultados das eq.(2.53) ate eq.(2.55) na eq.(2.52), obtemos a seguinte expressão para o erro quadrático médio no algoritmo RLS:

$$\xi_n = \xi_{\min} + \text{Tr}(\mathbf{R}\mathbf{K}_{n-1}) \quad (2.57)$$

sendo $\xi_{\min}$ o mínimo MSE do filtro quando uma estimativa perfeita de $\mathbf{w}_*$ é calculada. Substituindo eq.(2.49) na eq.(2.57) obtemos:

$$\xi_n = \xi_{\min} + \left(\frac{1-\lambda}{1+\lambda}\right)\left(\frac{1+\lambda^{n-1}}{1-\lambda^{n-1}}\right)\mathbf{R}^{-1}L\xi_{\min} \quad (2.58)$$

Este resultado descreve a curva de aprendizagem do algoritmo [RLS](#).

Tempo de aprendizagem do algoritmo RLS

O segundo termo do lado direito da eq.(2.58) é um valor positivo que indica o desvio de ξ_n de $\xi_{\min}$. Notamos que a velocidade que os termos convergem é determinado pelo termo exponencial λ^{n-1} , ou equivalentemente λ^n . Desta forma definimos o tempo de aprendizagem τ_{rls} associado com o algoritmo [RLS](#) usando a seguinte equação:

$$\lambda^n = e^{\frac{-n}{\tau_{rls}}} \quad (2.59)$$

Resolvendo a eq.(2.59) obtemos,

$$\tau_{rls} = \frac{-1}{\ln(\lambda)} \quad (2.60)$$

Para simplificar (2.60) usaremos a seguinte aproximação,

$$\ln(\lambda) = \ln(1 - (1 - \lambda)) \approx -(1 - \lambda). \quad (2.61)$$

Substituindo (2.61) em (2.60) temos:

$$\tau_{rls} \approx \frac{1}{(1 - \lambda)} \quad (2.62)$$

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

Excesso do erro quadrático médio e desajuste

Sabendo que quadrático médio (ξ_n) é maior que o erro quadrático médio mínimo ($\xi_{\min}$) temos, então, um excesso no erro erro final e podemos definir o excesso do erro quadrático médio ($EMSE$), como a diferença entre o erro quadrático médio atual e o erro quadrático médio mínimo ([FARHANG-BOROUJENY, 1999](#)).

Usando (2.58) obtemos:

$$EMSE = \left(\frac{1-\lambda}{1+\lambda}\right)L\xi_{\min}. \quad (2.63)$$

Definimos também o *ExcessoMSE* normalizado pelo erro médio mínimo, como o desajuste M_{RLS} :

$$M_{RLS} = \frac{EMSE}{\xi_{min}}. \quad (2.64)$$

Substituindo eq.(2.63) na eq.(2.64) temos

$$M_{RLS} = \left(\frac{1 - \lambda}{1 + \lambda} \right) L \quad (2.65)$$

sendo L a ordem do filtro.

2.5.6 Conclusão

Neste capítulo, realizamos uma revisão da superfície quadrática gerada quando se utiliza o erro quadrático médio como critério aplicado sobre o erro em filtro adaptativo. Mostramos a dedução dedução do algoritmo [RLS](#) e descrevemos as equações que determinam sua condição de convergência. O tempo de aprendizagem e o desajuste também foram enfatizados pois os mesmos são utilizados como referência comparativa de outros algoritmos adaptativos.

3 Algoritmo Baseado em uma única potência par do erro (RNQ)

3.1 Introdução

Neste capítulo é apresentado o desenvolvimento um algoritmo adaptativo baseado em uma potencia par do erro, inspirado no algoritmo [RLS](#) padrão, denominado de recursivo não quadrático [RNQ](#), a qual é escolhido uma função baseada em uma única potência par do erro como critério a se minimizado. O objetivo é determinado por um algoritmo que ajusta os pesos de um combinador linear adaptativo de forma tal que minimiza a função custo. É mostrado, também, que a superfície de desempenho obtida por esse critério oferece maior convergência, bem como um menor desajuste na busca do peso ótimo quando comparado com algoritmo .

3.2 Dedução do algoritmo

A estrutura básica do filtro adaptativo é composta do sinal desejado d_i , um vetor de entrada $\mathbf{u}_i = [u_i, u_{i-1}, \dots, u_{i-L+1}]$ e um vetor erro usado para atualizar o vetor peso $\mathbf{w}_n = [w_{0,n}, w_{1,n}, \dots, w_{L-1,n}]^T$ do filtro com objetivo de alcançar uma aproximação de d_i pela estimação do sinal de saída $y_i = \mathbf{w}_{n-1}^T \mathbf{u}_i$ após calcular o erro, $e_i = d_i - \mathbf{w}_{n-1}^T \mathbf{u}_i$, sendo $0 \leq i \leq n$ representa a amostra atual e L é 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 :

$$J_n = \sum_{j=1}^m \{ \lambda^{n-1} [e_i]^{2j} \}, \quad (3.1)$$

sendo j e n números positivos sendo $j > 1$ e $0 \ll \lambda < 1$, fator de 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 eq.(3.1) em relação a $\mathbf{w}$, obtemos,

$$\begin{aligned} \nabla J_n &= - \sum_{i=1}^n \{ \lambda^{n-i} \mathbf{u}_i [e_i]^{2j-1} \} \\ &= - \sum_{i=1}^n \{ \lambda^{n-i} \mathbf{u}_i 2j [d_i - \mathbf{w}_n^T \mathbf{u}_i]^{2j-1} \} \end{aligned} \quad (3.2)$$

Desenvolvendo o binômio $[d_i - \mathbf{w}_n^T \mathbf{u}_i]^{2j-1}$, e negligenciando as potências de alta ordem, podemos reescrever a eq.(3.2) como,

$$\begin{aligned} \nabla J_n &\approx - \sum_{i=1}^n \lambda^{n-i} \mathbf{u}_i 2j \{ d_i^{2j-1} - (2j-1) d_i^{2j-2} (\mathbf{w}_n^T \mathbf{u}_i) \} \\ &\approx -2j \left[\sum_{i=1}^n [\lambda^{n-i} d_i^{2j-1} \mathbf{u}_i] - (2j-1) \sum_{i=1}^n [\lambda^{n-i} d_i^{2j-1} \mathbf{u}_i \mathbf{u}_i^T] \mathbf{w}_n \right], \end{aligned} \quad (3.3)$$

sendo o vetor de correlação cruzada, $\mathbf{z}_n$, entre as entradas e as resposta desejada de dimensão $L \times 1$, definido pelo primeiro termo do lado direito da eq.(3.3) dado por:

$$\mathbf{z}_n = \sum_{i=1}^n [\lambda^{n-i} d_i^{2j-1} \mathbf{u}_i] \quad (3.4)$$

E a matriz de correlação do vetor de entrada de dimensão $L \times L$ definida pelo segundo termo da eq.(3.3),

$$\Phi_n = (2j-2) \sum_{i=1}^n [\lambda^{n-i} d_i^{2j-1} \mathbf{u}_i \mathbf{u}_i^T] \quad (3.5)$$

Dessa forma, podemos escrever a eq.(3.2) como,

$$\nabla J_n \approx -2j [\mathbf{z}_n - \Phi_n \mathbf{w}_n] \quad (3.6)$$

Assim, o valor ótimo do vetor peso, $\hat{\mathbf{w}}_n$, para a eq.(3.1) atinja o valor mínimo é definido pela equação normal escrita em forma matricial

$$\hat{\mathbf{w}}_n = \Phi_n^{-1} \mathbf{z}_n \quad (3.7)$$

A regra de atualização para o peso $\mathbf{w}_n$, segue o desenvolvimento:

Isolando o termo correspondente a $i = n$ na eq.(3.5), podemos escrever

$$\Phi_n = \lambda \Phi_{n-1} + [(2j-1) d_i^{2j-1}] \mathbf{u}_i \mathbf{u}_i^T \quad (3.8)$$

Analogamente podemos escrever a eq.(3.4) como,

$$\mathbf{z}_n = \lambda \mathbf{z}_{n-1} + [d_i^{2j-1}] \mathbf{u}_i \quad (3.9)$$

A forma das equações (3.8) e (3.9) facilita o cálculo da matriz de correlação Φ_n , que é necessário de acordo com a eq.(3.7).

Considerando que a matriz de correlação positiva definida positiva e não singular, aplica-se o lema de inversão de matrizes, eq.(2.26), para equação recursiva (3.8), onde são feitas as seguintes atribuições:

$$\mathbf{A} = \Phi_n \quad (3.10)$$

$$\mathbf{B}^{-1} = \lambda \Phi_{n-1} \quad (3.11)$$

$$\mathbf{C} = \mathbf{u}_n \quad (3.12)$$

$$D^{-1} = (2j - 1)d_n^{2j-2} \quad (3.13)$$

Substituindo (3.13) na eq.(2.27) do lema de inversão das matrizes, obtemos a relação de recursividade da matriz Φ_n :

$$\Phi_n^{-1} = \lambda^{-1} \Phi_{n-1}^{-1} - \frac{\lambda^{-1} \Phi_{n-1}^{-1} \mathbf{u}_n \mathbf{u}_n^T \lambda^{-1} \Phi_{n-1}^{-1}}{((2j - 1)d_n^{2j-1})^{-1} + \lambda^{-1} \mathbf{u}_n^T \Phi_{n-1}^{-1} \mathbf{u}_n} \quad (3.14)$$

Por conveniência computacional, escreve-se as seguintes igualdades,

$$\mathbf{P}_n = \Phi_{n-1}^{-1} \quad (3.15)$$

e

$$\mathbf{g}_n = \frac{\lambda^{-1} \mathbf{P}_{n-1} \mathbf{u}_n}{((2j - 1)d_n^{2j-1})^{-1} + \lambda^{-1} \mathbf{u}_n^T \mathbf{P}_{n-1} \mathbf{u}_n} \quad (3.16)$$

sendo $\mathbf{P}_n$ a inversa da matriz de autocorrelação de dimensão $L \times L$ e $\mathbf{g}_n$ o vetor ganho de dimensão $L \times 1$.

Assim podemos escrever a Equação (3.14) como

$$\mathbf{P}_n = \lambda^{-1} \mathbf{P}_{n-1} - \lambda^{-1} \mathbf{g}_n \mathbf{u}_n^T \mathbf{P}_{n-1} \quad (3.17)$$

Para facilitar futuras manipulações matemáticas, podemos reescrever a eq. (3.16) como segue:

$$((2j - 1)d_n^{2j-1})^{-1} \mathbf{g}_n = \mathbf{P}_n \mathbf{u}_n \quad (3.18)$$

3.2.1 Atualização do vetor peso

No desenvolvimento da estrutura de adaptação do algoritmo RNQA proposto utilizaremos as equações (3.7), (3.9) e (3.15) para expressar a estimativa vetor peso $\hat{w}_n$. Assim,

$$\begin{aligned} \hat{\mathbf{w}}_n &= \Phi_n \mathbf{z}_n \\ \hat{\mathbf{w}}_n &= \lambda \mathbf{P}_n \mathbf{z}_{n-1} + d_n^{2j-1} \mathbf{P}_n \mathbf{u}_n \end{aligned} \quad (3.19)$$

Substituindo a eq.(3.17) na eq.(3.20) obtemos

$$\begin{aligned}\hat{\mathbf{w}}_n &= \mathbf{P}_{n-1} \mathbf{z}_{n-1} - \mathbf{g}_n \mathbf{P}_{n-1} \mathbf{z}_{n-1} + \mathbf{P}_n \mathbf{u}_n \\ &= \hat{\mathbf{w}}_{n-1} + \mathbf{g}_n^T \hat{\mathbf{w}}_{n-1} + d_n^{2j-1} \mathbf{P}_n \mathbf{u}_n\end{aligned}\quad (3.20)$$

Usando o fato de que $\mathbf{P}_n \mathbf{u}_n$ é uma fatoração de $\mathbf{g}_n$, como mostrado na eq.(3.18), obtemos a seguinte equação,

$$\begin{aligned}\hat{\mathbf{w}}_n &= \hat{\mathbf{w}}_{n-1} - \mathbf{g}_n^T \mathbf{u}_n^T \hat{\mathbf{w}}_{n-1} + \frac{d_n}{(2j-1)} \mathbf{g}_n \\ &= \hat{\mathbf{w}}_{n-1} + \mathbf{g}_n \left[\frac{d_n}{(2j-1)} - \mathbf{u}_n^T \hat{\mathbf{w}}_{n-1} \right]\end{aligned}\quad (3.21)$$

Podemos ver que o termo entre colchetes na eq.(3.21) é a estimativa a priori do erro, o qual denotamos por,

$$\epsilon_n = \frac{d_n}{(2j-1)} - \mathbf{u}_n^T \hat{\mathbf{w}}_{n-1} \quad (3.22)$$

Assim a estrutura de atualização do algoritmo é dada por:

$$\hat{\mathbf{w}}_n = \hat{\mathbf{w}}_{n-1} + \mathbf{g}_n \epsilon_n \quad (3.23)$$

3.2.2 Resumo do algoritmo

Na Tabela (1) temos o resumo do algoritmo.

Inicialização do algoritmo RNQ
$\mathbf{P}_0 = \sigma^{-1} \mathbf{I} \quad \mathbf{w}_0 = 0$
Para cada instante no tempo $n = 1, 2, \dots$, computar
$\mathbf{g}_n = \frac{\mathbf{P}_{n-1} \mathbf{u}_n}{\left[(2j-1) d_n^{2j-1} \right]^{-1} + \lambda^{-1} \mathbf{u}_n^T \mathbf{P}_{n-1} \mathbf{u}_n}$
$\epsilon_n = \frac{d_n}{(2j-1)} - \hat{\mathbf{w}}_{n-1}^T \mathbf{u}_n$
$\hat{\mathbf{w}}_n = \hat{\mathbf{w}}_{n-1} + \mathbf{g}_n \epsilon_n$
$\mathbf{P}_n = \lambda^{-1} \mathbf{P}_{n-1} - \lambda^{-1} \mathbf{g}_n \mathbf{u}_n^T \mathbf{P}_{n-1}$

Tabela 1 – Resumo do Algoritmo RNQ para uma potência par do erro

3.3 Convergência do algoritmo

Esta seção apresenta estudo da convergência do algoritmo RNQ no contexto de um problema de identificação de sistemas adaptativos, da mesma forma que estudamos a convergência do algoritmo RNQ. 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 eq.(3.24),

$$d_n = \mathbf{w}_*^T \mathbf{u}_n + e_n \quad (3.24)$$

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

3.3.1 O comportamento médio do vetor peso no algoritmo

De (3.4), (3.7) e (3.24) obtemos

$$\hat{\mathbf{w}}_n = \Phi_n^{-1} \left\{ \sum_{i=1}^n \left[\lambda^{n-i} \mathbf{u}_i (\mathbf{w}_*^T \mathbf{u}_i + e_i)^{2j-1} \right] \right\} \quad (3.25)$$

Utilizando o desenvolvimento binomial do termo $(\mathbf{w}_*^T \mathbf{u}_i + e_i)^{2j-1}$, e desprezando os termos de alta potência (considerando que esses termos são próximo de zero), podemos reescrever a eq.(3.25) como

$$\hat{\mathbf{w}}_n \cong \mathbf{w}_* + \Phi_n^{-1} \sum_{i=1}^n \left[\lambda^{n-i} \mathbf{u}_i e_i^{2j-1} \right] \quad (3.26)$$

Aplicando o operador esperança em ambos os membros da eq. eqrefeq:reg1Rnq1 e sendo $\mathbf{u}_i$ e e_i são independentes para todo i , sendo de media zero, obtemos,

$$E[\hat{\mathbf{w}}_n] = \mathbf{w}_* \quad (3.27)$$

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.

3.3.2 Matriz de correlação do vetor desvio

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

$$\mathbf{v}_n = \hat{\mathbf{w}}_n - \mathbf{w}_* \quad (3.28)$$

sendo $\mathbf{w}_*$ a solução ótima.

Da eq.(3.26) e eq.(3.28), obtemos

$$\mathbf{v}_n = \Phi_n^{-1} \sum_{i=1}^n \left[\lambda^{n-i} \mathbf{u}_i e_i^{2j-1} \right]. \quad (3.29)$$

Definindo a matriz de correlação do vetor desvio da seguinte forma

$$\mathbf{K}_n = E[\mathbf{v}_n \mathbf{v}_n^T] \quad (3.30)$$

Substituindo a eq.(3.28) em (3.31) e sendo que $(\Phi_n^{-1})^T = \Phi_n^{-1}$, 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, (ELEFThERIOU; FALCONER, 1986)(SAYED, 2003), obtemos,

$$\mathbf{K}_n = \sigma^2 E[\Phi_n^{-1} (\mathbf{U}_n \Lambda_n \Lambda_n \mathbf{U}_n^T) \Phi_n^{-1}] \quad (3.31)$$

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

$$\mathbf{U}_n = [\mathbf{u}_1 \mathbf{u}_2 \dots \mathbf{u}_n]. \quad (3.32)$$

Pra rigoroso cálculo da eq.(3.31) devemos fazer as seguintes hipóteses (FARHANG-BOROUJENY, 2013):

- O vetor de entrada $\mathbf{u}_i$ possui amostras de um processo ergótico. Assim podemos usar as médias do tempo ao invés do conjunto de médias.
- O fator de esquecimento λ é muito próximo de 1.
- O tempo n na qual $\mathbf{K}_n$ é calculado é muito grande.

Sendo que a matriz Φ_n na eq. (3.5) é uma soma ponderada dos produtos internos

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

Assim, considerando as hipóteses acima, temos que

$$\Phi_n \approx (2j-1) d_n^{2j-2} \frac{1-\lambda^n}{1-\lambda} \mathbf{R} \quad (3.33)$$

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

Analogamente, obtemos,

$$(\mathbf{U}_n \Lambda_n \Lambda_n \mathbf{U}_n^T) \approx \frac{1-\lambda^{2n}}{1-\lambda^2} \mathbf{R}, \quad (3.34)$$

Substituindo eq.(3.34) e eq.(3.33) em (3.31) temos

$$\mathbf{K}_n = \sigma^2 \left\{ \frac{1}{[(2j-1)d_n^{2j-2}]^2} \left(\frac{1-\lambda}{1+\lambda} \right) \left(\frac{1+\lambda^n}{1-\lambda^n} \right) \mathbf{R}^{-1} \right\} \quad (3.35)$$

No estado estacionário, isto é, $n \rightarrow \infty$, e usando a eq.(3.35), obtemos,

$$\mathbf{K}_\infty = \sigma^2 \left\{ \frac{1}{[(2j-1)d_n^{2j-2}]^2} \left(\frac{1-\lambda}{1+\lambda} \right) \mathbf{R}^{-1} \right\} \quad (3.36)$$

3.3.3 Curva de aprendizagem

Das equações (3.22), (3.24) e (3.28) obtemos

$$\epsilon_n = \frac{1}{(2j-1)} (e_n - \mathbf{v}_{n-1}^T \mathbf{u}_n) \quad (3.37)$$

sendo $\mathbf{v}_{n-1}$ o vetor desvio no tempo $n-1$. O índice de desempenho estatístico para o algoritmo é o erro de estimativa a priori ϵ_n para definir o valor esperado do erro de grau $2j$,

$$\xi_n = E[|\epsilon_n|^{2j}], \quad (3.38)$$

sendo j um número positivo. Substituindo eq.(3.36) em (3.38), obtemos,

$$\xi_n = \frac{1}{(2j-1)^{2j}} E[|e_n - \mathbf{v}_{n-1}^T \mathbf{u}_n|^{2j}] \quad (3.39)$$

Utilizando o desenvolvimento binomial e observando que $\mathbf{v}_{n-1}$ é próximo de zero, após o estado transitório, podemos desconsiderar alguns termos do lado direito da eq.(3.39) 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 $E[|e_n|^{2j}]$ é o $2j$ -ésima potência de e_n , temos,

$$\xi_n \approx \frac{1}{(2j-1)^{2j}} E[e_n^{2j}] + \frac{1}{(2j-1)^{2j}} j(2j-1) E[e_n^{2j-2}] Tr\{\mathbf{R}\mathbf{K}_{n-1}\} + \frac{1}{(2j-1)^{2j}} Tr\{(\mathbf{R}\mathbf{K}_{n-1})^j\} \quad (3.40)$$

Substituindo eq.(3.35) em eq.(3.40) e definindo $\xi_{min} = \frac{1}{(2j-1)^{2j}} E[e_n^{2j}]$ como o mínimo de ϵ_n , quando $\mathbf{w} = \mathbf{w}_*$, obtemos a seguinte expressão do algoritmo,

$$\begin{aligned} \xi_n \approx \xi_{min} + \frac{j}{(2j-1)^{2j+1} d_n^{4j-4}} \left(\frac{1-\lambda}{1+\lambda} \right) \left(\frac{1+\lambda^{n-1}}{1-\lambda^{n-1}} \right) L \sigma^2 E[|e_n|^{2j-2}] \\ + \frac{1}{(2j-1)^{4j} d_n^{4j-4}} \frac{(1-\lambda)^j}{(1+\lambda)^j} \frac{(1+\lambda^{n-1})^j}{(1-\lambda^{n-1})^j} L (\sigma^2)^j \end{aligned} \quad (3.41)$$

sendo L a ordem do filtro.

A eq.(3.41) descreve a curva de aprendizagem do algoritmo RNQ para uma potencia par de erro.

3.3.4 Análise do tempo de aprendizagem

Analogamente a análise feita na dedução da constante de tempo do [RLS](#), no capítulo anterior, podemos verificar que a velocidade na qual segundo e terceiro termos do lado direito da eq.(3.41) convergem é determinada pelos termos exponenciais λ^n e λ^{nj} , respectivamente. Assim, podemos obter o tempo de convergência associado ao algoritmo, definido por τ_{RNQ1} e τ_{RNQ2} , da seguinte forma: primeiro, para o segundo termo do lado direito da eq.(3.41) temos,

$$\lambda^n = e^{\frac{-n}{\tau_{RNQ1}}} \quad (3.42)$$

que nos dá,

$$\tau_{RNQ1} = \frac{-1}{\ln(\lambda)} = \tau_{RLS} \quad (3.43)$$

Analogamente obtemos para o último termo do lado direito da eq.(3.41),

$$\lambda^{nj} = e^{\frac{-n}{\tau_{RNQ2}}} \quad (3.44)$$

$$\tau_{RNQ2} = \frac{-1}{j \ln(\lambda)} \quad (3.45)$$

Assim, definindo-se os tempos de convergência associados como τ_{RNQ1} e τ_{RNQ2} , obtém-se:

$$\lambda^n = e^{-n/\tau_{RNQ1}} \quad (3.46)$$

de onde segue

$$\tau_{RNQ1} = \frac{-1}{\ln(\lambda)} = \tau_{RLS} \quad (3.47)$$

De forma análoga,

$$\lambda^{nj} = e^{-n/\tau_{RNQ2}} \quad (3.48)$$

resultando em

$$\tau_{RNQ2} = \frac{-1}{j \ln(\lambda)} \quad (3.49)$$

Como $j > 1$, conclui-se que $\tau_{RNQ2} < \tau_{RNQ1}$, caracterizando um modo transitório adicional mais rápido no algoritmo [RNQ](#).

3.3.5 Desajuste

O desajuste do RNQ, M_{RNQ} , é definido como a diferença dimensional entre o erro quadrático médio MSE, e o MSE mínimo. Em outras palavras, esta é uma medida normalizada do custo de adaptação (WALACH; WIDROW, 1984).

O desajuste M do RLS é dado por (FARHANG-BOROUJENY, 1999),

$$M_{RLS} = \left(\frac{1-\lambda}{1+\lambda}\right)L \quad (3.50)$$

Para o algoritmo RNQ,

$$M_{RNQ} = \frac{\lim_{n \rightarrow \infty} (\xi_n - \xi_{min})}{\xi_{min}} \approx \left(\frac{1-\lambda}{1+\lambda}\right) \frac{j}{(2j-1)^{2j+1} d_n^{4j-4}} L \quad (3.51)$$

3.3.6 Análise comparativa

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

A curva de aprendizagem dada por (SAYED, 2003):

$$\xi_{nRLS} = \sigma^2 + \left(\frac{1-\lambda}{1+\lambda}\right) \left(\frac{1+\lambda^{n-1}}{1-\lambda^{n-1}}\right) \mathbf{R}^{-1} L \sigma^2 \quad (3.52)$$

Conforme a eq.(3.52), observa-se que o erro mínimo do algoritmo RLS é σ^2 e que o estado transitório é governado pelo segundo termo dessa equação.

Por outro lado, na eq.(3.41), observa-se que a curva de aprendizagem do algoritmo RNQ é composto por três por três estágios durante o aprendizado. O estágio estacionário é dado ξ_{min} . 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 masi rápido que o segundo termo na eq.(3.41) quando n tende para o infinito.

Para comparar a velocidade de adaptação dos dois algoritmos fazemos a relação entre τ_{RLS} e τ_{RNQ_2}

$$\frac{\tau_{RLS}}{\tau_{RNQ_2}} = j \quad (3.53)$$

sendo j positivo.

Da eq. (3.53), para $j = 1, \tau_{RLS} = \tau_{RNQ_2}$ e para $j > 1, \tau_{RNQ_2} < \tau_{RLS}$. Portanto observamos que a velocidade de convergência do algoritmo RNQ é mais rápida que a do algoritmo RLS.

Das eq.(3.50) e eq.(3.51) temos que o desajuste $M_{RNQ} = M_{RLS}$ para $j = 1$ e $M_{RNQ} = M_{RLS}$, para $j > 1$.

3.3.7 Complexidade computacional

Da eq.(3.23) é extraída a complexidade computacional do algoritmo para uma potência par do erro RNQ (SILVA et al., 2015). O algoritmo requer duas divisões, $L^2 + 5L + 2j$ multiplicações, $L^2 + 3L$ adições por interação, para um filtro de ordem L . Assim, esse algoritmo requer $2j - 1$ multiplicações e uma divisão escalar extras por interação quando o comparado com o algoritmo RLS padrão (HAYKIN, 1991). A complexidade computacional é da ordem de $O(L^2)$ que é comparável com o do algoritmo RLS.

4 Algoritmo RNQ baseado na soma de potências pares do erros autorregulado (RNQA)

4.1 Introdução

Neste capítulo apresentamos o Algoritmo Recursivo Não Quadrático Autorregulado (RNQA), derivado do algoritmo RNQ (SILVA et al., 2015). O RNQA baseia-se em uma função custo composta por somas ponderadas das potências pares do erro, utilizada como critério de minimização na dedução do algoritmo proposto. O aspecto chave da abordagem é que além de incluir o erro instantâneo na determinação do vetor ganho, que controla efetivamente a velocidade de adaptação do algoritmo, é a inclusão explícita do erro como termo adicional na atualização da matriz de auto covariância caracterizando uma regularização adaptativa que potencializa a taxa de convergência do algoritmo.

4.2 Dedução do algoritmo

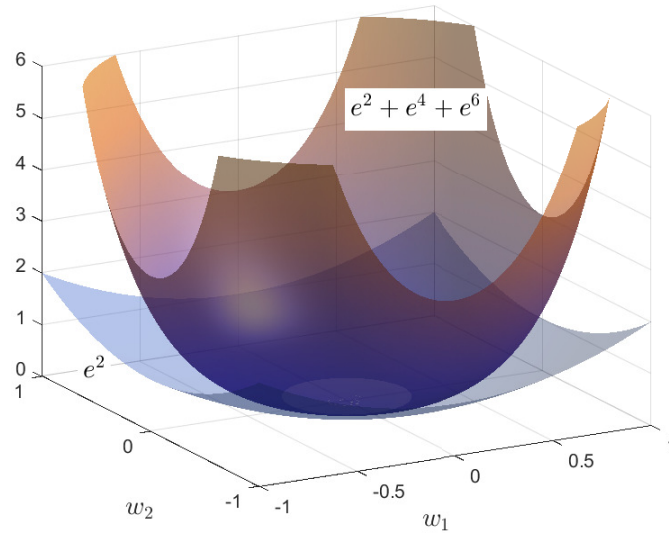
A estrutura básica do filtro adaptativo é composta do sinal desejado d_i , um vetor de entrada u_i e um vetor erro usado para atualizar o vetor peso do filtro com objetivo de alcançar uma aproximação de d_i pela estimação do sinal de saída $y_i = w_{n-1}^T u_i$ após calcular o erro, $e_i = d_i - w_{n-1}^T u_i$, sendo $0 \leq i \leq n$ representa a amostra atual e L o comprimento do filtro.

Seja J_n uma função par, contínua e simétrica baseada na soma das potências pares dos erros definida por

$$J_n = \sum_{j=1}^m k^{m-j} E[\lambda e_i^{2j}] \quad (4.1)$$

sendo m, j, k inteiros positivos. Outra característica dos elementos deste conjunto de funções é que para um valor fixo de m , podemos determinar intervalos $[-\delta, \delta]$, onde as curvas destas funções apresentam inclinação maior do que a curva da função quadrática, nesse mesmo intervalo. Esta característica é observada na Figura(3) onde estão plotadas os gráficos das funções $(e^2 + e^4 + e^6)$ e e^2 . Observar que estas funções não possuem um mínimo local, possuem apenas um mínimo global.

Figura 3 – Gráficos das funções $(e^2 + e^4 + e^6)$ e e^2 , onde podemos ver a maior inclinação da primeira, no intervalo $[-1; 1]$



Fonte: Elaborado pelo autor.

Para desenvolver uma generalização do algoritmo [RNQA](#), definimos da eq.(4.1) os seguinte risco empírico,

$$J_n = \sum_{j=1}^m k^{m-j} \sum_{i=1}^n \left\{ \lambda^{n-i} [e_i]^{2j} \right\} \quad (4.2)$$

sendo m, j, k inteiros positivos. Se $j = 1, k = 1, m = 1$ a expressão (4.2) equivale função custo [MSE](#). O fator peso exponencial λ é o fator de esquecimento convencional selecionado menor que zero e próximo de um, $0 \ll \lambda < 1$. A ponderação k^{m-j} é um termo multiplicativo que acelera a convergência do algoritmo. Para $j > 1$ define a potência par do erro e m adiciona diferentes potências do erro determinando a função custo como uma soma das potências pares do erro. Quanto maior o j mais inclinada é a a função de custo e maior o gradiente, exceto perto do mínimo cuja a inclinação se torna basicamente plana. Portanto um algoritmo de gradiente descendente utilizando um único valor de j terá uma diminuição muito rápida do erro, mas em seguida é feito uma busca do valor mínimo.

Esta intuição pode ser traduzida matematicamente da seguinte maneira: à medida que a potência do erro aumenta, o espalhamento dos autovalores da matriz de autocorrelação de entrada aumenta proporcionalmente. Portanto se $j = 2$ e $m = 1$ essa função de custo se iguala à função média do quarto erro (*least mean fourth* - [LMF](#)) ([WALACH; WIDROW, 1984](#)). A função custo do algoritmo do LMF apresenta, basicamente, uma grande inclinação quando comparada com a função de custo [MSE](#) em que todas as áreas da curva, com exceção de uma vizinhança do mínimo. Se $j > 1$ efetivamente escolhe uma potência par do erro e m adiciona as diferentes potências do erro. Quanto maior o valor j , mais íngreme

é a função de custo e maior o gradiente, exceto perto do mínimo cuja inclinação se torna basicamente plana. Portanto um algoritmo de gradiente descendente utilizando um único valor de j terá uma diminuição muito rápida do erro, mas em seguida é feito uma busca do valor mínimo.

Em (SILVA et al., 2015) é mostrado que a função de custo com $k = 1$, $m = 1$ e $j > 1$ pode ser vista como um algoritmo RLS padrão, mas onde observa que esse algoritmo utilizando uma única potência par do erro não é ótimo, pelo fato dessa função apresentar na vizinhança do mínimo uma inclinação basicamente plana, como discutido acima.

Em seguida, é mostrado como da eq.(4.2) pode ser usada para obter uma nova família de algoritmos do tipo RLS ainda mais rápidos em relação à velocidade de convergência.

Para obter o vetor peso ótimo $\hat{\mathbf{w}}_n$, é calculado o gradiente instantâneo da função J_n . Assim derivando a eq.(4.2) em relação a $\hat{\mathbf{w}}$, obtemos,

$$\begin{aligned}\nabla J_n &= \sum_{j=1}^m \left\{ -2jk^{m-j} \sum_{i=1}^n [\lambda^{n-i} e_i^{2j-1} \mathbf{u}_i] \right\} \\ &= \sum_{j=1}^m \left\{ -2a_j \sum_{i=1}^n [\lambda^{n-i} e_i^{\alpha_j} \mathbf{u}_i] \right\}\end{aligned}\tag{4.3}$$

sendo $a_j = k^{m-j}$ e $\alpha_j = 2j - 2$.

Assim, igualando ∇J a zero para que a função da eq.(4.2) atinja o valor mínimo, definimos o valor ótimo do vetor peso, $\hat{\mathbf{w}}_n$, pela equação na forma matricial

$$\hat{\mathbf{w}}_n = \Phi_n^{-1} \mathbf{z}_n\tag{4.4}$$

sendo Φ_n a matriz de autocorrelação do vetor de entrada de dimensão $L \times L$ definida por,

$$\Phi_n = \sum_{j=1}^m \left\{ a_j \sum_{i=1}^n \{ \lambda^{n-i} e_i^{\alpha_j} u_i^T u_i \} \right\}\tag{4.5}$$

e o vetor de correlação cruzada $\mathbf{z}_n$, entre as entradas e a resposta desejada com dimensões $L \times 1$, é definido por,

$$\mathbf{z}_n = \sum_{j=1}^m \left\{ a_j \sum_{i=1}^n [\lambda^{n-i} e_i^{\alpha_j} d_i \mathbf{u}_i^T] \right\}\tag{4.6}$$

Tanto a função de autocorrelação, eq. (4.5), quanto o vetor de correlação cruzada, eq.(4.6), são agora funções explícitas do erro instantâneo, diferentemente das grandezas convencionais em filtragem adaptativa, o que implica que o erro instantâneo afeta a forma da superfície de desempenho, mas a solução ótima ainda é independente do erro e a solução

de Wiener, o que implica maior controle sobre a dinâmica da aprendizagem sem afetar a solução final.

Isso significa que o erro instantâneo agora influencia o formato da superfície de desempenho, embora a solução ótima permaneça independente do erro e continue sendo a solução de Wiener (SILVA et al., 2015). Isso implica que o algoritmo ganha mais controle sobre a dinâmica de aprendizado sem afetar a solução final.

As equações recursivas para Φ_n e $\mathbf{z}_n$ são obtidos isolando o termo correspondente a $i = n$ do restante da soma no lado direito das equações. Da eq.(4.5), podemos escrever

$$\Phi_n = \lambda \Phi_{n-1} + \sum_{j=1}^m [a_j e_n^{\alpha_i}] \mathbf{u}_n \mathbf{u}_n^T \quad (4.7)$$

$$\mathbf{z}_n = \lambda \mathbf{z}_{n-1} + \sum_{j=1}^m [a_j e_n^{\alpha_i}] d_n \mathbf{u}_n \quad (4.8)$$

sendo Φ_{n-1} , o valor anterior da matriz de autocorrelação e $\mathbf{z}_n$ o valor anterior do vetor de correlação cruzada. Portanto, temos a seguinte recursão para atualizar o valor da matriz de correlação das entradas

As deduções das equações (4.7) e (4.8) facilita o cálculo da inversa da matriz de autocorrelação Φ_n , que é necessário para determinar o valor estimado, $\hat{\mathbf{w}}_n$, do vetor ótimo de acordo com a eq.(4.4).

Como a matriz de autocorrelação é positivamente definida e não singular, podemos aplicar o lema de inversão de matrizes para equação recursiva (4.7), fazendo as seguintes atribuições,

$$\begin{aligned} \mathbf{A} &= \Phi_n \\ \mathbf{B}^{-1} &= \lambda^{-1} \Phi_{n-1}^{-1} \\ \mathbf{C} &= \mathbf{u}_n \\ \mathbf{D}^{-1} &= \sum_{j=1}^m \{a_j e_n^{\alpha_i}\} \end{aligned} \quad (4.9)$$

Substituindo as definições (4.9) na eq. (2.27) obtemos a relação de recursividade para a inversa da matriz de autocorrelação Φ_n

$$\Phi_n^{-1} = \lambda^{-1} \Phi_{n-1}^{-1} - \frac{\lambda^{-1} \Phi_{n-1}^{-1} \mathbf{u}_n \mathbf{u}_n^T \Phi_{n-1}^{-1}}{\lambda [\sum_{j=1}^m a_j e_n^{\alpha_j}]^{-1} + \mathbf{u}_n^T \Phi_{n-1}^{-1} \mathbf{u}_n} \quad (4.10)$$

Por conviniência de notação, escreve-se as seguintes igualdades

$$\mathbf{P}_n = \Phi_{n-1}^{-1} \quad (4.11)$$

e

$$\mathbf{g}_n = \frac{\mathbf{P}_{n-1} \mathbf{u}_n}{\lambda [\sum_{j=1}^m a_j e_n^{\alpha_j}] + \mathbf{u}_n^T \mathbf{P}_{n-1} \mathbf{u}_n} \quad (4.12)$$

sendo $\mathbf{g}_n$ o vetor ganho de dimensão $L \times 1$.

Em comparação com o vetor ganho convencional do algoritmo [RLS](#), observa-se que a informações de erro instantâneo, que aparece no termo do somatório, modula diretamente o fator de esquecimento λ . O ganho não é apenas dependente da dinâmica do sinal de entrada, mas também uma função da resposta desejada através do erro.

Assim podemos escrever a eq. [4.10](#) como

$$\mathbf{P}_n = \lambda^{-1} (\mathbf{P}_{n-1} - \mathbf{g}_n \mathbf{u}_n^T \mathbf{P}_{n-1}) \quad (4.13)$$

Para facilitar futuras manipulações matemáticas, podemos reescrever a eq. [\(4.12\)](#) como segue:

$$\mathbf{g}_n = \mathbf{P}_{n-1} \mathbf{u}_n \left[\sum_{j=1}^m (a_j e_n^{\alpha_j}) \right] \quad (4.14)$$

4.2.1 Atualização do vetor peso

No desenvolvimento da estrutura de adaptação do algoritmo [RNQA](#) proposto utilizaremos as equações [\(4.4\)](#), [\(4.8\)](#) e [\(4.11\)](#) para expressar a estimativa vetor peso $\hat{\mathbf{w}}_n$ do vetor peso no instante n como segue:

$$\begin{aligned} \hat{\mathbf{w}}_n &= \Phi_n^{-1} \mathbf{z}_n \\ \hat{\mathbf{w}}_n &= \lambda \mathbf{P}_n \mathbf{z}_{n-1} + \mathbf{P}_n \left[\sum_{j=1}^m a_j e_n^{\alpha_j} \right] d_n \mathbf{u}_n \end{aligned} \quad (4.15)$$

Substituindo a eq. [\(4.13\)](#) na eq. [\(4.15\)](#)

$$\begin{aligned} \hat{\mathbf{w}}_n &= \mathbf{P}_n \mathbf{z}_{n-1} - \mathbf{g}_n \mathbf{u}_n^T \mathbf{P}_n \mathbf{z}_{n-1} + \left[\sum_{j=1}^m \{a_j e_n^{\alpha_j}\} \right] \mathbf{P}_n \mathbf{u}_n d_n \\ &= \hat{\mathbf{w}}_{n-1} - \mathbf{g}_n \mathbf{u}_n^T \hat{\mathbf{w}}_{n-1} + \left[\sum_{j=1}^m \{a_j e_n^{\alpha_j}\} \right] \mathbf{P}_n \mathbf{u}_n d_n \end{aligned} \quad (4.16)$$

Das Equações [\(4.49\)](#) e [\(4.16\)](#), obtemos a estrutura de atualização do algoritmo [RNQA](#) dada pela seguinte equação,

$$\hat{\mathbf{w}}_n = \hat{\mathbf{w}}_{n-1} + \mathbf{g}_n e_n \quad (4.17)$$

sendo e_n uma estimação a priori do erro dada por:

$$e_n = d_n - \hat{\mathbf{w}}_n \mathbf{u}_n^T \quad (4.18)$$

4.2.2 Matriz de autocorrelação normalizada

Diferentemente muito algoritmos que utilizam como método básico modificar a matriz de autocovariância para não tender a zero somando fator de regularização, (HAYKIN, 1991). como uma escolha de projeto que desloca o ponto de operação ao longo do trade-off entre erro em regime permanente e capacidade de rastreamento . O algoritmo RNQ, (SILVA et al., 2015), onde define

$$\mathbf{P}_n = \lambda^{-1}\mathbf{P}_{n-1} - \lambda^{-1}\mathbf{g}_n\mathbf{u}_n^T\mathbf{P}_{n-1} + \beta\mathbf{I} \quad (4.19)$$

sendo β é uma constante positiva muito pequena e $\mathbf{I}$ matriz identidade.

Esse o fator ajuste atua diretamente na matriz na inversa da matriz de autocorrelação, conforme a notação em (4.11), $\mathbf{P}_n = \Phi_n^{-1}$, que representa a inversa da matriz de autocorrelação.

No algoritmo proposto RNQA a modificação consiste em atuar de maneira dinâmica no processo de atualização da inversa da matriz autocorrelação usando seu erro instantâneo como parâmetro de ajuste, tornando o ajuste adaptativa da matriz. Escrevendo (4.19) como:

$$\Phi_n^{-1} = \lambda^{-1}\Phi_{n-1}^{-1} - \lambda^{-1}\mathbf{g}_n\mathbf{u}_n^T\Phi_{n-1}^{-1} + \beta\mathbf{I} \quad (4.20)$$

e multiplicando ambos os lados da eq.(4.20) por e_n^2/β :

$$(e_n^2/\beta)\Phi_n^{-1} = \lambda^{-1}(e_n^2/\beta)\Phi_{n-1}^{-1} - \lambda^{-1}\mathbf{g}_n\mathbf{u}_n^T(e_n^2/\beta)\Phi_{n-1}^{-1} + (e_n^2/\beta)\beta\mathbf{I} \quad (4.21)$$

assim introduzimos o erro a priori e_n na atualização da inversa da matriz , normalizando a matriz resultando a eq.(4.20), onde definimos $\hat{\Phi}_n^{-1} = (e_n^2/\beta)\Phi_n^{-1}$.

$$\hat{\Phi}_n^{-1} = \lambda^{-1}\hat{\Phi}_{n-1}^{-1} - \lambda^{-1}\mathbf{g}_n\mathbf{u}_n^T\hat{\Phi}_{n-1}^{-1} + e_n^2\mathbf{I} \quad (4.22)$$

Por conviniência de escrita $\hat{\mathbf{P}}_n = \hat{\Phi}_n^{-1}$ representa a inversa da matriz de autocorrelação auto ajustada. Sendo $e_n^2 > 0$ segue que $e_n^2\mathbf{I} \succ 0 \implies \hat{\mathbf{P}}_n \succ 0$, ou seja, a matriz de correlação $\hat{\mathbf{P}}_n$ é positiva definida positiva e não singular

$$\hat{\mathbf{P}}_n = \lambda\hat{\mathbf{P}}_{n-1} + \lambda^{-1}\mathbf{g}_n\mathbf{u}_n^T\hat{\mathbf{P}}_{n-1} + e_n^2\mathbf{I} \quad (4.23)$$

Podemos então, escrever a equação de ganho, Eq.(4.12):

$$\mathbf{g}_n = \hat{\mathbf{P}}_{n-1}\mathbf{u}_n\left[\sum_{j=1}^m a_j e_n^{\alpha_j}\right] \quad (4.24)$$

a estrutura de atualização final é dada por:

$$\hat{\mathbf{w}}_n = \hat{\mathbf{w}}_{n-1} + \mathbf{g}_n e_n, \quad (4.25)$$

Essa forma dinâmica não só regulariza mas também ajusta a matriz de autocovariância permitindo seguir variações de entradas não estacionárias garantindo um compromisso entre o estado de alerta e a capacidade de rastreamento.

As equações (4.12), (4.23), (4.18), (4.25) respectivamente, constituem o algoritmo Recursivo Não Quadrático Autorregulado (RNQA) baseado na soma das potências pares do erro com autorregularização, resumido na Tabela 2

Inicialização do algoritmo
$\mathbf{P}_0 = \gamma^{-1} \mathbf{I} \quad \mathbf{w}_0 = 0$
Para cada instante no tempo computar $n = 1, 2, \dots$
$e_n = d_n - \hat{\mathbf{w}}_{n-1} \mathbf{u}_n^T$
$\mathbf{g}_n = \mathbf{P}_{n-1} \mathbf{u}_n \left[\lambda \left\{ \sum_{j=1}^m a_j e_n^{\alpha_j} \right\} + \mathbf{u}_n^T \mathbf{P}_{n-1} \mathbf{u}_n \right]^{-1}$
$\hat{\mathbf{w}}_n = \hat{\mathbf{w}}_{n-1} + \mathbf{g}_n e_n$
$\mathbf{P}_n = \lambda^{-1} \mathbf{P}_{n-1} - \lambda^{-1} \mathbf{g}_n \mathbf{u}_n^T \mathbf{P}_{n-1} + e_n^2 \mathbf{I}$

Tabela 2 – Equações do Algoritmo RNQA

4.2.3 Análise do vetor ganho do algoritmo RNQA

O vetor ganho do RNQA é efetivamente modulado não apenas pela dinâmica do sinal de entrada mas também pelo erro instantâneo. Da reescrita da somatória do vetor ganho da eq.(4.24) como

$$\left[\sum_{j=1}^m a_j e_n^{\alpha_j} \right] = \left[\sum_{j=1}^m j k^{m-j} e_n^{2j-2} \right] = \sum_{j=1}^m \left[\frac{k^m}{k^j} \left(\frac{e_n^{2j}}{e_n^2} \right) \right] = \sum_{j=1}^m \left[\frac{k^m}{e^2} \left(\frac{e_n^{2j}}{k^j} \right) \right] = \sum_{j=1}^m \left[\frac{k^m}{e_n^2} \left(\frac{e_n^2}{k} \right)^j \right] \quad (4.26)$$

Quando o termo que compreende o quadrado do erro sobre k é muito grande, como normalmente acontece nos momentos iniciais de aprendizagem, o ganho g_n torna-se muito grande porque a base é elevada a j , portanto quanto maior a potência na função custo mais rápida pode ser a convergência esperado.

Quando o erro quadrado sobre de k se torna muito menor a somatória dos termos entra em ação, e k é sempre elevado para a potência m , o qual balanceia o efeito do erro se tornar exponencialmente pequeno. Portanto o ganho g_n é sempre grande tanto para erro grande quanto para erros pequenos.

A estrutura de atualização do RNQA similar a do algoritmo do RLS padrão, mas com mais flexibilidade em termos de velocidade de convergência, onde o erro incluído no vetor ganho $\mathbf{g}_n$ e na matriz de autocovariância $\mathbf{P}_n$, fornece uma maior velocidade de

convergência ao algoritmo. Esse fato é muito importante para entendermos sobre quais condições o algoritmo RNQA atingirá a solução ótima do MSE. Dessa forma, podemos rescrever as Equações (4.7) e (4.8) em termos de valor esperado como segue

$$\Phi_n = \sum_{j=1}^m \{a_j \lambda E[e_n^{\alpha_i} \mathbf{u}_n \mathbf{u}_n^T]\} \quad (4.27)$$

$$\mathbf{z}_n = \sum_{j=1}^m \{a_j \lambda E[e_n^{\alpha_i} d_n \mathbf{u}_n]\} \quad (4.28)$$

Se assumirmos que o erro é estatisticamente independente do sinal de entrada e do sinal desejado, temos

$$\Phi_n = \sum_{j=1}^m \{a_j \lambda E[e_n^{\alpha_i}] E[\mathbf{u}_n \mathbf{u}_n^T]\} \quad (4.29)$$

$$\mathbf{z}_n = \sum_{j=1}^m \{a_j \lambda E[e_n^{\alpha_i}] E[d_n \mathbf{u}_n]\} \quad (4.30)$$

A solução ótima torna-se

$$\hat{\Phi}_n^{-1} \mathbf{z}_n = E[\mathbf{u}_n \mathbf{u}_n]^{-1} \frac{E[d_n \mathbf{u}_n] \sum_{j=1}^m \{a_j \lambda E[e_n^{\alpha_i}]\}}{\sum_{j=1}^m \{a_j \lambda E[e_n^{\alpha_i}]\}} = \mathbf{R}^{-1} \mathbf{P} \quad (4.31)$$

que, de fato, é a solução MSE. Entretanto, o algoritmo RNQA, seguirá, na maioria das vezes, a solução MSE durante a adaptação.

4.2.4 Interpretação do efeito da matriz de covariância normalizada pelo quadrado do erro no algoritmo RNQA

Nesta seção o objetivo é analisar o mecanismo de ajuste adaptativo na matriz de autocovariância no algoritmo RNQA, mais especificadamente a influência do erro instantâneo sobre a dinâmica de atualização da matriz, como contribui para o comportamento adaptativo do algoritmo, afetando sua velocidade de convergência e estabilidade. Para isso, parte-se da expressão da curva de aprendizagem média, a qual relaciona o desempenho do algoritmo com a evolução do erro de estimação ao longo do tempo.

A curva de aprendizagem pode ser expressa como a soma entre o erro mínimo alcançável e um termo adicional associado ao traço do produto entre a matriz de covariância do sinal de entrada e a matriz de correlação do erro de estimação, isto é:

$$\xi(n) = \xi_{\min} + \text{Tr}(\mathbf{R} \mathbf{K}(n-1)), \quad (4.32)$$

onde $\xi_{\min}$ representa o erro mínimo alcançável pelo algoritmo, $\mathbf{R} = E[\mathbf{u}_n \mathbf{u}_n^T]$ é a matriz de covariância do sinal de entrada $\mathbf{u}_n$, e $\mathbf{K}_n$ denota a matriz de correlação do erro de estimação.

Nesse contexto, a matriz de correlação do erro de estimação é definida como

$$\mathbf{K}_n = E[\mathbf{v}_n \mathbf{v}_n^T], \quad (4.33)$$

em que o vetor de desvio é dado por

$$\mathbf{v}_n = \hat{\mathbf{w}}_n - \mathbf{w}_*. \quad (4.34)$$

Observando a função da curva de aprendizagem em (4.32), nota-se que ela é expressa em termos do traço do produto entre as matrizes $\mathbf{R}$ e $\mathbf{K}_{n-1}$. Nessa forma, a velocidade de convergência do algoritmo está diretamente relacionada à matriz de correlação do erro de estimação. De maneira mais geral, verifica-se que as curvas de aprendizagem são governadas pelo termo $Tr(\mathbf{R}\mathbf{K}_{n-1})$, cuja magnitude reflete a projeção do erro de estimação ponderada pela estrutura estatística do sinal de entrada. Consequentemente, a dinâmica de convergência do algoritmo pode ser analisada a partir da evolução temporal da matriz $\mathbf{K}_n$. A matriz $\mathbf{K}_n$ está associada à covariância do erro de estimação do algoritmo e relacionada à inversa da matriz de correlação $\hat{\Phi}$, utilizada no cálculo da atualização dos pesos.

Para capturar o efeito do termo $e(n)^2 \mathbf{I}$ na atualização de sua matriz. Da equação da atualização da inversa matriz (4.35):

$$\hat{\Phi}_n^{-1} = \lambda^{-1} \hat{\Phi}_{n-1}^{-1} - \lambda^{-1} \mathbf{g}_n \mathbf{u}_n^T \hat{\Phi}_{n-1}^{-1} + e_n^2 \mathbf{I} \quad (4.35)$$

consideramos apenas a contribuição associada ao erro instantâneo, escrevemos forma compacta

$$\hat{\Phi}_n^{-1} \approx \hat{\Phi}_{n-1}^{-1} + e_n^2 \mathbf{I} \quad (4.36)$$

Substituindo (4.36) na definição da matriz $\mathbf{K}_n$ em (3.31):

$$\mathbf{K}_n = \sigma^2 E[\hat{\Phi}_n^{-1} (\mathbf{U}_n \Lambda_n \Lambda_n^T \mathbf{U}_n^T) \hat{\Phi}_n^{-1}] \quad (4.37)$$

De maneira análoga ao desenvolvimento feitos nas equações (3.33), (3.34) e (3.35), obtem-se

$$\hat{\Phi}_n^{-1} + e_n^2 \mathbf{I} \approx \frac{1}{(2j-1)d_n^{2j-1}(1-\lambda^n)} \cdot \frac{1-\lambda}{1+\lambda} \cdot \frac{1+\lambda^n}{1-\lambda^n} \mathbf{R}^{-1} + e_n^2 \mathbf{I} \quad (4.38)$$

e reescrevemos $\mathbf{K}_n$ na eq.(4.37) como:

$$\mathbf{K}_n = \sigma^2 \left[\left\{ \frac{1}{(2j-1)d_n^{2j-1}(1-\lambda^n)} \cdot \frac{1-\lambda}{1+\lambda} \cdot \frac{1+\lambda^n}{1-\lambda^n} \mathbf{R}^{-1} + e_n^2 \mathbf{I} \right\} \right] \quad (4.39)$$

Efetuada a multiplicação expressão de $\mathbf{K}_n$, (4.39), pela esquerda pela matriz $\mathbf{R}$:

$$\mathbf{R}\mathbf{K}_n = \mathbf{R} \left[\sigma^2 \left\{ \frac{1}{(2j-1)d_n^{2j-1}(1-\lambda^n)} \cdot \frac{1-\lambda}{1+\lambda} \cdot \frac{1+\lambda^n}{1-\lambda^n} \mathbf{R}^{-1} + e_n^2 \mathbf{I} \right\} \right] \quad (4.40)$$

sendo $\mathbf{R}\mathbf{R}^{-1} = \mathbf{I}$ e $\mathbf{R}\mathbf{I} = \mathbf{R}$, resulta em:

$$\mathbf{R}\mathbf{K}_n = \sigma^2 \left\{ \frac{1}{(2j-1)d_n^{2j-1}(1-\lambda^n)} \cdot \frac{1-\lambda}{1+\lambda} \cdot \frac{1+\lambda^n}{1-\lambda^n} \mathbf{R} + e_n^2 \mathbf{R} \right\} \quad (4.41)$$

Para escrita simplificada tornando

$$\mu = \frac{1}{(2j-1)d_n^{2j-1}(1-\lambda^n)} \cdot \frac{1-\lambda}{1+\lambda} \cdot \frac{1+\lambda^n}{1-\lambda^n}$$

$$\mathbf{R}\mathbf{K}_n = \sigma^2 \{ \mu \mathbf{I} + e_n^2 \mathbf{R} \} \quad (4.42)$$

Aplicando o traço à matriz produto (4.42), temos :

$$\text{Tr}(\mathbf{R}\mathbf{K}_n) = \sigma^2 \{ \mu L + e_n^2 \text{Tr}(\mathbf{R}) \} \quad (4.43)$$

onde $\mathbf{R} = E[\mathbf{u}_n \mathbf{u}_n^T]$ e matriz de autocovariância, $L = \text{Tr}(\mathbf{I})$ o comprimento do filtro. Sendo $\mathbf{u}_n^T \mathbf{u}_n = \text{Tr}(\mathbf{u}_n \mathbf{u}_n^T)$, a relação entre energia média e traço da autocorrelação pode ser escrita como:

$$\begin{aligned} E[\|\mathbf{u}_n\|^2] &= E[\mathbf{u}_n^T \mathbf{u}_n] \\ &= E[\text{Tr}(\mathbf{u}_n \mathbf{u}_n^T)] \\ &= \text{Tr}(E[\mathbf{u}_n \mathbf{u}_n^T]) \\ &= \text{Tr}(\mathbf{R}), \end{aligned} \quad (4.44)$$

Substituindo a equivalência da eq. (4.44) na eq.(4.43) podemos reescrever a eq. (4.32) como:

$$\xi(n) = \xi_{\min} + \sigma^2 \{ [\mu L + e_n^2 E[\|\mathbf{u}_n\|^2]] \} \quad (4.45)$$

Avaliando o traço do produto matricial $\mathbf{R}\mathbf{K}_n$ em termos de autovalores, retornando a eq.(4.43), temos a matriz de autocovariância $\mathbf{R} = E[\mathbf{u}_n \mathbf{u}_n^T]$ simétrica e definida positiva (HAYKIN, 2014), sua decomposição espectral é dada por $\mathbf{R} = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^T$, sendo $\mathbf{Q}$ é uma

matriz ortogonal, isto é, $\mathbf{Q}^T \mathbf{\Lambda} \mathbf{Q} = \mathbf{I}$ e $\mathbf{\Lambda} = \text{diag}\{\lambda_1, \lambda_2, \dots, \lambda_L\}$ matriz diagonal de autovalores λ_i da matriz $\mathbf{R}$. Assim:

$$\text{Tr}(\mathbf{R}) = \sum_{i=1}^L \lambda_i \quad (4.46)$$

e substituindo eq.(4.46) na eq.(4.43) escrevemos:

$$\text{Tr}(\mathbf{R} \mathbf{K}_n) = \sigma^2 \{ \mu L + e_n^2 \sum_{i=1}^L \lambda_i \} \quad (4.47)$$

Então modo alternativo, escrevemos a eq.(4.32)

$$\xi(n) = \xi_{\min} + \sigma^2 \{ \mu L + e_n^2 \sum_{i=1}^L \lambda_i \} \quad (4.48)$$

onde λ_i são os autovalores de $\mathbf{R}$.

4.2.5 Conclusão

Conforme as expressões eq.(4.50), (4.47), observa-se de forma equivalente que o efeito de incluir o e_n^2 como termo de regularização na atualização da matriz de autocovariância do algoritmo RNQA, esse termo atua como fator multiplicativo na energia média do sinal de entrada, $e_n^2 E[||\mathbf{u}_n||^2]$, equivale a potencializar a velocidade de convergência, caracterizando uma autorregularização adaptativa:

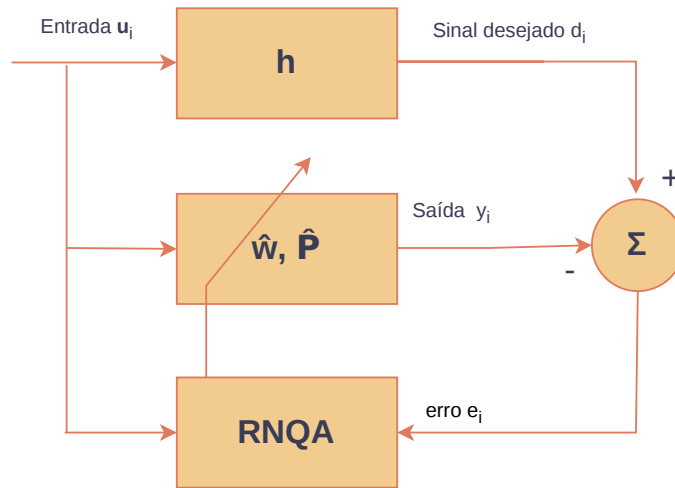
4.3 Resultados

O algoritmo proposto baseado na função (4.2), contém além dos três parâmetros livres: m , a ordem mais alta da função de custo, que também controla o número de termos na soma; o parâmetro k que funciona como um termo de ganho extra, bem como um parâmetro de escala para o erro; e o fator de esquecimento λ . Possui também uma regularização contínua de sua matriz de covariância P atualizada pela informação de erro instantâneo e_n^2 .

4.3.1 Identificação de sistema usando um filtro FIR

Para as simulações criamos uma configuração, conforme na Figura 4, um filtro passa-baixa FIR com resposta ao impulso dada por $\mathbf{h}$, comprimento $L = 30$, frequência de corte $f_c = 0,5\pi$ e $f_c = 0,2\pi$. Usamos o sinal u_i com uma distribuição gaussiana, a ser filtrado por h . O sinal desejado d_i é obtido filtrando o sinal de entrada u_i por $\mathbf{h}$. A saída do filtro adaptativo denotado por y_i .

Figura 4 – Estrutura do filtro adaptativo RNQA com configuração identificação de sistemas, sendo u_i o vetor de entrada, $\mathbf{h}$ um filtro FIR, d_i o sinal desejado, e_i o sinal de erro e y_i a saída do filtro.



Fonte: Elaborado pelo autor.

O desempenho do algoritmo RNQA é avaliado em termos da velocidade de convergência, utilizando-se a curva de aprendizagem como ferramenta de análise. Define-se como instante de convergência o número de iterações necessário para que o erro médio atinja uma faixa de 5% em relação ao erro em regime permanente. A inclusão do algoritmo RNQ (SILVA et al., 2015), baseado em termo de regularização constante, permite evidenciar o mecanismo de autorregularização proposto no RNQA. Emprega-se também como refe-

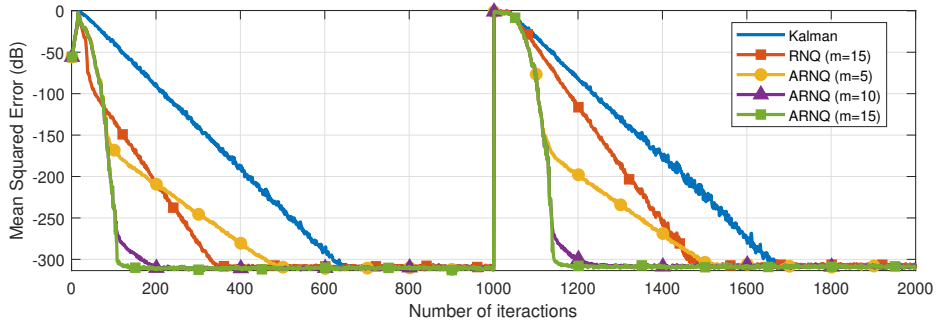
rência comparativa o algoritmo de estimação paramétrica baseado no filtro de Kalman proposto por Söderström (SÖDERSTRÖM, 1989), amplamente utilizado em problemas de rastreamento adaptativo.

Condições iniciais e parâmetros :

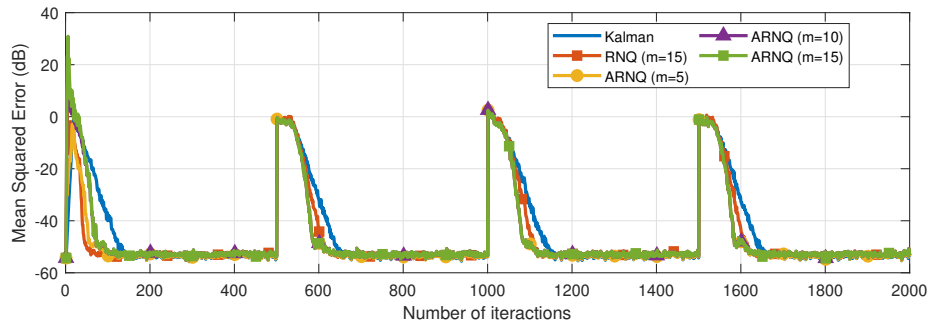
- RNQA: $\mathbf{P}_0 = 10^{-5}I, \lambda = 0,96$, ($m=5,10$ e 15), termo de regularização e_n^2 ,
- RNQ: $\mathbf{P}_0 = 10^{-5}I, \lambda = 0,96$, ($m=15$), termo de regularização constante $\beta = 10^{-4}$,
- Kalman: $\mathbf{P}_0 = 10^{-5}I, \mathbf{R}_1 = 10^4$

Nessa estrutura de identificação de sistemas, temos os resultados analisados na Figura 5.

Figura 5 – Comparação das curvas de aprendizagem do RNQA com $m = 5$ com base em e_n^2 , o Filtro de Kalman para identificação do sistema e o RNQ com ($m = 15$), usando um fator de regularização fixo de $\beta = 10^{-4}$. a) As simulações foram conduzidas com um sistema livre de ruído. b) As simulações foram conduzidas com ruído gaussiano aditivo com SNR = 60 dB.



a) Sistema sem ruído.



b) Sistema com ruído.

Fonte: Elaborado pelo autor.

4.3.2 Discussão

O RNQA com diferentes quantidades de potências pares, $m = 5$, $m = 10$ e $m = 15$ com matriz autorregulado por $e(n)^2$, resultam em diferentes taxas de convergências. Na

Figura 5(a), as curvas de aprendizado dos algoritmos com sistema sem ruído, $\eta_i = 0$, sendo que na interação $n = 1000$, os parâmetros foram alterados. Enquanto o algoritmo de Kalman (SöDERSTRÖM, 1989) levou 700 iterações para alcançar seu estado estacionário o RNQA alcança entre 100 e 500, ou seja, o RNQA foi muito mais rápido, apresentou maior velocidade de convergência do que o algoritmo RNQ e o algoritmo de Filtro de Kalman com destaque ao RNQA para $m=15$, Tabela (3).

Tabela 3 – Número aproximado de iterações para convergência dos algoritmos RNQA, RNQ e Kalman. Sistema sem ruído

Métrica	RNQA			RNQ	Kalman
	$m = 5$	$m = 10$	$m = 15$	$m = 15$	
Iterações para convergência	≈ 500	≈ 200	≈ 100	≈ 300	≈ 700

Fonte: Elaboração do autor.

Na Fig. 5(b) as simulações foram realizadas com ruído aditivo gaussiano com SNR 60dB. O mesmo comportamento ocorreu com a simulação com ruído, isto é o algoritmo RNQA apresenta desempenho melhor em relação ao RNQ e Kalman.

Na simulação com ruído, Fig. 5(b), nas primeiras interações é interessante notar certa oscilação do comportamento do RNQA onde perde em convergência em relação ao RNQ padrão com $m=15$ com fator de regularização constante $\beta = 10^{-4}$. Isso justifica porque o RNQA nos instantes iniciais o erro instantâneo e_n^2 atuando na matriz de autocovariância, $P_n = \lambda^{-1}(P_{n-1}G_n u_n^T P_{n-1}) + e_n^2 I$, é bastante grande, como esperado, e as características de parâmetros iniciais do RNQ com máximo parâmetro m esta com os melhores parâmetros iniciais apresentar melhor desempenho, Tabela (4). A vantagem do RNQA ocorre quando acontece a mudança brusca dos parâmetros. As variações do erro já adaptado atuando dinamicamente no ajuste da matriz P_n , em conjunto com a maior potência, $m = 15$, torna o RNQA em maior estado de alerta convergindo mais rápido superando os algoritmos Kalman e RNQ em convergência conforme visualizado a Fig. 5(b), onde evidencia o melhor desempenho do RNQA nos ciclos de mudança de parâmetros nas iterações 500, 1000 e 1500.

Tabela 4 – Número aproximado de iterações para convergência dos algoritmos RNQA, RNQ e Kalman. Sistema com ruído

Métrica	RNQA			RNQ	Kalman
	$m = 5$	$m = 10$	$m = 15$	$m = 15$	
Iterações para convergência	≈ 70	≈ 80	≈ 80	≈ 40	≈ 170

Fonte: Elaboração do autor.

O fato da vantagem RNQA às mudanças brusca dos parâmetros, são avaliadas

pelas eq.(4.26), (4.50):

$$\mathbf{g}_n = \mathbf{P}_{n-1} \mathbf{u}_n \left(\sum_{j=1}^m \left[\frac{k^m}{e_n^2} \left(\frac{e_n^2}{k} \right)^j \right] \right) \quad (4.49)$$

$$\xi(n) = \xi_{\min} + \sigma^2 \{ [\mu L + e_n^2 E[\|\mathbf{u}_n\|^2]] \} \quad (4.50)$$

- **Fase de aprendizagem**(mudanças brusca dos parâmetros, ($n = 500, n = 1000, n = 1500$))

Valores elevado e_n^2 : o ganho $\mathbf{g}_n$ é grande , o termo $e_n^2 E[\|\mathbf{u}_n\|^2]$ torna-se dominante potencializa a convergência mais rápida acelerando o processo de adaptação.

- **Regime permanente**

Valor reduzido de e_n^2 : o ganho $\mathbf{g}_n$ mantém grande o termo $e_n^2 E[\|\mathbf{u}_n\|^2]$ é atenuado. Isso resulta em atualizações mais suaves, contribuindo para a estabilização do algoritmo.

O ajuste dinâmico da matriz de autocovariância em função do erro permite ao algoritmo alternar entre regimes de rápida adaptação e baixa variância. O mecanismo confere ao RNQA capacidade rapidez adicional de rastreamento em cenários não estacionários.

O termo e_n^2 pode ser interpretado como um mecanismo automático de detecção de regime de operação do algoritmo: durante o regime transiente, valores elevados de erro promovem maior regularização, enquanto no regime permanente, a redução do erro implica uma atenuação natural do termo regularizador.

4.3.3 Conclusão

Nesse capítulo apresentamos o desempenho do algoritmo RNQA onde observamos que o erro instantâneo aparece na atualização do vetor ganho e na atualização da matriz de autocovariância que controla efetivamente a velocidade de adaptação do algoritmo.

Os testes foram realizados tanto para o caso em que o algoritmo é capaz de identificar o sistema com um erro desprezível (identificação perfeita) quanto quando o erro é muito grande. Através dos resultados apresentados mostramos que o algoritmo RNQA como função de custo baseado na soma ponderada das potências pares do erro e mecanismo de autorregularização adaptativa teve um desempenho melhor do que o algoritmo RNQ padrão (SILVA et al., 2015) e Kalman proposto por Söderström (SÖDERSTRÖM, 1989) em ambas as condições, em termos de velocidade de convergência.

5 Conclusão e trabalhos futuros

Nesse trabalho propomos o algoritmo [RNQA](#) com regularização dinâmica na atualização da matriz inversa de autocorrelação de maneira simples que proporciona convergência mais rápida, mantendo custo computacional equivalente ao [RNQ](#) e [RLS](#). No algoritmo [RNQA](#) o erro instantâneo tem função dupla, atuando na atualização do vetor ganho e na autorregularização da matriz de autocovariância controlando e potencializando efetivamente a velocidade de adaptação do algoritmo.

Apresentamos uma análise comparativa do [RNQA](#) com o [RNQ](#) padrão ([SILVA et al., 2015](#)) e o algoritmo de Kalman ([SÖDERSTRÖM, 1989](#)) no que se diz respeito à velocidade de convergência nos casos de sinais puros e corrompidos por ruídos gaussianos.

Os testes foram realizados tanto para o caso em que o algoritmo é capaz de identificar o sistema com um erro desprezível (identificação perfeita) quanto o erro é muito grande.

Os resultados apresentados mostram que o algoritmo [RNQA](#) é uma escolha alternativa muito útil na condição de aprendizagem, onde a estatística do sinal muda com o tempo, uma vez que tem um melhor desempenho na velocidade de convergência sem aumento considerável na complexidade computacional.

Em relação a trabalhos futuros, algumas possibilidades merecem ser investigadas. A primeira é um estudo mais detalhado do fator de ponderação apresentado no somatório do algoritmo [RNQA](#), considerando a base desse fator como um número real qualquer. Uma possibilidade de extensão do estudo é a realização da análise de convergência do algoritmo proposto.

Referências

- ALI, A.; REHMAN, A.-U.; ALI, R. An improved gain vector to enhance convergence characteristics of recursive least squares algorithm. v. 4, 11 2010.
- BENESTY, J.; PALEOLOGU, C.; CIOCHINĂ, S. On regularization in adaptive filtering. *IEEE Transactions on Audio, Speech, and Language Processing*, v. 19, n. 4, p. 740–752, 2011.
- BJÖRCK, Å. *Numerical Methods for Least Squares Problems*. [S.l.]: SIAM, 1996.
- CAI, Y.; LAMARE, R. C. de. Low-complexity variable forgetting factor techniques for rls algorithms in interference rejection applications. 2013. Disponível em: <https://arxiv.org/abs/1302.5960>.
- ELEFThERIOU, E.; FALCONER, D. Tracking properties and steady-state performance of rls adaptive filter algorithms. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, v. 34, n. 5, p. 1097–1110, 1986.
- FARHANG-BOROUJENY, B. *Adaptive Filters: Theory and Applications*. Wiley, 1999. ISBN 9780471983378. Disponível em: <https://books.google.com.br/books?id=jxd-QgAACAAJ>.
- FARHANG-BOROUJENY, B. *Adaptive filters:theory and applications(2th Edition)*. [S.l.: s.n.], 2013. 800 p. ISSN 978-1-119-97954-8.
- GOLUB, G. H.; LOAN, C. F. V. *Matrix Computations*. 4. ed. [S.l.]: Johns Hopkins University Press, 2013.
- HAYKIN, S. *Adaptive Filter theory*. [S.l.]: Prentice-Hall, 1991.
- HAYKIN, S. *Adaptive Filter Theory (5th Edition)*. [S.l.: s.n.], 2014.
- HEBB, D. O. The organization of behavior. *Wiley*, 1949.
- KALMAN, R. E. A new approach to linear filtering and prediction problems. *Journal of basic engineering*, American Society of Mechanical Engineers, v. 82, n. 1, p. 35–45, 1960.
- LI, . Proportionate rls with l1 norm regularization: Performance analysis and fast implementation. *Digital Signal Processing*, v. 122, p. 103366, 2022.
- LJUNG, L. *System Identification: Theory for the User*. 2. ed. [S.l.]: Prentice Hall, 1999.
- MCCULLOCH, W.; PITTS, W. A logical calculus of the ideas immanent in nervous activity. *The Organization of Behavior*, 1949.
- MOMOH O. C. UBADIKE, I. A. K. M. A. I.-B. P. U. C. M. B. A. M. O. Vss-lms: Lms algorithm experimental approach. *ISSUE-2*, v. 3, 2020. ISSN 2637-0883.
- OTOPELEANU, R.-A. et al. Robust data-reuse regularized recursive least-squares algorithms for system identification applications. *Sensors*, v. 25, n. 16, p. 5017, 2025.

- ROSENBLATT, F. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological Review*, American Psychological Association, v. 65, n. 6, p. 386–408, 1958.
- SADIKI, T.; SLOCK, D. T. Steady-state performance comparison of bayesian and standard adaptive filtering. *2006 Fortieth Asilomar Conference on Signals, Systems and Computers*, p. 253–257, 2006.
- SAYED, A. H. *Fundamentals Of Adaptive Filtering*. [s.n.], 2003. Disponível em: <<https://api.semanticscholar.org/CorpusID:60598442>>.
- SCHMIDT, . Adaptive system identification with regularized fir models. In: *IFAC-PapersOnLine*. [S.l.: s.n.], 2022. v. 55, n. 12, p. 1–6.
- SILVA, C. da et al. An algorithm based on non-squared sum of the errors. *Signal Processing*, v. 117, p. 188–191, 2015. ISSN 0165-1684. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0165168415001139>>.
- SLOCK, D. T. M. On the convergence behavior of the lms and the normalized lms algorithms. *IEEE Transactions on Signal Processing*, v. 44, n. 12, p. 3011–3023, 1996.
- SÖDERSTRÖM, P. S. T. *System Identification*. [S.l.: s.n.], 1989. I. 612 p.
- TIKHONOV, A. N.; ARSENIN, V. Y. *Solutions of Ill-Posed Problems*. [S.l.]: Winston and Sons, 1977.
- TURING, A. M. On computable numbers, with an application to the entscheidungsproblem. *Proceedings of the London Mathematical Society*, s2-42, n. 1, p. 230–265, 1936.
- WALACH, E.; WIDROW, B. The least mean fourth (lmf) adaptive algorithm and its family. *IEEE Transactions on Information Theory*, IEEE, v. 30, n. 2, p. 275–283, 1984.
- WIDROW, B.; HOFF, M. E. Adaptive switching circuits. In: *IRE WESCON Convention Record*. [S.l.: s.n.], 1960. v. 4, p. 96–104.
- WIDROW, B.; STEARNS, S. D. *Adaptive Signal Processing*. [S.l.]: Prentice-Hall, 1985. (Signal Processing Series).
- WIENER, N. *Extrapolation, Interpolation, and Smoothing of Stationary Time Series: With Engineering Applications*. New York: The Technology Press of the Massachusetts Institute of Technology and John Wiley & Sons, 1949.
- YOUSEF, N.; SAYED, A. A unified approach to the steady-state and tracking analyses of adaptive filters. *IEEE Transactions on Signal Processing*, v. 49, n. 2, p. 314–324, 2001.
- ZHU, H. et al. A novel robust kalman filter with unknown non-stationary heavy-tailed noise. *Automatica*, v. 127, p. 109511, 2021. ISSN 0005-1098. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0005109821000315>>.

Anexos

ANEXO A – O filtro de Kalman como estimador de parâmetros

O filtro de Kalman como estimador de parâmetros

Um dos métodos usados para estimar os parâmetros é o Filtro de Kalman. Se o vetor de parâmetros for considerado como processo de caminhada aleatória a equação de modelo de espaço de estado pode ser escrita como

$$\theta_{t+1} = \theta(t) + \nu(t) \quad (\text{A.1})$$

$$y_{t+1} = \varphi^T \theta(t) + \eta(t) \quad (\text{A.2})$$

Aqui, o vetor de estado são os vetores de parâmetros $\{W(t)\}$ e $\{e(t)\}$ são os processos de ruído branco; $E[w(t)w^T(s)] = R1(t)\delta_{ts}$, $E[e(t)e(s)] = R2(t)\delta_{ts}$, $E[w(t)e(s)] = 0$, $E[\theta(0)] = \bar{\theta}(0)$, $E[(\theta_0 - \bar{\theta}_0)(\theta_0 - \bar{\theta}_0)^T] = P(0)$ e $\theta(0)$ foram assumidos como independentes de $w(t)$ e $\nu(t)$. A matriz de transição do sistema é unitária.

O filtro de Kalman como estimador de parâmetros é apresentado em (SÖDERS-TRÖM, 1989). A estimativa dos coeficientes do filtro, nesse caso o estado $X_n = h_n$ onde observações $Y_n = y_n$ são feitas através de um vetor de medição $H = u_n$. Utilizando um filtro de resposta ao impulso finita (FIR) como sistema artificial, definido na forma a Equação ??.

Considerando o modelo de parâmetros constante

$$y_{n+1} = \varphi^T \theta + \eta_n \quad (\text{A.3})$$

descrito no espaço de estado

$$x_{n+1} = x_n \quad (\text{A.4})$$

$$y_{n+1} = \varphi^T x_n + \eta_n \quad (\text{A.5})$$

sendo x_n o vetor de estado $x_n = [a_1 \dots a_{na} b_1 \dots b_{nb}]^T = \theta$. A estimativa do estado ótimo $\hat{x}_{n+1}$ pode ser calculado em função das medidas $y_1 \dots y_n$ pelo filtro de Kalman.

Aplicando o filtro de Kalman ao modelo de estado (A.4) resulta precisamente no algoritmo mínimo quadrado (LS) básico. Uma modificação na equação (A.4) permite melhor

rastreamento dos parâmetros variante no tempo:

$$x_{n+1} = x_n + v_n \quad (\text{A.6})$$

sendo $E\{v_n v_n^T\} = R_1 \delta$

Isso implica que o vetor de parâmetro é modelado como um variável de deslocamento aleatório (*Random walk*). A matriz R_1 somada à matriz de covariância do erro do processo, P_n , pode ser usada para descrever o quanto rápido as diferentes componentes do estado $x_n = \theta^T$ espera-se variar. Aplicando o filtro de Kalman ao modelo (A.6), (A.5)

$$x_{n+1} = x_n + v_n$$

$$y_{n+1} = \phi^T x_n + e_n$$

obtém-se as seguintes equações recursivas:

$$\hat{\theta}_n = \hat{\theta}_{n-1} + K_n \varepsilon_n$$

$$\varepsilon_n = y_n - \phi^T \hat{\theta}_{n-1}$$

$$K_n = P_{n-1} \phi_n / [1 + \phi_n^T P_{n-1} \phi_n]$$

$$P_n = P_{n-1} - P_{n-1} \phi_n \phi_n^T P_{n-1} / [1 + \phi_n^T P_{n-1} \phi_n] + R_1$$

A matriz R_1 no algoritmo é uma variável de projeto definida para o equilíbrio entre o estado de alerta e a capacidade de rastrear variações temporais dos parâmetros, boas propriedades de convergência e pequenas variâncias das estimativas para sistemas invariantes no tempo. É um exemplo possível escolher que a matriz seja diagonal com diferentes elementos diagonais. Esta escolha de R_1 pode ser conveniente para descrever diferentes variações de tempo para os diferentes parâmetros.