CÁSSIO GUIMARÃES LOPES

FILTRAGEM ADAPTATIVA INCREMENTAL

Tese apresentada à Escola Politécnica da Universidade de São Paulo como requisito para obtenção do Título de Professor Livre Docente, junto ao Engenharia de Sistemas Eletrônicos.

CONSULTA
FT-3391

São Paulo
2014

CÁSSIO GUIMARÃES LOPES

FILTRAGEM ADAPTATIVA INCREMENTAL

Tese apresentada à Escola Politécnica da Universidade de São Paulo como requisito para obtenção do Título de Professor Livre Docente, junto ao Engenharia de Sistemas Eletrônicos.

Área de Concentração:
Sistemas Eletrônicos

São Paulo
2014

UNIVERSIDADE DE SÃO PAULO

Fls. 135

Rub. aca

REITORIA

Proc. USP nº 08.1.1955.3.8
Portaria nº 957
1597

O REITOR DA UNIVERSIDADE DE SÃO PAULO, usando de suas atribuições legais, **EXPEDE** a presente Portaria para declarar que tendo CÁSSIO GUIMARÃES LOPES, Professor Doutor, ref. MS-3, do QDUSP-PG, em Regime de Dedicação Integral à Docência e à Pesquisa, lotado na Escola Politécnica (Departamento de Engenharia de Sistemas Eletrônicos), obtido o título de Livre-Docente, conforme elementos constantes do Processo USP nº 08.1.1955.3.8, fica, a contar de 05/09/2014, designado para exercer a função de Professor Associado, ref. MS-5, do mesmo Quadro, Parte, Regime e Lotação, nos termos do artigo 84 do Estatuto da Universidade de São Paulo.

Reitoria da Universidade de São Paulo 23 SET. 2014

MARCO ANTONIO ZAGO
Reitor

Por Delegação do Sr. Reitor
Art. 42 do Estatuto da USP
VAHAN AGOPYAN
Vice-Reitor

PUBLICADO NO D. O.
EM 25 SET 2014
SVPUBL-01-

realizado
no período de
02 a 04 de julho de 2014

ESCOLA POLITÉCNICA DA UNIVERSIDADE DE SÃO PAULO

Informamos que o Professor Doutor Cássio Guimarães Lopes, obteve o titulo de livre docente, edital 161/2013, realizando o concurso no período de 02 a 04 de julho de 2014.

A tese será atualizada e posteriormente encaminhada à biblioteca para substituir à atual, pelo próprio professor, deixada no acervo quando foi nomeado em 25 de setembro de 2014.

Atenciosamente

Gustavo
SVORCC
915225 / 915447

Vita brevis,
ars longa,
occasio praeceps,
experimentum periculosum,
iudicium difficile.

AGRADECIMENTOS

Aos que fazem este conjunto finito e indexado de dias valer à pena.

ABSTRACT

This work proposes the concept of incremental adaptive filtering by introducing incremental structures as an alternative for combining adaptive filters (AF), which presents advantages as compared to the existing parallel counterparts in the literature. It presents a simple time-varying supervisor for the $N = 2$ AFs case to drive the LMS→LMS and RLS→LMS incremental combinations, which outperform their parallel versions in stationary scenarios. The design of a proper supervisor for incremental combinations remains an open problem, however there are two approaches that enable the use of fixed supervisors. In the first approach, the tracking performance of a family of incremental combinations of two AFs is developed for fixed supervisors, deriving analytical models that describe well simulations carried out for the LMS→LMS and RLS→LMS cases. The optimal supervisor is motivated and the universality of such combinations is shown, as well as their superiority as compared to the parallel combinations of the same AFs. The second approach, longer incremental chains ($N \gg 2$) equipped with data reuse techniques are studied. The data reuse incremental algorithm has been introduced (DR-$\{\text{LMS}\}^N$) which is shown to outperform the affine projections algorithm (APA) at lower computational complexity. In the sequel, another incremental algorithm is introduced, based on the sign-error LMS (se-LMS) operating on data sharing mode; the resulting algorithm, the $\{\text{se-LMS}\}^N$, is able to match the LMS performance at lower complexity. As an extension, the LMS filter is replaced by the $\{\text{se-LMS}\}^N$ in the (DR-$\{\text{LMS}\}^N$) incremental chain, generating a new algorithm that achieves the APA performance again at lower complexity. The underlying concept in this approach is to employ AFs combinations to implement sophisticated adaptive rules more efficiently. Finally, new directions and extensions to the current work are discussed at the end of the text.

Keywords – Adaptive Filters Combinations, Incremental Combinations, Adaptive Filtering, Data Reuse.

RESUMO

Este trabalho propõe o conceito de filtragem adaptativa incremental, introduzindo estruturas incrementais para combinações de filtros adaptativos (FA) como uma alternativa vantajosa às combinações paralelas existentes. Propõe um projeto simples de supervisor incremental variante no tempo para o caso de dois filtros, combinando LMS→LMS e RLS→LMS e mostrando a superiodade destes arranjos em comparação às suas contrapartidas paralelas em cenários estacionários. O projeto de um supervisor adequado para casos genéricos ainda é um problema em aberto; em consequência, aborda-se dois cenários em que é possível utilizar estruturas incrementais com supervisores fixos. No primeiro caso, estuda-se o desempenho de rastreamento de uma família de combinações incrementais de $N = 2$ filtros com supervisores fixos, desenvolvendo modelos analíticos que descrevem bem as simulações para os casos LMS→ LMS e RLS→ LMS. Mostra-se que é possível otimizar o supervisor, comprovando a universalidade da estrutura incremental e sua superioridade frente às estruturas paralelas. No segundo caso, estuda-se cadeias incrementais mais longas ($N \gg 2$) com a incorporação de reuso de dados. Desenvolveu-se o algoritmo LMS incremental de reuso de dados (DR-$\{\text{LMS}\}^N$) que supera em desempenho o algoritmo APA e com complexidade computacional menor. Na sequência, desenvolveu o algoritmo sign-error LMS (se-LMS) de compartilhamento de dados ($\{\text{se-LMS}\}^N$), que iguala o desempenho do algoritmo LMS, com complexidade menor. Por fim, substituiu-se o $\{\text{LMS}\}^N$ pelo $\{\text{se-LMS}\}^N$ na cadeia DR-$\{\text{LMS}\}^N$, gerando o novo algoritmo incremental DR-$\{\{\text{se-LMS}\}^N\}^L$ que iguala o desempenho do APA, com complexidade reduzida. A conceito subliminar nesta abordagem é a utilização de combinações de FAs para implementar regras adaptativas sofisticadas de maneira mais eficiente. Por último, são apresentadas perspectivas para extensões dos trabalhos conduzidos.

Palavras-Chave – Combinações de Filtros Adaptativos, Combinações Incrementais, Filtragem Adaptativa, Reuso de Dados.

SUMÁRIO

Lista de Figuras ix

1 Introdução 1

1.1 Objetivos e Contribuições 4

1.2 Organização do Trabalho 5

2 Fundamentos de Filtragem Adaptativa 7

2.1 Notação 7

2.2 O Problema de Filtragem Adaptativa 9

2.3 Soluções Iterativas 12

2.4 Soluções Adaptativas 13

2.4.1 Algoritmo LMS 15

2.4.2 Algoritmo NLMS 15

2.4.3 Algoritmo Projeções Afins (APA) 16

2.4.4 Algoritmo RLS 17

2.5 A Importância da Análise de Desempenho 18

3 Combinação Incremental de Filtros Adaptativos 21

3.1 Combinações Paralelas 22

3.2 Estruturas Incrementais 27

3.3 Capacidade de Rastreamento Incremental 31

3.3.1 Modelo de dados e conservação da energia 33

3.3.2 Relação de Variância 37

3.3.3 Rastreamento e supervisor ótimo 38

3.4 LMS→LMS 40

3.5 RLS→LMS 43

4 Reutilização de Dados em Formas Incrementais **54**

4.1 Algoritmo LMS Incremental 55

4.1.1 Técnicas de Reutilização de Dados 55

4.1.2 Reuso de Dados em Combinações de FAs 57

4.1.3 A regra LMS e reuso de dados 59

4.2 Algoritmo Sign-error LMS Incremental 63

4.2.1 Uma Implementação Eficiente do Algoritmo NLMS 66

4.2.2 Aproximações Sucessivas do Passo de Cálculo 68

4.2.3 Minimizando o número de componentes 70

4.2.4 O Algoritmo DR-$\{\text{LMS}\}^N$ via $\{\text{se-LMS}\}^N$ 70

5 Epílogo **74**

5.1 Considerações Finais 74

5.2 Perspectivas 77

Referências **78**

LISTA DE FIGURAS

1 O problema de filtragem adaptativa. 10

2 A combinação paralela de FAs independentes. 23

3 A estrutura paralela para $N = 2$ FAs. 24

4 A universalidade do algoritmo CLMS. 25

5 Implementação da realimentação cíclica na estrutura paralela. . . 26

6 Combinação CLMS com cyclic feedback (2), CLMS com transferência de pesos (3), LMS incremental (1) (3.10). Mesmos parâmetros de [1] e [2]. 27

7 A estrutura incremental para $N = 2$ FAs componentes. 29

8 Supervisores $\lambda_s(i)$ para o CLMS (3.6) e incremental (3.10) da Fig.9. 31

9 Comparação: μ_1-LMS e μ_2-LMS, LMS-LMS paralelo (CLMS), LMS→LMS incremental (3.10) e incremental simultâneo (3.11) (INC-COOP1 e INC-COOP2). 32

10 Comparação: CLMS, CLMS com $\gamma\mu_2$, e LMS→LMS incremental simultâneo (INC-COOP2); ambos com $\gamma = 0.1$ [3]. 33

11 RLS-LMS paralelo (3.8) e RLS→LMS incremental (3.12) (INC-COOP1). Parâmetros obtidos de [4]. 34

12 Superfície de desempenho LMS→LMS: EMSE em função de λ e μ_o. Caso estacionário ($\sigma_q^2 = 0$) e correlacionado com $\sigma_u^2 = 1$ e $\sigma_v^2 = 0.1$. A estrutura não apresenta universalidade. 42

13 Superfície de desempenho LMS→LMS: EMSE em função de λ e μ_o. Caso não-estacionário ($\sigma_q^2 = 10^{-6}$), correlacionado com $\sigma_u^2 = 1$ e $\sigma_v^2 = 0.1$. A estrutura é universal, com diversos λ^o não-triviais em função de μ_o. 43

14 Ábacos EMSE x λ para LMS→LMS: teoria (curva de baixo magenta), simulação (curva do meio preta) e combinação paralela (curva de cima ciano); $\mu_o = 0.3$, $r_o = 0.01$ e $\sigma_q^2 = 10^{-6}$. A estrutura incremental é universal e claramente superior à combinação paralela. 45

15 Ábacos EMSE x λ para LMS→LMS: teoria (curva de baixo magenta), simulação (curva do meio preta); a combinação paralela divergiu; $\mu_o = 0.6$, $r_o = 0.01$ e $\sigma_q^2 = 10^{-6}$. A estrutura incremental é universal e estável até $\lambda = 0.6$. A combinação paralela divergiu para todo λ. 46

16 Combinação RLS→LMS. Casos estacionário correlacionado (a) e não-estacionário branco (b). A estrutura não apresenta universalidade nestes casos. 47

17 Combinação RLS→LMS. Casos não-estacionário correlacionado. A universalidade é nítida, com o erro diminuindo conforme μ_o aumenta. 48

18 Ábacos EMSE x λ para RLS→LMS (Teoria e simulação) e combinação paralela; $\mu_o = 0.2$ e $\sigma_q^2 = 10^{-4}$. Em azul tracejado o supervisor ótimo λ^o. 49

19 Comparação para $\mu_o = 0.2$ e $\lambda = 0.4$: RLS→LMS simulado (preto) e teoria (tracejado); e combinação paralela. Rodadas 500 realizações. 50

20 Ábacos EMSE x λ para RLS→LMS (Teoria e simulação) e combinação paralela; $\mu_o = 0.4$ e $\sigma_q^2 = 10^{-4}$. 51

21 Comparação para $\mu_o = 0.4$ e $\lambda = 0.6$: RLS→LMS simulado (preto) e teoria (tracejado); e combinação paralela. Rodadas 500 realizações. 51

22 (a) Comparação para $\mu_o = 0.9$, $\lambda = 0.6$ e $\sigma_q^2 = 10^{-4}$: Os FAs individuais LMS e RLS; a estrutura RLS→LMS; a combinação paralela com λ fixo (CVX_1) e com $\lambda(i)$ adaptativo (CVX_2). Sinais correlacionados, 500 realizações. (b) Versão ampliada das curvas convergentes. 52

23 Ábacos para RLS→LMS para diversos valores de σ_q^2 e $\mu_o = 0.2$. . 53

24 Compartilhamento de dados em estruturas incrementais. 56

25 Relação entre as várias estratégias de reutilização de dados. 57

26 Reuso de dados em estruturas incrementais. 59

27 Comparação: APA versus DR-$\{LMS\}^N$. Sinais brancos em cenário estacionário, com K=10. 61

28 Comparação: APA versus DR-$\{LMS\}^N$. Sinais brancos em cenário estacionário, com K=20. 62

29 Comparação: APA versus DR-$\{LMS\}^N$ (INC-DR-LMS). Sinais correlacionados em cenário estacionário, K=10, N=20. 63

30 Comparação: APA versus DR-$\{LMS\}^N$ (INC-DR-LMS). Sinais correlacionados em cenário não-estacionário, K=10, N=20. 64

31 Comparação caso branco estacionário: se-LMS, LMS e $\{\text{se-LMS}\}^N$ em ponto fixo (16 bits). Sign-error: $\mu = 0.004$; LMS: $\mu = 0.083$; e $\{\text{se-LMS}\}^N$: $N = 9$, $\mu_n = 2^{-3-n}$. 65

32 Evolução de uma realização $\bar{\mu}(i)$ e $|e_n(i)|$ durante uma iteração i: $M = 10$, $N = 16$, e $\mu_n = 2^{-4-n}$ (Precisão de MATLAB). $|e_n(i)|$ é normalizado por $|u_i|^2$ para compatibilidade com a escala do gráfico. 69

33 Cenário não-estacionário com $M = 10$ e quantização ponto fixo (16 bits). (a) Entrada branca. **se-LMS**: $\mu = 2^{-7}$; **LMS**: $\mu = \mu^o = 0.0916$; e $\{\textbf{se-LMS}\}^N$: $N = 5$ and $\mu_n = 2^{-5-n}$. (b) Entrada correlacionada. **se-LMS**: $\mu = 2^{-6}$; **LMS**: $\mu = 0.02$; e **NLMS**: $\mu = 0.09$ e $\epsilon = 2^{-F+1}$; e $\{\textbf{se-LMS}\}^N$: $N = 5$ e $\mu_n = 2^{-5-n}$. . . . 73

34 APA e Algoritmo 2 em cenário estacionário—$M = 100$ e $K = 10$ (simulação em precisão de MATLAB devido à sensibilidade do algoritmo APA a palavras de comprimento reduzido). (a) Entrada branca. **APA**: $\mu = 0.3$ e $\epsilon = 10^{-6}$; e **DR-**$\{\{\textbf{se-LMS}\}^N\}^L$: $L = 10$, $N = 20$, e $\mu_n = 2^{-n}$. (b) Entrada Correlacionada. **APA**: $\mu = 0.06$ e $\epsilon = 10^{-6}$; e **DR-**$\{\{\textbf{se-LMS}\}^N\}^L$: $L = 30$, $N = 10$, and $\mu_n = 2^{-15-n}$. 73

1 INTRODUÇÃO

Originada no contexto de redes adaptativas e otimização distribuída [5], [6], a combinação incremental de filtros adaptativos (FA) pode apresentar desempenho global substancialmente melhor do que os FAs individuais que as compõe, se projetadas corretamente. Em cenários importantes, a pergunta que permeia todo o desenvolvimento do trabalho é:

Como aumentar o desempenho de sistemas adaptativos sem aumentar correspondentemente a complexidade computacional?

Combinações de FAs vêm se estabelecendo como uma técnica de melhoria de desempenho de sistemas adaptativos quando um único FA padrão não apresenta desempenho satisfatório [2, 7].

Nos primeiros trabalhos abordando a combinação de FAs os experts, ou os filtros adaptativos componentes, evoluiam de maneira independente, sem que o resultado de um influenciasse o comportamento dos outros [8], [4], [9]. Os mesmos dados eram apresentados a todos, que os processavam usando as suas próprias regras de adaptação com os parâmetros subjacentes. Assim, todos os FAs componentes possuiam a mesma entrada e alimentavam o mesmo supervisor, o qual se adaptava a fim de gerar uma estimativa global que superasse em desempenho os experts individuais.

Estas primeiras combinações organizavam, portanto, os FAs componentes em paralelo. Inúmeros trabalhos estão disponíveis, combinando FAs com passos de

cálculo diferentes, ordens diferentes, regras adaptativas diferentes, supervisores diferentes e mesmo abordando aplicações fora do contexto de identificação de sistemas, como predição adaptativa e rastreamento de sinais [1, 4, 7–10]. Todos eles empregam experts que operam independentes uns dos outros, configurando uma topologia ou estrutura *paralela*.

Conforme estudado em diferentes trabalhos, a estrutura paralela é capaz de atingir a universalidade em diversos cenários; contudo possui uma desvantagem intrínseca: o *efeito da estagnação*. Este efeito é resultado da natureza competitiva desta estrutura: o supervisor consulta todos os seus experts ao mesmo tempo, selecionando e ponderando aquele que apresentar melhor resultado. É mais notável em combinações simples, com poucos filtros, particularmente para dois FAs em ambientes estacionários. No caso bem estabelecido de dois FAs LMS com passos diferentes, o supervisor é capaz de entregar em sua saída a rápida convergência do filtro com maior passo e a exatidão do filtro de menor passo [3]. Assim, no transiente o FA com passo maior rapidamente reduz o erro tornando-se dominante, o que faz com que o supervisor ignore o filtro preciso, e o processamento realizado por este último é desperdiçado. Em contrapartida, no regime permanente o maior nível de ruído do FA rápido o exclui da competição, tal que o supervisor seleciona a acurácia do filtro de menor passo; o desperdício computacional agora é do FA rápido.

Técnicas ad-hoc de transferências condicionais de pesos do FA rápido para o FA preciso [4] e de realimentação cíclica da saída global [1] foram desenvolvidas; as primeiras melhorando a estagnação da combinação paralela, mas não eliminando; as últimas praticamente eliminando a estagnação.

Neste contexto, as combinações ou estruturas incrementais foram introduzidas para eliminar naturalmente a estagnação das combinações paralelas [3]. Nelas, o processamento cooperativo contrasta com o competitivo das formas paralelas. Os

FAs componentes são reorganizados em cascata, ou em série (não confudir com TANDEM), tal que a estimativa de um expert é injetado como condição inicial para o próximo filtro da cascata; o último filtro fornece sua estimativa como ponto de partida para o primeiro FA na iteração seguinte, e o conjunto segue operando desta maneira, cooperativamente [5].

As técnicas incrementais são recentes na literatura, há um vasto campo a ser explorado, uma vez que todos os trabalhos existentes sobre combinações em paralelo podem ser revisitados e seus FAs reorganizados na forma incremental. Os resultados disponíveis demonstram que as combinações incrementais apresentam vantagens consideráveis frente às suas versões paralelas; em ambientes estacionários e não-estacionários, para sinais correlacionados ou brancos [3].

É possível, ainda, incorporar de maneira natural técnicas de reuso de dados, e motiva-se a comparação direta dos algoritmos de reuso de dados incremental [11] com aqueles já bem estabelecidos, como o algoritmo de projeções afins (APA) [12], [13], [14], entre outros. Os novos algoritmos são capazes de superar o APA, e com complexidade computacional menor. Na verdade, introduz-se o conceito contra-intuitivo de combinações de FAs como técnicas de redução de complexidade.

O grande desafio da estrutura incremental reside, paradoxalmente, na sua maior vantagem; no caso paralelo, os experts evoluem de maneira independente, competindo, o que torna possível desacoplar o projeto do supervisor da adaptação dos FAs individuais. No caso incremental, com os FAs em cascata e cooperando e se influenciando diretamente, o desacoplamento não é possível, o que torna o projeto do supervisor um desafio; e ainda permanece um problema em aberto.

Contudo, este trabalho apresenta um projeto simples de supervisor incremental, que se ajusta em resposta à adaptação dos FAs, e dois cenários em que o uso de supervisores incrementais fixos podem ser explorados.

No caso não-estacionário, é possível empregar supervisores fixos; ainda assim, obter ganhos expressivos de desempenho quando comparados às formas paralelas equipadas com supervisores adaptativos [8]. É conduzido um estudo teórico de desempenho no sentido médio quadrático, gerando modelos analíticos que podem orientar o projeto dos supervisores fixos. Isto é feito para uma família de $N = 2$ FAs arranjados na forma incremental e com compartilhamento de dados. As combinações incrementais LMS→LMS e RLS→LMS são apresentadas como casos particulares e a universalidade é comprovada e explorada em situações diversas. A superioridade da forma incremental frente à paralela, ambas com supervisores fixos, é evidenciada em todos os casos estudados, mesmo quando a paralela é equipada com supervisor adaptativo.

Para cadeias incrementais maiores ($N \gg 2$), é possível empregar supervisores fixos e reuso de dados à regras da família LMS, como o próprio LMS e o sign-error LMS, formando estruturas de reuso de dados que combatem e superam outros algoritmos bem estabelecidos, como o APA e o TRUE-LMS, e com complexidade computacional reduzida.

1.1 Objetivos e Contribuições

Sumariza-se a seguir as principais diretivas deste trabalho: Motivar as estruturas incrementais (híbridas) como uma alternativa viável às estruturas paralelas.

- Incorporar o reuso de dados em combinações incrementais, aumentando drasticamente o desempenho da cadeia resultante; induzir o reuso de dados no caso paralelo;

- Motivar o uso de processamento incremental via regra LMS com vantagens em termos de desempenho médio quadrático e de complexidade computacional, ao invés de algoritmos mais complexos, como o APA;

- Motivar o processamento incremental usando a regra sign-error LMS, como substituto para o LMS padrão, com melhor desempenho e menor complexidade, tanto em ambientes estacionários, como não estacionários;

- Combinar os itens anteriores, chegando a um algoritmo que explora ostensivamente o reuso de dados via regra sign-error LMS, superando o APA em desempenho e complexidade computacional;

- Desenvolver modelos analíticos de desempenho para combinações incrementais LMS→LMS e RLS→LMS, validando os modelos via simulações; sugerir a superioridade da estrutura incremental em comparação com a paralela;

- Instigar o estudo das inúmeras variantes possíveis dos casos estudados, estendendo para regras diferentes, estruturas de filtragem alternativas, caso não-supevisionado e outros critérios de desempenho, etc.

1.2 Organização do Trabalho

O Capítulo 2 introduz conceitos e definições sobre fitragem adaptativa, necessários para dar suporte aos capítulos seguintes.

O Capítulo 3 introduz as estruturas incrementais, colocando-as em perspectiva frente às formas paralelas. Emprega-se combinações simples, de dois filtros. Motiva o uso de supervisores fixos, para o qual são desenvolvidos modelos analíticos para o desempenho médio quadrático de uma família de combinações incrementais, que confirmam o que é observado em simulações.

Como uma extensão natural, o Capítulo 4 propõe a incorporação de técnicas de reuso de dados em cadeias incrementais mais longas, motivando sua comparação com estratégias de reuso celebradas da literatura, com o APA, com grandes vantagens para as novas técnicas.

Finalmente, o trabalho é encerrado com o Capítulo 5, que discorre considerações finais, também descrevendo a pesquisa em andamento e futuras direções a serem seguidas.

2 FUNDAMENTOS DE FILTRAGEM ADAPTATIVA

2.1 Notação

Será adotada a notação da referência [14]. Há quatro tipos de variáveis neste trabalho: *escalares, vetoriais, matriciais e aleatórias.*

Conceitualmente é importante diferenciar uma variável aleatória de suas realizações (valores determinísticos), o que não é feito muitos trabalhos sobre filtros adaptativos. Assim, uma variável aleatória é representada em **negrito**, enquanto suas realizações usam fonte normal. Por exemplo, $\boldsymbol{x}$ e x. Desta forma, todas as quantidades que aparecerem ao longo do texto em **negrito**, são quantidades aleatórias ou randômicas. As quantidades em fonte normal, são determinísticas.

Adota-se LETRAS MAIÚSCULAS para matrizes e letras minúsculas para escalares e vetores, X e x, respectivamente. A exceção a esta regra está na descrição de parâmetros, como a ordem de um filtro, ou o número de realizações, ou o número de filtros usados em uma cascata. A ordem de um filtro, por exemplo, é denotada por M, enquanto o número de filtros, por N. Ainda, o número de pares de dados usados nos algoritmos de reutilização de dados é denotado por K, que também usa-se para o rank do algoritmo de projeções afins (APA). Estas exceções ficarão claras pelo contexto.

Para denotar dependência temporal de uma quantidade *escalar* usa-se parênteses. Assim, $u(i)$ é um escalar variante no tempo; é o sinal de entrada ou de refência

do FA. Para indexar tempo em vetores e matrizes usa-se índices subescritos; logo, u_i é um vetor que coleta amostras de $u(i)$. Neste trabalho, o vetor u_i, também chamado *vetor de regressão*, é um *vetor linha*, ao passo que na literatura o vetor de regressão costuma ser usado como sendo um vetor coluna.

A matriz de regressores, quando vista como quantidade aleatória, é denotada como $\boldsymbol{U}_i$, e suas realiações U_i. De forma análoga, $\boldsymbol{d}(i)$ modela o sinal desejado como uma quantidade escalar aleatória, enquanto $\boldsymbol{d}_i$ coleta amostras de $\boldsymbol{d}(i)$.

Usa-se como variáveia de indexação letras minúsculas usuais; assim tempo discreto é denotado pela letra minúscula i. O n-ésimo filtro em uma cascata denota-se por n, e o k-ésimo para de dados detona-se por k.

O símbolo u_i^* denota o transposto conjugado do vetor u_i. Se u_i for real, tem-se $u_i^* = u_i^T$.

Valores esperados são denotados por E, sem a utilização de parênteses, a menos que estritamente necessário. Assim, $E\boldsymbol{x}_i\boldsymbol{x}_i^* = E(\boldsymbol{x}_i\boldsymbol{x}_i^*)$ para algum vetor aleatório $\boldsymbol{x}_i$.

Usa-se letras gregas para representar algumas quantidades-padrão em filtragem adaptativa. São também típicamente claras pelo contexto. Em particular, λ representa um autovalor de uma matriz, mas também pode ser o parâmetro do supervisor em uma combinação convexa, por exemplo. O parâmetro de regularização é usado como ϵ, e o fator de esquecimento (Algoritmo RLS) é η. Usa-se, também, $\eta(i)$ para o MSD, $\zeta(i)$ para o EMSE e $\xi(i)$ para o MSE, como será definido na próxima seção.

Para descrição das combinações incrementais adota-se uma notação relativamente intuitiva. Para o FA RLS injetando sua estimativa no FA LMS denota-se RLS$\rightarrow$LMS. No caso paralelo, escreve-se RLS$--$LMS. Para cadeias incrementais longas, por exemplo, N filtros LMS, escreve-se $\{\text{LMS}\}^N$. Quando há reuso de

dados (Data Reuse - DR) incorporado, emprega-se DR-$\{LMS\}^N$.

Reune-se estas definições no início do trabalho para alertar o leitor; elas são repetidas ao longo do trabalho para clareza e facilidade de refência, sempre que conveniente.

2.2 O Problema de Filtragem Adaptativa

Um filtro (digital) adaptativo (FA) é um estimador automático de sinais (e sistemas) livre de modelo e que não requer conhecimento direto e exato da estatística dos sinais envolvidos. Tipicamente, trata-se de um filtro transversal com M coeficientes [1], representado por um vetor-coluna w, que é treinado por um algoritmo adaptativo.

Na abordagem supervisionada, o FA observa dois sinais, reais ou complexos: o sinal *regressor* $u(i)$, e o sinal *desejado* $d(i)$, conforme mostra a Figura 1. Na iteração i, o FA combina M amostras sucessivas do sinal regressor, capturadas em um vetor-linha de regressão $u_i = [u(i) \;\; u(i-1) \;\; \ldots \;\; u(i-M+1)]$, para gerar uma estimativa $y(i) = u_i w_{i-1}$ do sinal desejado que minimize o erro de estimação $e(i) = d(i) - u_i w_{i-1}$ de acordo com algum critério de otimização. Esta formulação permite resolver uma infinidade de problemas em diversas áreas da engenharia, como Processamento de Sinais, Telecomunicações, Controle, entre outras [14, 15].

A formulação geralmente adotada para se estudar filtros adaptativos é o de identificação de sistemas, no qual o sinal de entrada $u(i)$ alimenta o FA e a planta desconhecida. A saída da planta é medida, gerando o sinal desejado $d(i)$. O FA procura, então, determinar a resposta ao impulso truncada w que melhor representa a planta desconhecida do ponto de vista entrada-saída, minimizando o erro $e(i)$ [15].

[1] Conceitualmente é um filtro de resposta ao impulso finita – FIR. Filtros de resposta impulsiva infinita (IIR) também podem ser empregados.

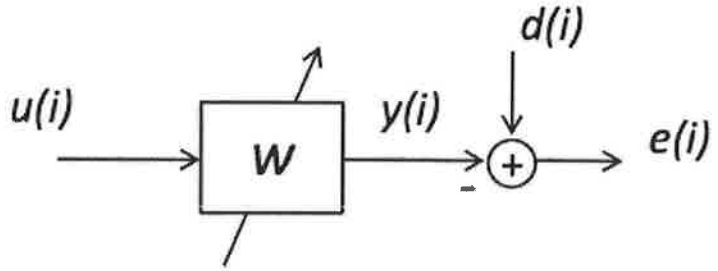


Figura 1: O problema de filtragem adaptativa.

Formalmente, o problema de filtragem adaptativa recém descrito pode ser motivado dentro do arcabouço teórico de Teoria de Estimação (Linear) Ótima no sentido Bayesiano [16]. Na abordagem Bayesiana de estimação, modela-se a quantidade a ser estimada $d(i)$ como uma quantidade aleatória $\boldsymbol{d}(i)$. Nesta formulação, tem-se acesso ao vetor-linha de regressão

$$\boldsymbol{u}_i = [\boldsymbol{u}(i)\ \boldsymbol{u}(i-1)\ \ldots\ \boldsymbol{u}(i-M+1)] \tag{2.1}$$

que é correlacionado com o sinal desejado. Desta forma, estima-se o sinal desejado via produto interno entre o vetor de regressão e o vetor de pesos do filtro w

$$\boldsymbol{d}(i) = \boldsymbol{u}_i w \tag{2.2}$$

O problema de otimização que se coloca é projetar w tal que o *erro de estimação*, agora modelado como uma quantidade aleatória

$$\boldsymbol{e}(i) = \boldsymbol{d}(i) - \boldsymbol{u}_i w \tag{2.3}$$

seja minimizado em algum sentido estatístico. Em filtragem adaptativa, adota-se o critério do erro médio quadrático, ou "*mean-square error*" (MSE). Busca-se o vetor ótimo w^o que minimiza a função custo [14], [17]

$$J(w) = \mathrm{E}\,|\boldsymbol{e}(i)|^2 \tag{2.4}$$

Expandindo-se a função acima tem-se

$$J(w) = \sigma_d^2 - R_{du}^* w - w^* R_{du} + w^* R_u w \tag{2.5}$$

em que * denota transposto conjugado e os momentos estatísticos dos dados são definidos como

$$\sigma_d^2 \triangleq \mathrm{E}\,|\boldsymbol{d}(i)|^2 \ , \qquad R_{du} \triangleq \mathrm{E}\,\boldsymbol{d}(i)\boldsymbol{u}_i^* \ , \qquad R_u \triangleq \mathrm{E}\,\boldsymbol{u}_i^* \boldsymbol{u}_i \tag{2.6}$$

A quantidade σ_d^2 é a variância do sinal desejado, o vetor (matriz) R_{du} captura a correlação cruzada entre o sinal desejado $\boldsymbol{d}(i)$ e o vetor de regressão $\boldsymbol{u}_i$, enquanto a matriz R_u é a matriz de auto-covariância do vetor de regressão. Os sinais são assumidos estacionários, portanto as matrizes são invariantes no tempo.

A função custo (2.5) é um hiperparabolóide com único ponto de mínimo em $w = w^o$ se $R_u > 0$, ou seja, se R_u for positiva definida. O vetor ótimo w^o que minimiza (2.5) satisfaz as chamadas *Equações Normais* (EN)

$$R_u w^o \ = \ R_{du} \tag{2.7}$$

As EN sempre admitem solução; a solução será única se $R_u > 0$ [17]. Se R_u for singular, então infinitas soluções podem ser determinadas, da forma

$$\bar{w}^o = w^o + c \tag{2.8}$$

em que w^o é a solução de norma mínima e c é qualquer vetor pertencente ao espaço nulo de R_u. Todas as soluções $\bar{w}^o$ levam ao mesmo erro mínimo teórico, dado por

$$J_{min} \triangleq J(w^o) = \sigma_d^2 - R_{du}^* R_u^{-1} R_{du} \tag{2.9}$$

2.3 Soluções Iterativas

Uma maneira alternativa de se determinar o vetor ótimo w^o é recorrer às soluções iterativas. Dada uma estimativa inicial (arbitrária) w_{-1}, pode-se calcular o vetor ótimo de maneira recursiva para $i \geq 0$ via

$$w_i = w_{i-1} + \mu p_i \tag{2.10}$$

em que p_i é um vetor de atualização e μ é um escalar real positivo denominado passo de cálculo, que controla a evolução do processo iterativo.

Para soluções recursivas do tipo (2.10) define-se a *curva de aprendizado*, que captura a evolução do aprendizado em função da iteração (temporal) i

$$J(i) \triangleq J(w_{i-1}) \tag{2.11}$$

Em filtragem adaptativa, é comum utilizar o gradiente da funcão custo para orientar o processo de otimização, ou seja

$$p_i = -B\nabla_w^* J(w_{i-1}) \tag{2.12}$$

onde $B > 0$ é uma matriz positiva definida que determina o tipo de algoritmo iterativo. Para o critério do erro médio quadrático, o gradiente da função custo depende somente das matrizes R_{du} e R_u, sendo dado por

$$\nabla_w J = -R_{du}^* + w^* R_u \tag{2.13}$$

Há duas escolhas típicas para a matriz B. Para $B = I$ (matriz identidade), gera-se o *algoritmo do gradiente descendente*

$$w_i = w_{i-1} + \mu(R_{du} - R_u w_{i-1}) \tag{2.14}$$

Para $B = [\nabla^2 J(w)]^{-1}$ (Inverso da Hessiana) gera-se o *algoritmo de Newton*. Pela

equação do gradiente (2.13), calcula-se a Hessiana

$$\left[\nabla^2 J(w)\right] = R_u \tag{2.15}$$

que leva ao algoritmo de Newton, regularizado por um número pequeno positivo, o fator de regularização $\epsilon > 0$

$$w_i = w_{i-1} + \mu(\epsilon I + R_u)^{-1}(R_{du} - R_u w_{i-1}) \tag{2.16}$$

O que se deseja de algoritmos recursivos é que convirjam para o vetor ótimo, ou que sejam *estáveis*

$$w_i \stackrel{i\to\infty}{\longrightarrow} w^o \tag{2.17}$$

e que o erro $J(i)$ diminua (monotonicamente) até o valor mínimo predito pela teoria (2.9)

$$J(i) \stackrel{i\to\infty}{\longrightarrow} J_{min} \tag{2.18}$$

Para isto é preciso garantir a estabilidade das implementações recursivas. Nesse sentido, o valor do passo μ e a escolha da matriz B governam a evolução dos algoritmos, tendo impacto direto sobre a convergência e a estabilidade dos mesmos. Garante-se a estabilidade para os algoritmos (2.14) e (2.16) quando o passo μ está nas faixas, respectivamente

$$0 < \mu < \frac{2}{\lambda_{max}} \qquad \text{e} \qquad 0 < \mu < 2 \tag{2.19}$$

em que λ_{max} é o máximo *autovalor* da matriz R_u.

2.4 Soluções Adaptativas

Raramente se conhece de antemão os momentos estatísticos dos dados, em específico as matrizes (vetores) R_{du} e R_u. Em situações reais, usam-se apro-

ximações instantâneas para tais matrizes que dependem diretamente dos dados disponíveis: o sinal de referência $u(i)$ e o sinal desejado $d(i)$. Quando se substitui as quantidades estatísticas exatas (R_u e R_{du}) presentes nos algoritmos iterativos (2.14) e (2.16) por aproximações que dependem dos dados, os algoritmos iterativos tornam-se *adaptativos*.

Diferentes aproximações para ∇J e para $\nabla^2 J$ refletem-se, também, em aproximações diferentes para R_u e R_{du}. Diferentes escolhas para estas quantidades geram uma miríade de algoritmos adaptativos diferentes, com propriedades e características intrínsecas.

Estas aproximações tornam os algoritmos iterativos implementáveis em qualquer situação prática, apesar do desconhecimento das quantidades estatísticas requeridas. Contudo, as aproximações fazem com que o comportamento dos algoritmos adaptativos resultantes se desvie daquele previsto pela teoria de estimação; em particular, o comportamento médio quadrático irá desviar da curva de aprendizado $J(i)$, com taxas de convergência diferentes e excedendo o erro mínimo J_{min}, ou seja, os algoritmos apresentam um *erro em excesso*.

Uma vez que se garanta a estabilidade, algoritmos adaptativos diferentes levam o filtro a convergir com taxas diferentes e estabilizar em patamares de erro diferentes em regime estacionário. Os principais fatores que afetam o aprendizado do filtro, uma vez selecionado um algoritmo adaptativo, são o nível de correlacionamento dos sinais (principalmente o sinal de entrada $u(i)$), o grau de não-estacionaridade do ambiente (quando $w^o \rightarrow w_i^o$) e a razão sinal-ruído ("Signal-to-noise Ratio" – SNR).

Vários algoritmos adaptativos foram desenvolvidos ao longo dos anos. A seguir descreve-se alguns dos mais utilizados.

2.4.1 Algoritmo LMS

O algoritmo mais celebrado e estudado na literatura é o algoritmo Least Mean-Square (LMS), também referido como algoritmo do gradiente estocástico. Adotando-se as aproximações [14]

$$R_u \approx u_i^* u_i \quad \text{e} \quad R_{du} \approx d(i) u_i^*$$

e substituindo-as em (2.14) resulta em

$$w_i = w_{i-1} + \mu u_i^* (d(i) - u_i w_{i-1}) \tag{2.20}$$

Trata-se de um algoritmo bastante robusto, assim como a sua variante normalizada (NLMS) [18]. Uma vez escolhida a ordem do filtro, resta projetar o passo de cálculo μ, que governa o aprendizado do algoritmo. Para estabilidade e para otimizar a taxa de convergência, uma boa estratégia de projeto é [5]

$$\mu = \frac{\mu_0}{\sigma_u^2 M}, \qquad 0 < \mu_0 \leq 1 \tag{2.21}$$

para $\mu_0 \to 1$ o filtro tende a convergir mais rápido, porém com maior erro em excesso em regime. Para $\mu_0 \to 0$ o filtro converge mais lentamente, porém estabiliza em patamares de erro em excesso menores.

2.4.2 Algoritmo NLMS

As aproximações para R_u e R_{du} para o algoritmo LMS normalizado são as mesmas que aquelas do LMS; contudo, o algoritmo iterativo base é o algoritmo de Newton; assim, substituindo as aproximações e aplicando o Lema de inversão de matrizes [14], a recursão fica

$$w_i = w_{i-1} + \frac{\mu_0}{\epsilon + \|u_i\|^2} u_i^* (d(i) - u_i w_{i-1}) \tag{2.22}$$

O algoritmo NLMS comporta-se melhor que o LMS para sinais correlacionados, sendo menos sensível, também, a variações da potência do sinal de entrada $u(i)$ devido à normalização regularizada $\epsilon + \|u_i\|^2$.

Note que μ_0 é similar àquele empregado no projeto do passo do LMS; isto permite uma comparação razoável entre os dois algoritmos em termos de convergência, pelo menos quando o sinal de entrada é descorrelacionado.

2.4.3 Algoritmo Projeções Afins (APA)

O algoritmo de projeções afins ("Affine Projections Algorithm"–APA) emprega uma massa de dados maior que os algoritmos LMS e NLMS para realizar uma iteração completa, por isso constitui o que na literatura se denomina de algoritmo de *reutilização de dados* [19], [20], [13,21–23]. Ele emprega K consecutivos pares de dados $\{d(i), u_i\}$ a cada iteração, coletados no vetor coluna $K \times 1$ e na matriz $K \times M$ de regressão

$$d_i = col\{d(i), d(i-1), \ldots, d(i-K+1)\}, \quad U_i = col\{u_i, u_{i-1}, \ldots, u_{i-K+1}\}$$

Usando-se as aproximações

$$R_u \approx \frac{1}{K} U_i^* U_i \quad \text{e} \quad R_{du} \approx \frac{1}{K} U_i^* d_i \tag{2.23}$$

e invocando o Lema de inversão de matrizes chega-se à recursão do APA

$$w_i = w_{i-1} + \mu_0 U_i^* (\epsilon I + U_i U_i^*)^{-1} (d_i - U_i w_{i-1}) \tag{2.24}$$

em que a inversão é operada sobre uma matriz $K \times K$, o que é conveniente, uma vez que geralmente $K \leq M$.

A equação do APA opera com um vetor de erros de estimação $e_i = d_i - U_i w_{i-1}$, ao invés do costumeiro erro escalar $e(i)$. Ainda, é importante notar que para $K = 1$ recupera-se o algoritmo NLMS.

2.4.4 Algoritmo RLS

O algoritmo dos Mínimos Quadrados Recursivo, ou "Recursive Least Squares" (RLS) pode ser motivado como a solução exata de um problema de mínimos quadrados determinístico, ou como a solução aproximada (Estocástica) de uma instância do algoritmo de Newton. A teoria dos mínimos quadrados é rica e confere grande estrutura ao algoritmo RLS assim motivado, fugindo do escopo deste trabalho. Aqui, aborda-se a formulação estocástica do RLS.

O algoritmo RLS é obtido a partir da recursão de Newton usando a mesma aproximação para o gradiente que o LMS e o NLMS [14]

$$(R_{du} - R_u w_{i-1}) \approx u_i^*(d(i) - u_i w_{i-1}) \tag{2.25}$$

A matriz R_u é aproximada por

$$R_u \approx \frac{1}{i+1} \sum_{j=0}^{i} \eta^{i-j} u_j^* u_j^* \tag{2.26}$$

em que $0 \ll \eta < 1$ é o fator de esquecimento e ajusta a janela de observação, a qual pode ser interpretada como a memória do algoritmo.

No algoritmo RLS, ao invés de calcular a aproximação para R_u acima e invertê-la, como requer a recursão de Newton, define-se uma matriz extra P_i que calcula diretamente a inversa requerida de maneira recursiva (para detalhes, ver [14]). Após algumas manipulações, determina-se as recursões que implementam o algoritmo RLS

$$P_i = \eta^{-1}\left(P_{i-1} - \frac{\eta^{-1} P_{i-1} u_i^* u_i P_{i-1}}{1 + \eta^{-1} u_i P_{i-1} u_i^*}\right) \tag{2.27}$$

$$w_i = w_{i-1} + P_i u_i^*(d(i) - u_i w_{i-1}) \tag{2.28}$$

com $P_{-1} = \epsilon^{-1} I$. Para ambientes estacionários, escolhe-se $\eta \to 1$. Para o caso não-estacionário, é costumeiro escolher $\eta \in [0.95, 0.99]$.

O algoritmo RLS apresenta taxas de convergência elevadas, se comparado aos algoritmos anteriores. Exceto, possivelmente, no caso não-estacionário, no qual a presença de memória pode prejudicar sua capacidade de rastreamento.

2.5 A Importância da Análise de Desempenho

Observando as equações dos algoritmos adaptativos apresentados, percebe-se que FAs são sistemas variantes no tempo, não-lineares e que observam sinais $d(i)$ e $u(i)$ costumeiramente modelados como processos estocásticos. Para operação adequada é necessário assegurar sua convergência para níveis de erro compatíveis com a aplicação visada. Devido às aproximações adotadas na implementação adaptativa dos algoritmos iterativos originais, a curva de aprendizado $J(i)$ não descreve corretamente o comportamento dos FAs. É necessário desenvolver um modelo mais elaborado. Isto é feito através da *análise de desempenho* dos sistemas adaptativos.

O estudo é feito derivando-se modelos matemáticos que traduzem a evolução estatística de certas figuras de mérito que capturam o desempenho do sistema em termos dos parâmetros do filtro (ex: ordem M), do algoritmo (ex: passo μ) e das propriedades estatísticas dos sinais envolvidos.

Tipicamente, adota-se como medida de desempenho o erro médio quadrático (Mean-Square Error - MSE), o desvio médio quadrático (Mean-Square Deviation-MSD) e o erro médio quadrático em excesso (Excess Mean-Square Error-EMSE) [?]. Tais métricas são definidas como

$$MSD(i) \triangleq \mathrm{E}\,\|w^o - \boldsymbol{w}_{i-1}\|^2 \tag{2.29}$$

$$EMSE(i) \triangleq \mathrm{E}\,|\boldsymbol{u}_i(w^o - \boldsymbol{w}_{i-1})|^2 \tag{2.30}$$

$$MSE(i) \triangleq \mathrm{E}\,|\boldsymbol{e}(i)|^2 \tag{2.31}$$

O MSD mede o quanto a solução entregue pelo filtro $\boldsymbol{w}_{i-1}$ desvia do valor ótimo w^o, o EMSE mede o quanto a estimativa do filtro excede a estimativa ótima $u_i w^o$ teórica, e o MSE mede o quanto o filtro é capaz de aproximar o sinal desejado $d(i)$, e pode ser medido em sua saída, ao contrário do MSD e do EMSE que são medidas conceituais, não podendo ser medidas em situações práticas.

A análise de desempenho é uma tarefa bastante complicada, dada a natureza da lei que governa os FAs, porém é fundamental para avaliar questões básicas [15]:

- O filtro irá convergir?
- O nível de erro em regime está dentro de limites tolerados?
- Como projetar o sistema para garantir tais limites?
- Como o filtro se comporta em ambientes não-estacionários?

Na literatura, o estudo do desempenho costuma ser agrupado de acordo com o regime de operação do FA: análise do transiente, análise em regime permanente e análise de rastreamento. Para um estudo mais exato [15], [24], [25], algoritmos diferentes normalmente requerem abordagens diferentes para cada regime, e para cada regra de aprendizado diferente. Contudo, existem técnicas que permitem, em certo sentido, que a análise seja unificada, mesmo para algoritmos diferentes [14], [26]. Elas são conhecidas como relações de conservação da energia ("energy conservation relations"-ECR), e serão adotadas na análise de desempenho.

Neste trabalho (Ver Capítulo 3), serão desenvolvidos modelos para o EMSE e o MSE em regime estacionário, quando o filtro já convergiu

$$EMSE \triangleq \lim_{i\to\infty} \mathrm{E}\,|\boldsymbol{u}_i(w^o - \boldsymbol{w}_{i-1})|^2 \tag{2.32}$$

$$MSE \triangleq \lim_{i\to\infty} \mathrm{E}\,|\boldsymbol{e}(i)|^2 \tag{2.33}$$

No Capítulo 4 diferentes estruturas de filtragem serão avaliadas, observando o transiente e a convergência do filtro, porém sem derivar modelos analíticos. Isto

será feito através da evolução média quadrática *aproximada* dos filtros, estimada "empiricamente" através de promediações de vários experimentos ou realizações dos processos estocásticos envolvidos na evolução do filtro

$$MSD(i) \approx \frac{1}{R}\sum_{r=1}^{R} \|w^o - w_{i-1}^{(r)}\|^2 \tag{2.34}$$

$$EMSE(i) \approx \frac{1}{R}\sum_{r=1}^{R} |u_i^{(r)}(w^o - w_{i-1}^{(r)})|^2 \tag{2.35}$$

$$MSE(i) \approx \frac{1}{R}\sum_{r=1}^{R} |d^{(r)}(i) - u_i^{(r)} w_{i-1}^{(r)}|^2 \tag{2.36}$$

em que (r) indexa a realização do processo estocástico.

3 COMBINAÇÃO INCREMENTAL DE FILTROS ADAPTATIVOS

Combinações de FAs vêm se estabelecendo como uma técnica de melhoria de desempenho de sistemas adaptativos quando um único FA padrão não apresenta desempenho satisfatório. Tais arranjos de filtros envolvem a escolha dos FAs componentes, o tipo de supervisor que governa a combinação e a topologia em que os FAs são organizados e apresentados ao supervisor.

As combinações de FAs podem ser interpretadas como um instanciamento do problema de estimação universal [27]. Em estimação universal, um conjunto de experts apresenta suas estimativas para um supervisor, que as agrega ponderando seus desempenhos individuais e gera uma estimativa global. Esta estimativa deve ser tão boa quanto a melhor estimativa individual apresentada pelos experts, de acordo com o critério de desempenho adotado.

No contexto de filtragem adaptativa, os experts são implementados por FAs, e o supervisor é representado por um conjunto de pesos, tal que a estimativa global é gerada somando-se as estimativas individuais dos FAs ponderadas pelos pesos do supervisor. O critério adotado é costumeiramente o erro médio quadrático MSE (Mean-Square Error).

Na literatura, a grande maioria dos trabalhos emprega a estrutura paralela [1, 4, 7–10], [28–31], em que os FAs evoluem independentemente. Recentemente, foi introduzido o conceito de combinações incrementais, em que os FAs

são estruturados em cascata, operando de forma cooperativa. Esta nova topologia apresenta características próprias, podendo apresentar larga vantagem em relação às formas paralelas, se bem projetada.

3.1 Combinações Paralelas

Tipicamente, os FAs são organizados em uma topologia *paralela*, na qual evoluem de maneira independente e apresentam suas estimativas ao supervisor, que também se adapta na tentativa de gerar uma estimativa global universal, em geral no sentido médio quadrático. A Fig. 2 ilustra o arranjo.

Muitas combinações paralelas da literatura, em termos topológicos, podem ser colocadas em uma forma genérica [1,4,7,9,10], [30,31], na qual o n-ésimo FA componente é dado por

$$w_{n,i} = w_{n,i-1} + \mu_n p_{n,i} \tag{3.1}$$

onde $w_{n,i}$ é um vetor coluna $M_n \times 1$ de coeficientes e $p_{n,i} = -B_n[\nabla J_n(w_{n,i-1})]^*$ captura a regra de aprendizado empregada pelo filtro componente n. Quando os FAs componentes possuirem ordens diferentes, os vetores $w_{n,i}$ são completados com zeros para compatibilizar as dimensões. A saída ou filtro global w_{i-1} é obtida ponderando as estimativas individuais

$$w_{i-1} = \sum_{n=1}^{N} \lambda_n(i) w_{n,i-1} \tag{3.2}$$

em que o conjunto $\{\lambda_n(i)\}$ representa a ação do supervisor na iteração i. É comum empregar supervisores afins [9] e convexos [7,8]

$$\textstyle\sum_{n=1}^{N} \lambda_n(i) = 1 \qquad \text{Afim} \tag{3.3}$$

$$\textstyle\sum_{n=1}^{N} \lambda_n(i) = 1 \quad , \qquad 0 < \lambda_n(i) < 1 \qquad \text{Convexo} \tag{3.4}$$

projetando-se o supervisor, representado pelo conjunto de parâmetros $\{\lambda_n(i)\}$, de

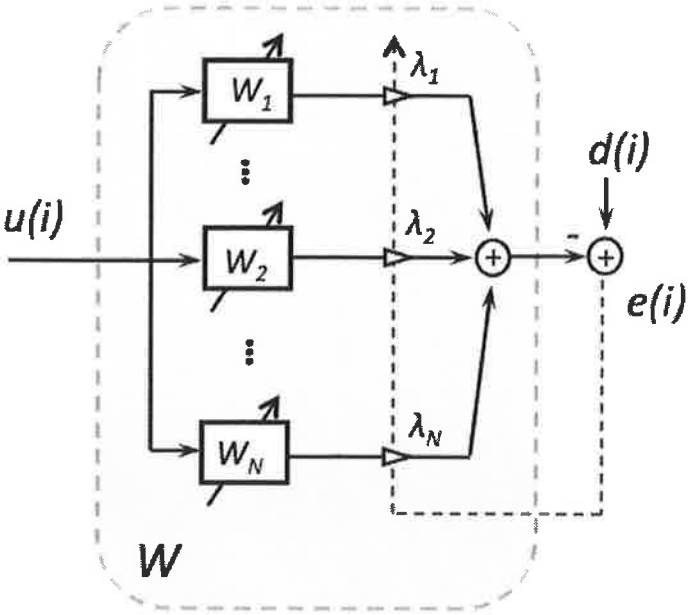


Figura 2: A combinação paralela de FAs independentes.

modo a minimizar o erro global $e(i) = d(i) - u_i w_{i-1}$ no sentido médio quadrático. O uso de uma função de *ativação* auxilia na imposição das restrições de convexidade e afinidade [7]. Desta forma, faz-se $\lambda_n(i) = g(a_n(i))$, e adota-se a estratégia do gradiente descendente para treinar os parâmetros $a_n(i)$

$$a_n(i+1) = a_n(i) - \mu_{a,n} \nabla_a e^2(i) \tag{3.5}$$

Um dos trabalhos pioneiros sobre combinações de FAs seguido as linhas acima descritas foi o algoritmo CLMS ("Convex LMS") [8], como ilustra a Fig. 3, em que dois FAs LMS de mesma ordem, mas com passos de cálculo distintos μ_1–LMS e μ_2–LMS, em que $\mu_1 > \mu_2$ são agregados por um supervisor adaptativo. Desde sua publicação, inúmeros trabalhos combinando FAs com ordens, regras adaptativas e regras de supervisão diferentes vem sendo propostos no intuito de melhorar a taxa de convergência de FAs individuais em ambientes estacionários, e de aumentar a capacidade de rastreamento em ambientes não-estacionários.

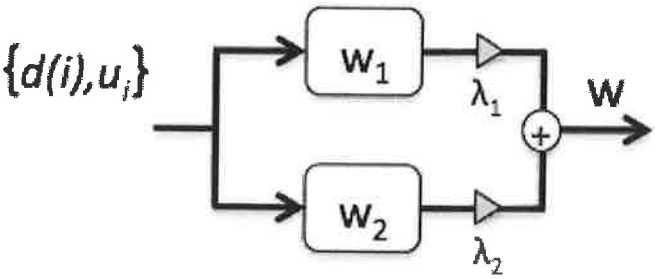


Figura 3: A estrutura paralela para $N = 2$ FAs.

Outro trabalho pioneiro, que generalizou o algoritmo CLMS em vários aspectos importantes e utilizou a combinação paralela em predição adaptativa e rastreamento de frequência foi [7].

Na sua forma simples, o CLMS é implementado via

$$\begin{aligned}
\lambda(i) &= \frac{1}{1+e^{-a(i-1)}} \\
w_{i-1} &= \lambda(i)w_{1,i-1} + (1-\lambda(i))w_{2,i-1} \\
e(i) &= d(i) - u_i w_{i-1} \\
a(i) &= a(i-i) + \mu_a e(i) u_i (w_{1,i-1} - w_{2,i-1})\lambda(i)(1-\lambda(i)) \\
w_{n,i} &= w_{n,i-1} + \mu_n u_i^*(d(i) - u_i w_{n,i-1}) , \qquad = 1,2
\end{aligned} \tag{3.6}$$

sendo capaz de alcançar o erro em regime do FA exato (μ_2-LMS) de maneira rápida, conforme ilustra a Fig. 4. Apesar da universalidade, é possível notar o *fenômeno de estagnação*, que é uma limitação intrínseca da estrutura paralela.

Durante o transiente, o filtro mais preciso (μ_2-LMS) demora a convergir e é desconsiderado pelo supervisor; no regime permanente o filtro rápido (μ_1-LMS) é excluído pelo supervisor devido ao seu maior erro em excesso. O transiente é governado pelo filtro rápido e o regime permanente, pelo filtro preciso. Há sempre um filtro usando recursos computacionais que não são utilizados para compor a saída global.

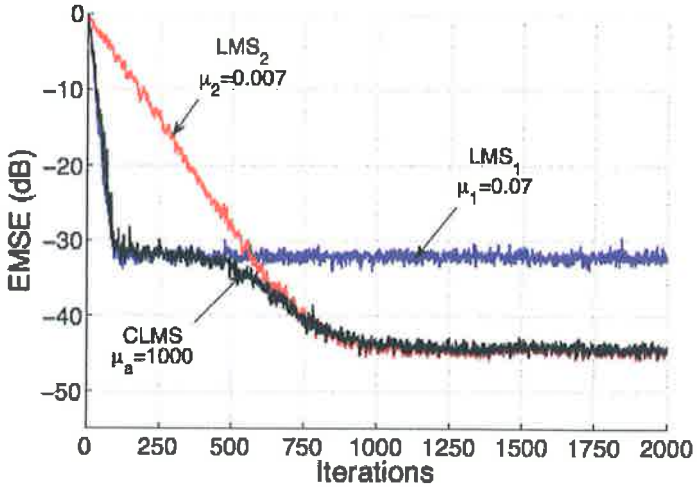


Figura 4: A universalidade do algoritmo CLMS.

O fenômeno da estagnação, acima descrito, pode ser amenizado (parcialmente) implementando transferências condicionais de coeficientes do FA rápido para o FA exato [2] [8], [4]. Isto pode ser entendido como uma estratégia ad-hoc de cooperação entre os FAs componentes em uma estrutura que é essencialmente competitiva.

Contudo, a estrutura paralela, com ou sem tra[illegible]ência de coeficientes entre os FAs, opera em modo feedforward, não há feedback; portanto, [illegible] há efeito de regulação: os FAs componentes não se beneficiam da operação do supe[illegible]sor. Se algum dos FAs componentes operar de maneira precária, ou mesmo se instabilizar, não é possível ajustar a sua operação. Neste sentido, a técnica de realimentação cíclica, ou "cyclic feedback", proposta recentemente [1], contorna naturalmente o problema da estagnação, podendo também auxiliar na melhoria de desempenho dos FAs individuais. Conforme ilustra a Fig. 5, realimenta-se para cada FA componente, a cada L iterações, a saída supervisionada global w_{i-1}, que representa a melhor estimativa disponível; o efeito líquido, no caso estacionário, é a aceleração do FA exato. É implementada adicionando uma única equação ao

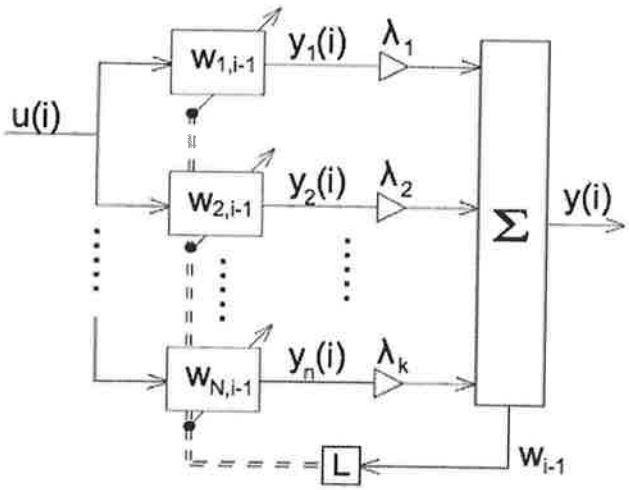


Figura 5: Implementação da realimentação cíclica na estrutura paralela.

algoritmo CLMS (3.6), resultando em

$$
\begin{aligned}
\lambda(i) &= \frac{1}{1+e^{-a(i-1)}} \\
w_{i-1} &= \lambda(i)w_{1,i-1} + (1-\lambda(i))w_{2,i-1} \\
e(i) &= d(i) - u_i w_{i-1} \\
a(i) &= a(i-i) + \mu_a e(i) u_i (w_{1,i-1} - w_{2,i-1})(\lambda(i)(1-\lambda(i)) \\
w_{n,i-1} &= \delta_L w_{i-1} + (1-\delta_L) w_{n,i-1} \\
w_{n,i} &= w_{n,i-1} + \mu_n u_i^* (d(i) - u_i w_{n,i-1}) \ , \quad n = 1,2
\end{aligned}
\tag{3.7}
$$

em que $\delta_L = i \mod L$ é o delta de Kronecker. A Fig. 6 ilustra o efeito do cyclic feedback na combinação paralela. Note que a realimentação cíclica é mais efetiva no combate da estagnação do que a transferência de coeficientes. Contudo, ambas são incapazes de superar a estrutura incremental, que será abordada em detalhe na próxima seção.

Uma variante do algoritmo CLMS combina regras adaptativas híbridas em paralelo [4], em particular os FAs RLS e LMS. O algoritmo foi proposto na tentativa de explorar a velocidade de convergência do RLS, com a janela instantânea

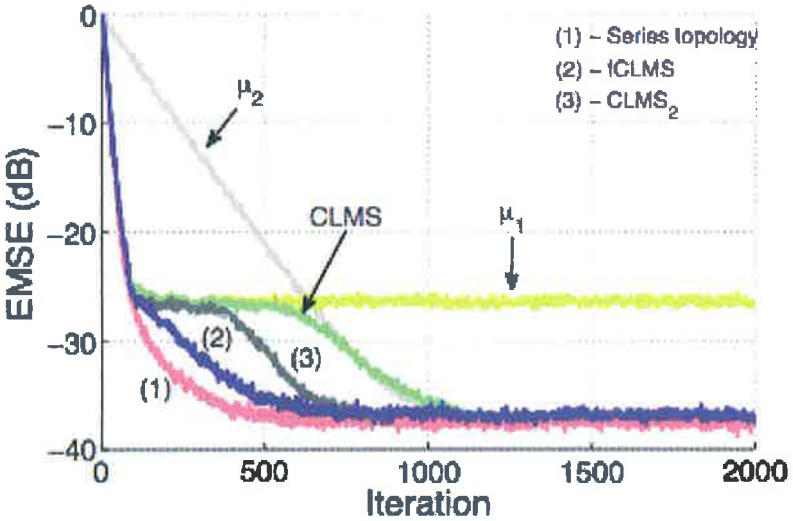


Figura 6: Combinação CLMS com cyclic feedback (2), CLMS com transferência de pesos (3), LMS incremental (1) (3.10). Mesmos parâmetros de [1] e [2].

de observação do LMS, o que é útil para certos cenários de rastreamento

$$
\begin{aligned}
\lambda(i) &= \frac{1}{1+e^{-a(i-1)}} \\
e(i) &= d(i) - u_i w_{i-1} \\
P_i &= \eta^{-1}\left[P_{i-1} - \frac{\eta^{-1}P_{i-1}u_i^* u_i P_{i-1}}{1+\eta^{-1}u_i P_{i-1} u_i^*}\right] \quad , \quad P_{-1} = \epsilon^{-1} I \\
w_{1,i} &= w_{1,i-1} + \lambda P_i u_i^*(d(i) - u_i w_{1,i-1}) \\
w_{2,i} &= w_{2,i} + (1-\lambda)\mu u_i^*(d(i) - u_i w_{2,i}) \\
a(i) &= a(i-1) + \mu_a e(i) u_i (w_{1,} - w_{2,i-1})\lambda(i)(1-\lambda(i))
\end{aligned}
\tag{3.8}
$$

A Fig. 11 mostra um exemplo da combinação paralela para o algoritmo 3.8.

3.2 Estruturas Incrementais

Historicamente, a estrutura incremental foi proposta como um rearranjo topológico da forma paralela para combater o efeito da estagnação do algoritmo CLMS. Nas combinações paralelas, o melhor que o supervisor pode fazer é apresentar como estimativa global o melhor dos mundos: selecionar o transiente do

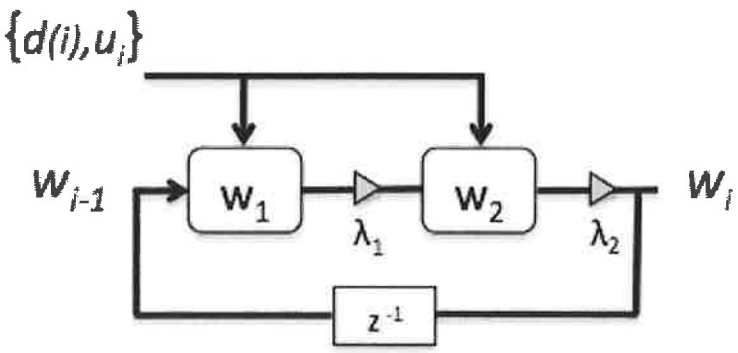


Figura 7: A estrutura incremental para $N = 2$ FAs componentes.

Apesar da vantagem estrutural cooperativa, o projeto do supervisor no caso incremental é mais envolvente, já que os FAs cooperam diretamente. No caso $N = 2$ filtros, em cenários estacionários, um projeto funcional é fazer o supervisor ser guiado pelo erro médio quadrático. Isto pode ser obtido filtrando-se o erro quadrático instantâneo, via $a(i) = \alpha a(i-1) + \beta e^2(i)$ para $0 < \alpha < 1$ e $\beta = 1 - \alpha$ para SNR alta e $\beta = 0.1 \cdot (1 - \alpha)$ para cenários ruidosos. O resultado é mapeado no intervalo $[0, 1]$ via alguma função de mapeamento, por exemplo, a função tangente hiperbólica $\lambda(i) = \tanh(a(i)/2)$.

Adotando-se esse supervisor, tem-se a contrapartida incremental LMS→LMS do algoritmo CLMS (3.6) [3]

$$\begin{aligned} e(i) &= d(i) - u_i w_{i-i} \\ a(i) &= \alpha a(i-1) + \beta e^2(i) \\ \lambda(i) &= \tanh(a(i)/2) \\ w_{1,i} &= w_{i-1} + \lambda(i)\mu_1 u_i^*(d(i) - u_i w_{i-1}) \\ w_i &= w_{1,i} + (1 - \lambda(i))\mu_2 u_i^*(d(i) - u_i w_{1,i}) \end{aligned} \tag{3.10}$$

Uma vantagem da estrutura incremental é a *operação simultânea* dos FAs, sem a restrição de convexidade. Não há, de fato, razão aparente para impor con-

vexidade sobre o conjunto de coeficientes do supervisor; uma análise parcial, sob estudo, confirma isto. Neste arranjo, os dois FAs operam durante o transiente; conforme a o processo adaptativo avança, o supervisor concentra-se progressivamente no FA exato, que opera com um fator de contração do passo γ, que permite diminuir o erro em regime, sem afetar a convergência. Um projeto funcional neste sentido é uma variante do projeto convexo anterior, rearranjando o mesmo supervisor

$$\begin{aligned} w_{1,i} &= w_{i-1} + \lambda(i)\mu_1 u_i^*(d(i) - u_i w_{i-1}) \\ w_i &= w_{1,i} + \gamma\left[\lambda(i)\mu_1 + (1-\lambda(i))\mu_2\right] u_i^*(d(i) - u_i w_{1,i}) \end{aligned} \tag{3.11}$$

As Figs. 9 e 10 mostram a operação desta versão simultânea (3.11) do arranjo incremental LMS→LMS. Na Fig. 9 nota-se que a variante simultânea consegue convergir com a mesma taxa que o arranjo incremental (3.10), mas atingindo nível de erro 10dB abaixo dos outros sistemas. A Fig. 10 repete o exemplo da Fig. 9 com o CLMS empregando o LMS mais lento com passo $\gamma\mu_2$, o que o permite alcançar o mesmo erro do incremental simultâneo, contudo com tempo de convergência expressivamente maior.

A estrutura incremental também contempla arranjos híbridos, com regras adaptativas diferentes. Assim, a versão incremental RLS→LMS da combinação paralela híbrida RLS–LMS (3.8) é dada por

$$\begin{aligned} e(i) &= d(i) - u_i w_{i-i} \\ a(i) &= \alpha a(i-1) + \beta e^2(i) \\ \lambda(i) &= \tanh(a(i)/2) \\ P_i &= \eta^{-1}\left[P_{i-1} - \frac{\eta^{-1}P_{i-1}u_i^* u_i P_{i-1}}{1+\eta^{-1}u_i P_{i-1} u_i^*}\right], \quad P_{-1} = \epsilon^{-1} I \\ w_{1,i} &= w_{i-1} + \lambda(i) P_i u_i^*(d(i) - u_i w_{i-1}) \\ w_i &= w_{1,i} + (1-\lambda(i))\mu_2 u_i^*(d(i) - u_i w_{i-1}) \end{aligned} \tag{3.12}$$

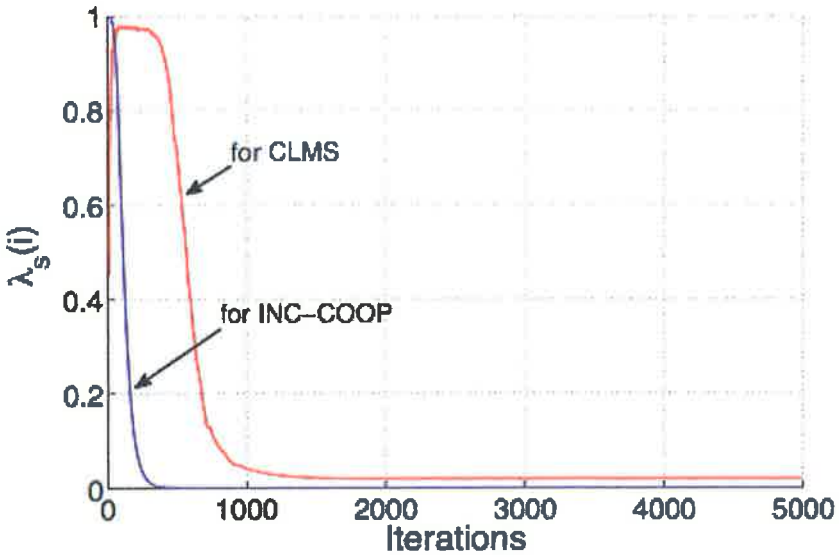


Figura 8: Supervisores $\lambda_s(i)$ para o CLMS (3.6) e incremental (3.10) da Fig.9.

A Fig. 11 mostra a comparação das combinações paralela e incremental dos FAs RLS e LMS. A superioridade da estrutura incremental é clara.

Derivou-se um algoritmo gradiente-descendente para adaptar o supervisor incremental $\lambda(i)$, ao invés de empregar o erro filtrado. Contudo, esta abordagem carece de mais investigação, ainda não tendo retornado resultados satisfatórios.

Um ponto importante das estruturas incrementais é que a associação dos FAs componentes *não é comutativa*, ou seja, o algoritmo RLS→LMS não é exatamente igual ao algoritmo LMS→RLS. Isto ainda não foi estudado, contudo as simulações não demostram grandes diferenças em relação à ordem dos FAs na cascata.

3.3 Capacidade de Rastreamento Incremental

Apesar do bom desempenho do supervisor incremental filtrado e do mesmo apresentar bons resultados para SNR diversas [3], trata-se de um projeto ad-hoc, e sua sintonia no caso genérico não-estacionário, com sinais não necessariamente gaussianos, pode ser difícil, ou mesmo apresentar desempenho insatisfatório.

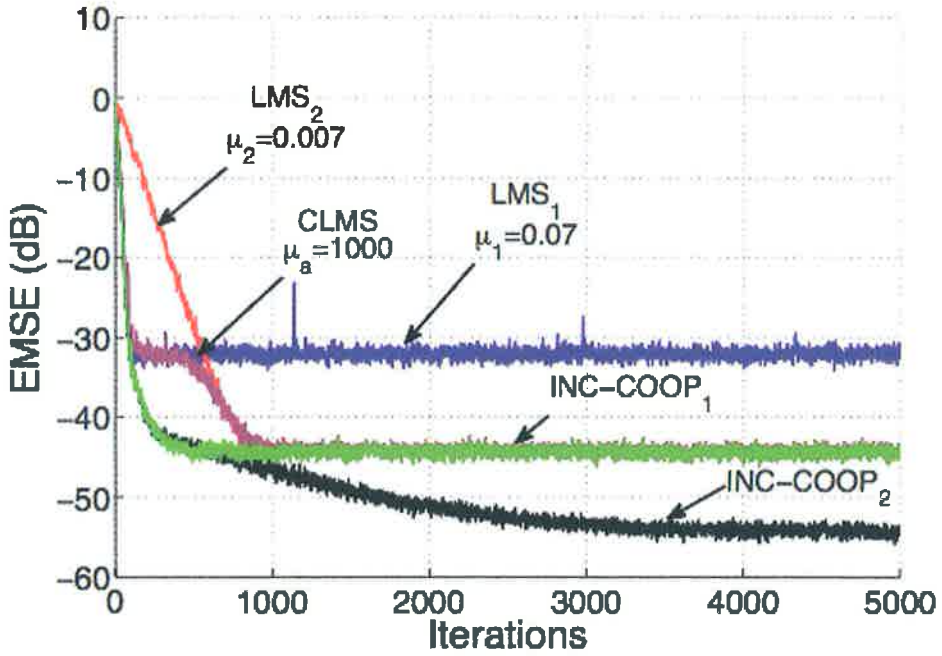


Figura 9: Comparação: μ_1-LMS e μ_2-LMS, LMS-LMS paralelo (CLMS), LMS→LMS incremental (3.10) e incremental simultâneo (3.11) (INC-COOP1 e INC-COOP2).

O desafio de projetar tal supervisor adaptativo advém, principalmente, do acoplamento explícito entre os FAs componentes no caso incremental; não é possível desacoplar o projeto do supervisor adaptativo da adaptação dos FAs componentes, conforme ocorria na estrutura paralela. Assim, a cooperação explícita, embora estruturalmente vantajosa, dificulta o projeto do supervisor.

Como, então, utilizar a estrutura incremental de maneira eficiente, se o projeto de um supervisor incremental adequado ainda permanece um problema em aberto? Ocorre que a organização dos FAs componentes no caso incremental gera uma estrutura cooperativa em cascata, em que é possível explorá-la mesmo para supervisores *incrementais fixos*, isto é, $\lambda_n(i) \to \lambda_n$.

O cenário-chave em que um supervisor fixo é útil, podendo levar a uma considerável melhoria de desempenho, é aquele em que a planta a ser identificada é não-estacionária. É conduzida análise de desempenho em ambientes não-estacionários

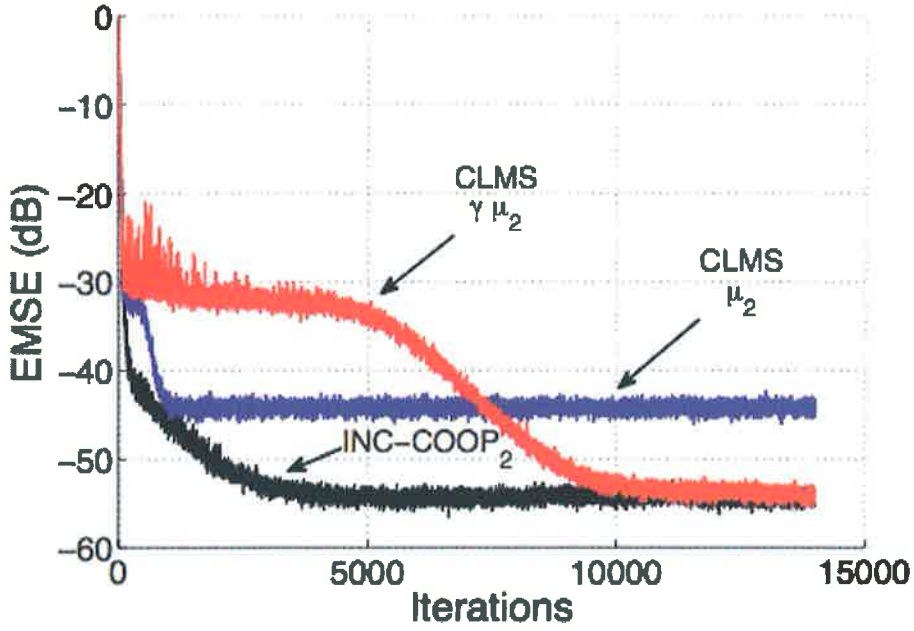


Figura 10: Comparação: CLMS, CLMS com $\gamma\mu_2$, e LMS→LMS incremental simultâneo (INC-COOP2); ambos com $\gamma = 0.1$ [3].

para quantificar a capacidade de rastreamento das estruturas incrementais, em particular de uma família de combinações incrementais da forma

$$\begin{aligned} w_1 &= w_{i-1} + \lambda H_{1,i} u_i^* [d(i) - u_i w_{i-1}] \\ w_i &= w_1 + (1-\lambda) H_{2,i} u_i^* [d(i) - u_i w_1]. \end{aligned} \tag{3.13}$$

As matrizes $H_{n,i}$, $n = 1, 2$, são dependentes dos dados e são definidas de acordo com a regra de adaptação envolvida. Elas cobrem as regras LMS, NLMS e RLS, entre outras, tal que as cadeias incrementais LMS→LMS (3.10) e RLS→LMS (3.12), equipadas com supervisor fixo λ, tornam-se casos particulares de (3.13).

3.3.1 Modelo de dados e conservação da energia

A análise de desempenho no sentido médio quadrático é conduzida via relações de conservação de energia (ECR - "Energy Conservation Relation") [14]. O resultado deste estudo produz modelos analíticos para o EMSE e MSE que descrevem

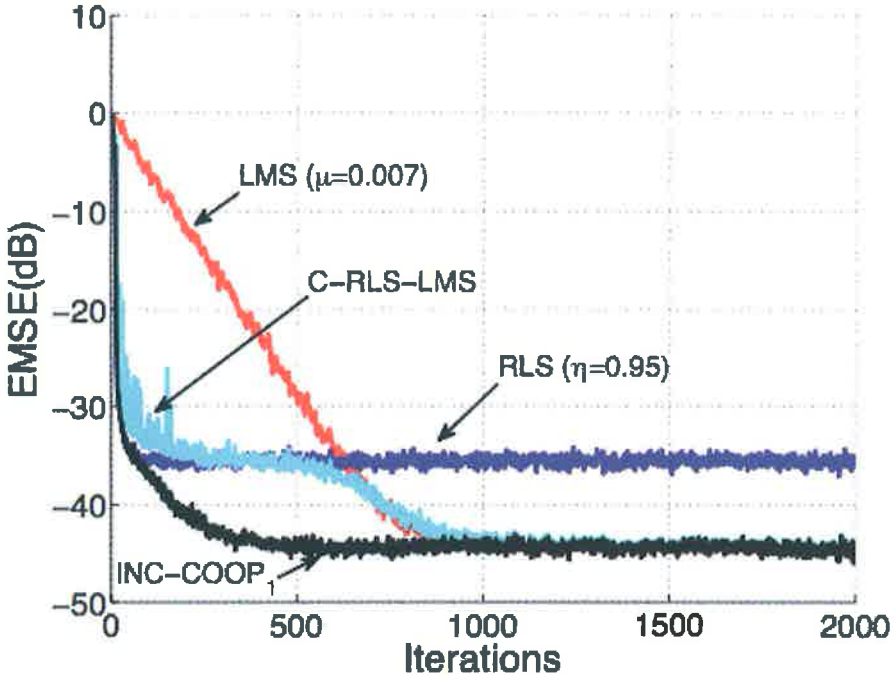


Figura 11: RLS-LMS paralelo (3.8) e RLS→LMS incremental (3.12) (INC-COOP1). Parâmetros obtidos de [4].

a capacidade de rastreamento de estruturas da forma (3.13) quando a planta sob identificação é não-estacionária.

Os sinais envolvidos na implementação da cascata incremental são usualmente modelados como processos estocásticos, ou seja $\boldsymbol{d}(i)$ e $\boldsymbol{u}_i$. Assim, as equações *determinísticas* (3.13), que seriam implementadas em um processador digital de sinais (DSP), tornam-se *equações estocásticas*, que são um modelo para a combinação incremental

$$\boldsymbol{w}_1 = \boldsymbol{w}_{i-1} + \lambda \boldsymbol{H}_{1,i} \boldsymbol{u}_i^* [\boldsymbol{d}(i) - \boldsymbol{u}_i \boldsymbol{w}_{i-1}] \tag{3.14}$$

$$\boldsymbol{w}_i = \boldsymbol{w}_1 + (1-\lambda) \boldsymbol{H}_{2,i} \boldsymbol{u}_i^* [\boldsymbol{d}(i) - \boldsymbol{u}_i \boldsymbol{w}_1]. \tag{3.15}$$

Além de estocásticas, as equações que descrevem filtros adaptativos (e suas combinações) são sistemas não-lineares e variantes no tempo; diante disto, visando tratabilidade matemática torna-se necessário adotar hipóteses simplificadoras. Estas hipóteses são reunidas em um *modelo de dados*, descrito a seguir [24,32,33]:

1. Existe um vetor $\boldsymbol{w}_i^o$ tal que $\boldsymbol{d}(i) = \boldsymbol{u}_i \boldsymbol{w}_i^o + \boldsymbol{v}(i)$;

2. O vetor de coeficientes desconhecido obedece uma lei do tipo "random walk", ou seja $\boldsymbol{w}_i^o = \boldsymbol{w}_{i-1}^o + \boldsymbol{q}_i$;

3. O ruído é modelado como uma seqüencia em que $\{\boldsymbol{v}(i)\}$ é i.i.d. com variância constante $\sigma_v^2 = E|\boldsymbol{v}(i)|^2$

4. O ruído $\{\boldsymbol{v}(i)\}$ é independente do sinal $\boldsymbol{u}_j$ para todo i, j;

5. A sequência de vetores $\{\boldsymbol{q}_i\}$ possui matriz de covariância $Q \triangleq E\boldsymbol{q}_i\boldsymbol{q}_i^*$ e é independente de $\{\boldsymbol{v}(i), \boldsymbol{u}_j\}$ para todo i, j;

6. As condições iniciais $\{\boldsymbol{w}_{-1}, \boldsymbol{w}_{-1}^o\}$ são independentes de todos os sinais $\{\boldsymbol{d}(j), \boldsymbol{u}_j, \boldsymbol{v}(j), \boldsymbol{q}_j\}$;

7. A matriz de covariância do regressor é denotada por $R_u = E\boldsymbol{u}_i^*\boldsymbol{u}_i > 0$

8. As variáveis $\{\boldsymbol{d}(i), \boldsymbol{v}(i), \boldsymbol{u}_i, \boldsymbol{q}_i\}$ tem média nula;

9. O vetor de coeficientes $\boldsymbol{w}_i^o$ tem média constante w^o;

Para prosseguir, é conveniente para a análise colocar as duas equações recursivas da combinação em uma única recursão em termos de uma matriz $\boldsymbol{H}_i$ que depende dos dados; substituindo-se (3.14) em (3.15) tem-se

$$\boldsymbol{w}_i = \boldsymbol{w}_{i-1} + \boldsymbol{H}_i \boldsymbol{u}_i^* \boldsymbol{e}(i) \tag{3.16}$$

com

$$\boldsymbol{H}_i \ = [\lambda \boldsymbol{H}_{1,i} + (1-\lambda)\boldsymbol{H}_{2,i}(1 - \lambda\|\boldsymbol{u}_i\|^2_{\boldsymbol{H}_{1,i}})] \tag{3.17}$$

Em que $\boldsymbol{H}_i$ é uma matriz que depende do sinal regressor $\boldsymbol{u}(i)$, e $\|\boldsymbol{u}_i\|^2_{\boldsymbol{H}_i} \triangleq \boldsymbol{u}_i \boldsymbol{H}_i \boldsymbol{u}_i^*$ é a norma ponderada de $\boldsymbol{u}_i$. No caso geral, $\|x\|^2_\Sigma = x\Sigma x^*$.

As técnicas ECR são um balanço de energia sobre a equação do sistema

adaptativo a partir das seguintes definiçoes de erros a priori e a posteriori

$$\begin{cases} \tilde{\boldsymbol{w}}_{i-1} \triangleq (\boldsymbol{w}_{i-1}^o - \boldsymbol{w}_{i-1}) & \text{vetor de erro nos coeficientes} \\ \boldsymbol{e}_a(i) = \boldsymbol{u}_i(\boldsymbol{w}_i^o - \boldsymbol{w}_{i-1}) & \text{erro de estimação } \textit{a priori} \\ \boldsymbol{e}_p(i) = \boldsymbol{u}_i\tilde{\boldsymbol{w}}_i & \text{erro de estimação } \textit{a posteriori} \end{cases} \tag{3.18}$$

O objetivo é determinar modelos analíticos para a combinação incremental adaptativa das quantidades EMSE e MSE em regime permanente, definidos como

$$EMSE = \zeta \triangleq \mathrm{E}\,|\boldsymbol{e}_a(i)|^2 \tag{3.19}$$

$$MSE = \xi \triangleq \mathrm{E}\,|\boldsymbol{e}(i)|^2 \tag{3.20}$$

Pelo modelo de dados, tem-se

$$\begin{aligned} \boldsymbol{e}(i) &= \boldsymbol{d}(i) - \boldsymbol{u}_i\boldsymbol{w}_{i-1} \\ &= \boldsymbol{u}_i\boldsymbol{w}_i^o + \boldsymbol{v}(i) - \boldsymbol{u}_i\boldsymbol{w}_{i-1} \\ &= \boldsymbol{e}_a(i) + \boldsymbol{v}(i) \end{aligned} \tag{3.21}$$

Assim, calculando-se ζ determina-se ξ via

$$\xi = \zeta + \sigma_v^2 \tag{3.22}$$

O primeiro passo é manipular as equações recursivas da combinação incremental para expressá-las em termos das quantidades de erros recém definidas. A partir da ECR resultante, é possivel derivar uma relação de variância que permite calcular o EMSE e o MSE da combinação (Para detalhes, veja [14], [26]).

Subtraindo (3.16) de $\boldsymbol{w}_i^o$ resulta em

$$(\boldsymbol{w}_i^o - \boldsymbol{w}_i) = (\boldsymbol{w}_i^o - \boldsymbol{w}_{i-1}) - \boldsymbol{H}_i\boldsymbol{u}_i^*\boldsymbol{e}(i). \tag{3.23}$$

pré-multiplicando o resultado por $\boldsymbol{u}_i$ e usando as definições de erros (3.18) leva a

$$\boldsymbol{e}_p(i) = \boldsymbol{e}_a(i) - \|\boldsymbol{u}_i\|^2_{\boldsymbol{H}_i}\boldsymbol{e}(i). \tag{3.24}$$

Isolando-se $\boldsymbol{e}(i)$ em (3.24) e substituindo em (3.23) tem-se

$$(\boldsymbol{w}_i^o - \boldsymbol{w}_i) + \frac{\boldsymbol{H}_i\boldsymbol{u}_i^*}{\|\boldsymbol{u}_i\|^2_{\boldsymbol{H}_i}}\boldsymbol{e}_a(i) = (\boldsymbol{w}_i^o - \boldsymbol{w}_{i-1}) + \frac{\boldsymbol{H}_i\boldsymbol{u}_i^*}{\|\boldsymbol{u}_i\|^2_{\boldsymbol{H}_i}}\boldsymbol{e}_p(i)\ . \tag{3.25}$$

A equação (3.25) envolve as definições de erros a priori e a posteriori; o balanço de energia é feito sobre ela em termos destes erros. Desta forma, usando $\boldsymbol{H}_i^{-1}$ como uma matriz de ponderação e igualando as normas quadráticas ponderadas em (3.25) obtém-se

$$\|\boldsymbol{w}_i^o - \boldsymbol{w}_i\|^2_{\boldsymbol{H}_i^{-1}} + \overline{\mu}(i)|\boldsymbol{e}_a(i)|^2 = \|\boldsymbol{w}_i^o - \boldsymbol{w}_{i-1}\|^2_{\boldsymbol{H}_i^{-1}} + \overline{\mu}(i)|\boldsymbol{e}_p(i)|^2, \tag{3.26}$$

em que $\overline{\mu}(i) \triangleq (\|\boldsymbol{u}_i\|^2_{\boldsymbol{H}_i})^\dagger = \dfrac{1}{\|\boldsymbol{u}_i\|^2_{\boldsymbol{H}_i}}$, se $\boldsymbol{u}_i \neq 0$ ou igual a zero caso contrário, com † representando a pseudo-inversa [34].

3.3.2 Relação de Variância

Tomando-se o valor esperado em (3.26) tem-se que

$$E\|\tilde{\boldsymbol{w}}_i\|^2_{\boldsymbol{H}_i^{-1}} + E\overline{\mu}(i)|\boldsymbol{e}_a(i)|^2 = E\|\boldsymbol{w}_i^o - \boldsymbol{w}_{i-1}\|^2_{\boldsymbol{H}_i^{-1}} + E\overline{\mu}(i)|\boldsymbol{e}_p(i)|^2 \tag{3.27}$$

Usando o modelo random-walk para $\boldsymbol{w}_i^o$ no lado direito de (3.27) chega-se a

$$E\|\tilde{\boldsymbol{w}}_i\|^2_{\boldsymbol{H}_i^{-1}} + E\overline{\mu}(i)|\boldsymbol{e}_a(i)|^2 = E\|\tilde{\boldsymbol{w}}_{i-1}\|^2_{\boldsymbol{H}_i^{-1}} + E\|\boldsymbol{q}_i\|^2_{\boldsymbol{H}_i^{-1}} + E\overline{\mu}(i)|\boldsymbol{e}_p(i)|^2 \tag{3.28}$$

Em regime permanente ($i \to \infty$), é razoável assumir que $\tilde{\boldsymbol{w}}_{i-1}$ é independente de $\boldsymbol{u}_i$; assim, como $\boldsymbol{H}_i$ é uma função de $\boldsymbol{u}_i$, tem-se pela definição de norma

ponderada que

$$
\begin{aligned}
E\|\tilde{\boldsymbol{w}}_i\|^2_{\boldsymbol{H}_i^{-1}} &= E\|\tilde{\boldsymbol{w}}_i\|^2_{E\boldsymbol{H}_i^{-1}} = E\|\tilde{\boldsymbol{w}}_i\|^2_{H^{-1}} \\
E\|\tilde{\boldsymbol{w}}_{i-1}\|^2_{\boldsymbol{H}_i^{-1}} &= E\|\tilde{\boldsymbol{w}}_{i-1}\|^2_{E\boldsymbol{H}_i^{-1}} = E\|\tilde{\boldsymbol{w}}_{i-1}\|^2_{H^{-1}}
\end{aligned}
$$

onde assumiu-se (em regime) que $H \triangleq E\boldsymbol{H}_i$, e utilizou-se a aproximação $E(\boldsymbol{H}_i^{-1}) \approx [E(\boldsymbol{H}_i)]^{-1} = H^{-1}$. Ainda, após a convergência do filtro, é de se supor que $E\|\tilde{\boldsymbol{w}}_{i-1}\|^2_{H^{-1}} \rightarrow E\|\tilde{\boldsymbol{w}}_i\|^2_{H^{-1}}$; também considerando o modelo de dados, (3.28) resulta em

$$
E\overline{\mu}(i)|\boldsymbol{e}_a(i)|^2 = E\|\boldsymbol{q}_i\|^2_{H^{-1}} + E\overline{\mu}(i)|\boldsymbol{e}_p(i)|^2 \tag{3.29}
$$

Aplicando-se (3.24) em (3.29) chega-se à *relação de variância*

$$
E\|\boldsymbol{u}_i\|^2_H|\boldsymbol{e}(i)|^2 + E\|\boldsymbol{q}_i\|^2_{H^{-1}} = 2R_e\{E\boldsymbol{e}_a^*(i)\boldsymbol{e}(i)\} \tag{3.30}
$$

3.3.3 Rastreamento e supervisor ótimo

Da relação de variância (3.30) pode-se calcular expressões para o EMSE e o MSE em regime permanente. Aplicando (3.21) em (3.30), e invocando o *princípio da separação* [14] (em regime permanente $\|\boldsymbol{u}_i\|^2_H$ independente de $\boldsymbol{e}_a(i)$) tem-se

$$
2E|\boldsymbol{e}_a(i)|^2 = \sigma_v^2 E\|\boldsymbol{u}_i\|^2_H + E\|\boldsymbol{u}_i\|^2_H E|\boldsymbol{e}_a(i)|^2 + E\|\boldsymbol{q}_i\|^2_{H^{-1}}, \tag{3.31}
$$

Em (3.31) o termo $E|\boldsymbol{e}_a(i)|^2$ é a própria definição de EMSE. Desta maneira, substituindo $E\|\boldsymbol{u}_i\|^2_H = Tr(R_uH)$ e $E\|\boldsymbol{q}_i\|^2_{H^{-1}} = Tr(QH^{-1})$ em (3.31) resulta no EMSE da família de arranjos incrementais da forma (3.13), quando rastreando um sistema não-estacionário:

$$
\boxed{\zeta = \frac{\sigma_v^2 Tr(R_uH) + Tr(QH^{-1})}{2 - Tr(R_uH)}} \tag{3.32}
$$

O MSE é obtido diretamente por (3.22)

$$\boxed{\xi = \zeta + \sigma_v^2} \tag{3.33}$$

Ambas as Equações (3.32) e (3.33) valem sob as premissas adotadas no modelo de dados e requerem o cálculo dos momentos R_u e H, que dependem do sinal de entrada $\boldsymbol{u}(i)$ e das regras adaptativas adotadas para os FAs componentes.

Dependendo da natureza da matriz H, que captura as regras de adaptação dos FAs componentes, do grau de não-estacionaridade σ_q^2 e do nível de ruído σ_v^2, é possível determinar o(s) supervisor(es) ótimo(s) λ^o que minimiza(m) ζ em (3.32). Contudo, a solução pode não ser determinada na forma fechada, para regras adaptativas complexas. Soluções podem ser encontrada via gradiente descendente, ou na forma fechada para regras mais simples

$$\boxed{\lambda^o = \{\lambda | \nabla_\lambda \zeta(\lambda^o) = 0\}}, \qquad \nabla_\lambda^2 \zeta(\lambda^o) > 0 \tag{3.34}$$

O valor ótimo λ^o para o supervisor incremental pode ser usado nas implementações incrementais, ou pode ser visto como um resultado téorico que deve guiar o projeto de futuros supervisores incrementais adaptativos.

Nas próximas seções, o modelo (3.32) será particularizado para as combinações incrementais LMS→LMS e RLS→LMS e validado através de simulações, comprovando o bom desempenho das estruturas incrementais para rastreamento, especialmente se comparadas às combinações paralelas.

3.4 LMS→LMS

Por conveniência, repete-se as equações do algoritmo LMS→LMS

$$w_{1,i} = w_{i-1} + \lambda\mu_1 u_i^*(d(i) - u_i w_{i-1}) \tag{3.35}$$

$$w_i = w_{1,i} + (1-\lambda)\mu_2 u_i^*(d(i) - u_i w_{1,i}) \tag{3.36}$$

Neste caso, tem-se para as matrizes $H_{1,i}$ e $H_{2,i}$

$$\boldsymbol{H}_{1,i} = \mu_1 I \ , \qquad \boldsymbol{H}_{2,i} = \mu_2 I \tag{3.37}$$

em que μ_1 e μ_2 são os passos de cálculo adotados para os FAs LMS, definidos como

$$\mu_1 = \frac{\mu_o}{M\sigma_u^2} \ , \qquad \mu_2 = \frac{r_o\mu_o}{M\sigma_u^2} \ , \tag{3.38}$$

em que $0 < \mu_o < 1$ e $r_o \in [0,1]$ é um parâmetro que quantifica a razão μ_2/μ_1. Os passos são definidos de maneira que μ_1 é rápido mas garantidamente estável, e μ_2 implementa o filtro preciso, sendo ajustado cerca de uma ou duas décadas abaixo de μ_1 ($0 < r_o \ll 1$). Para a matriz $\boldsymbol{H}_i$ tem-se

$$\boldsymbol{H}_i = [\lambda\mu_1 + (1-\lambda)\mu_2(1 - \lambda\mu_1\|\boldsymbol{u}_i\|^2)] \cdot I \ , \tag{3.39}$$

que prontamente leva a

$$H = \left[\lambda\mu_1 + (1-\lambda)\mu_2[1 - \lambda\mu_1 Tr(R_u)]\right] \cdot I \tag{3.40}$$

do qual pode-se calcular o EMSE (MSE) por (3.32) e (3.33).

Usando (3.40) em (3.32) gera-se superfícies de desempenho em função de μ_o e de λ, que são ilustradas nas figuras na sequência. Regressores são Gaussianos, brancos ou altamente correlacionados. Para obter sinais correlacionados, filtra-se um sinal branco via processo auto-regressivo de primeira ordem $\sqrt{1-b^2}/(1-bz^{-1})$, $b = 0.98$. A ordem do sistema desconhecido é $M = 20$,

$\sigma_u^2 = 1$ e $\sigma_v^2 = 0.1$. Para o caso não-estacionário, $\sigma_q^2 = 10^{-6}$ ($\sigma_q^2 = 0$ recupera o caso estacionário). As curvas foram geradas com 200 realizações, e tirando a média temporal das últimas 300 amostras das curvas para cálculo dos valores em regime permanente.

A Fig. 12 mostra a superfície do EMSE para sinais correlacionados e planta estacionária ($\sigma_q^2 = 0$); conforme $\lambda \to 0$ (resgata o filtro preciso) e $\mu_0 \to 0$, o EMSE tende a zero, o que é esperado para cenários estacionários, e o supervisor incremental fixo ótimo é trivializado $\lambda^o \to 0$.

Um exemplo de planta não-estacionária é apresentado na superfície da Fig. 13, que repete os dados da figura anterior, mas agora com $\sigma_q^2 = 10^{-6}$. Nota-se que existe uma infinidade de pares $[\mu_o, \lambda^o]$ em que a estrutura é *universal*, com a combinação atingindo erros menores que os FAs componentes individuais.

Comparações entre a estrutura incremental e a paralela (rotuladas CLMS) são feitas nas Figs. 14 e 15, obtidas para $\mu_o = 0.3$ e $\mu_o = 0.6$, respectivamente, e simuladas para λ fixo no intervado $[0, 1]$. Na Fig. 14 é nítida a superioridade da estrutura incremental para toda a faixa de λ. Já na Na Fig. 15, com passos maiores, a combinação paralela divergiu para todo λ. A estrutura incremental permaneceu estável e universal para $0 < \lambda < 0.6$. A divergência foi causada pelo correlacionamento dos dados e valores de passo altos. Em ambas as curvas, nota-se boa concordância entre o previsto pelo modelo teórico desenvolvido e os resultados de simulação.

Um fato importante vem da equação (3.39): ela sugere que a estrutura incremental LMS→LMS pode ser motivada como um FA LMS com passo variável

$$\boldsymbol{\mu}(i) = \lambda\mu_1 + (1-\lambda)\mu_2(1 - \lambda\mu_1\|\boldsymbol{u}_i\|^2) \tag{3.41}$$

Desta forma, a estrutura incremental implementada como em (3.35) está sujeita ao limite teórico de desempenho àquele de um FA LMS em regime de rastrea-

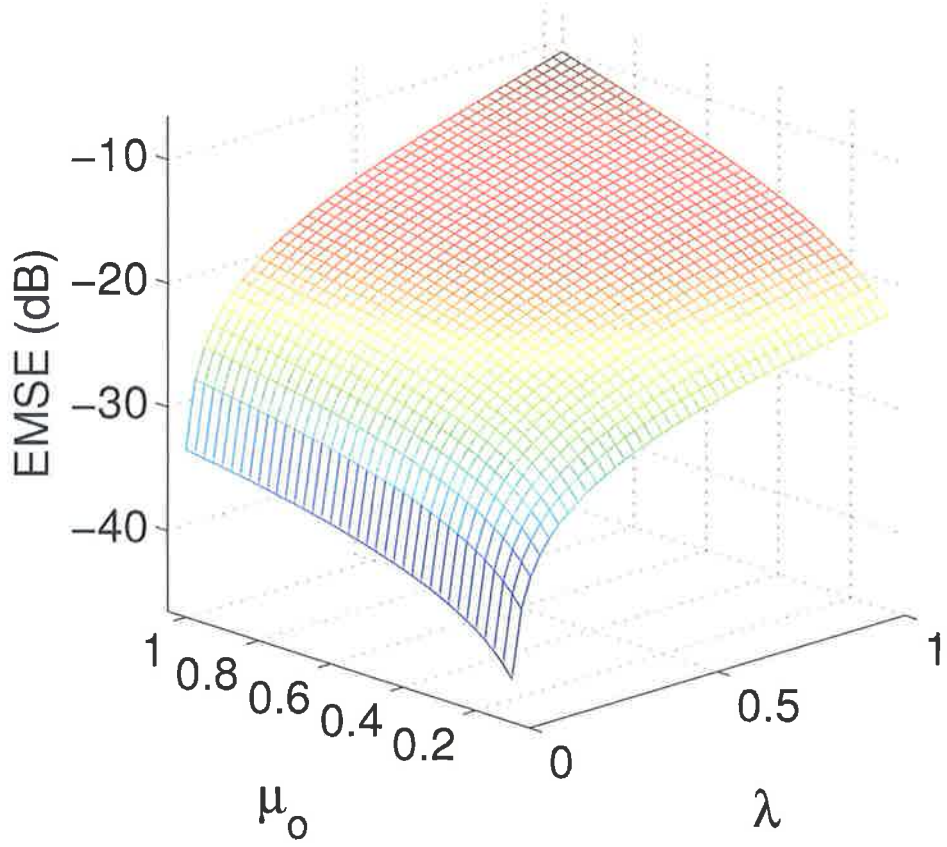


Figura 12: Superfície de desempenho LMS→LMS: EMSE em função de λ e μ_o. Caso estacionário ($\sigma_q^2 = 0$) e correlacionado com $\sigma_u^2 = 1$ e $\sigma_v^2 = 0.1$. A estrutura não apresenta universalidade.

mento, no qual é possível determinar um passo ótimo μ^o que minimize o EMSE. No contexto da combinação incremental, é possível achar diversos valores de λ tal que o valor médio do passo em (3.41) seja igual a μ^o. Isto ocorre porque a estrutura de dados adotada usa o mesmo par de dados $\{d(i), u_i\}$ para todos os $N = 2$ FAs (também vale para $N > 2$). Este modo de operação é denominado *compartilhamento de dados*. Outra possibilidade é o *reuso de dados*, que emprega uma massa de dados maior na cascata de filtros, e faz o desempenho resultante superar com larga vantagem o LMS simples, particularmente para $N > 2$. Isto

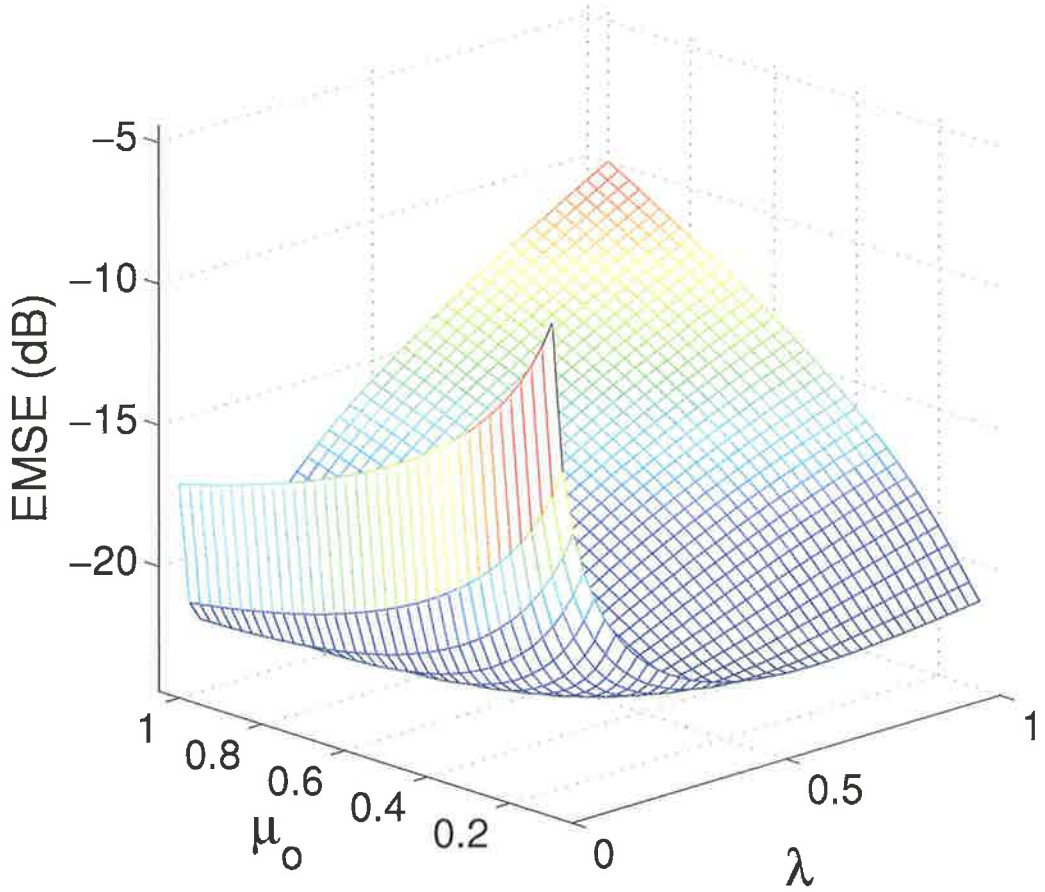


Figura 13: Superfície de desempenho LMS→LMS: EMSE em função de λ e μ_o. Caso não-estacionário ($\sigma_q^2 = 10^{-6}$), correlacionado com $\sigma_u^2 = 1$ e $\sigma_v^2 = 0.1$. A estrutura é universal, com diversos λ^o não-triviais em função de μ_o.

será abordado no próximo capítulo.

3.5 RLS→LMS

O algoritmo RLS→LMS é a contrapartida incremental do algoritmo paralelo RLS–LMS (3.8). Para facilidade de referência, coloca-se novamente as suas

equações para supervisor fixo λ

$$P_i = \eta^{-1}\left[P_{i-1} - \frac{\eta^{-1}P_{i-1}u_i^* u_i P_{i-1}}{1+\eta^{-1}u_i P_{i-1}u_i^*}\right], \quad P_{-1} = \epsilon^{-1}I \tag{3.42}$$

$$w_{1,i} = w_{i-1} + \lambda P_i u_i^*(d(i) - u_i w_{i-1}) \tag{3.43}$$

$$w_i = w_{1,i} + (1-\lambda)\mu u_i^*(d(i) - u_i w_{1,i}) \tag{3.44}$$

O arranjo incremental $RLS \rightarrow LMS$ procura aliar o efeito "branqueador" indireto do FA RLS com a janela de observação instantânea do LMS, o que é util para cenários adversos, como na identificação de sistemas não-estacionários em que o sinal regressor $u(i)$ é (altamente) correlacionado. É possível explorar, ainda, estruturas mais simples de implementação do RLS [35], o que será conduzido em trabalhos futuros.

Para a combinação (3.42), tem-se

$$\boldsymbol{H}_{1,i} = \boldsymbol{P}_i \quad \text{e} \quad \boldsymbol{H}_{2,i} = \mu I \tag{3.45}$$

em que $\boldsymbol{P}_i$ é obtido pela recursão [32],

$$\boldsymbol{P}_i = \eta^{-1}\left[\boldsymbol{P}_{i-1} - \frac{\eta^{-1}\boldsymbol{P}_{i-1}\boldsymbol{u}_i^*\boldsymbol{u}_i\boldsymbol{P}_{i-1}}{1+\eta^{-1}\boldsymbol{u}_i\boldsymbol{P}_{i-1}\boldsymbol{u}_i^*}\right], \boldsymbol{P}_{-1} = \epsilon^{-1}I, \tag{3.46}$$

com ϵ um parâmetro de regularização, η o fator de esquecimento, I a $M \times M$ matriz identidade, relembrando que $\mu = \mu_o/(M\sigma_u^2)$, $\mu_o \in (0,1]$. Portanto,

$$\boldsymbol{H}_i = [\lambda \boldsymbol{P}_i + \mu(1-\lambda)(1-\lambda\|\boldsymbol{u}_i\|_{\boldsymbol{P}_i}^2)I]. \tag{3.47}$$

Para calcular o EMSE e MSE da estrutura, é necessário determinar $E\boldsymbol{H}_i$ a

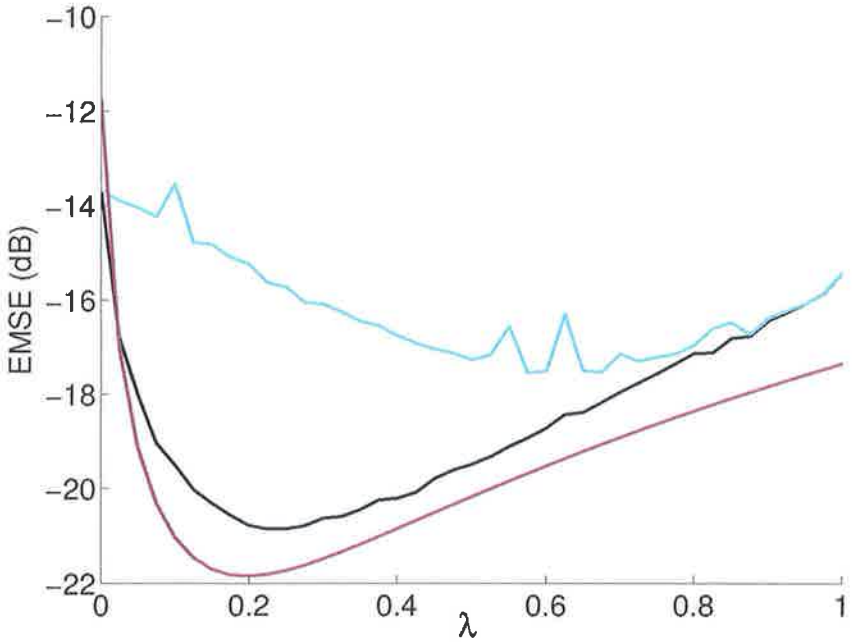


Figura 14: Ábacos EMSE x λ para LMS→LMS: teoria (curva de baixo magenta), simulação (curva do meio preta) e combinação paralela (curva de cima ciano); $\mu_o = 0.3$, $r_o = 0.01$ e $\sigma_q^2 = 10^{-6}$. A estrutura incremental é universal e claramente superior à combinação paralela.

partir de (3.47). Em regime permanente, com $i \to \infty$, é razoável calcular

$$E\boldsymbol{P}_i^{-1} = \frac{R_u}{1-\eta} \triangleq P^{-1} \tag{3.48}$$

$$E\boldsymbol{P}_i \approx [E(\boldsymbol{P}_i^{-1})]^{-1} = (1-\eta)R_u^{-1} = P \tag{3.49}$$

$$E\|\boldsymbol{u}_i\|_{\boldsymbol{P}_i}^2 \approx E\|\boldsymbol{u}_i\|_P^2 = Tr(R_u P) = (1-\eta)M, \tag{3.50}$$

resultando em

$$H = \{\lambda(1-\eta)R_u^{-1} + \mu(1-\lambda)[1-\lambda(1-\eta)M]I\}. \tag{3.51}$$

Substituindo (3.51) em (3.32), junto com (3.33) retorna o EMSE (MSE) da estrutura incremental RLS→LMS. É uma função de vários parâmetros; neste trabalho fixa-se η e M e, para uma dada R_u e Q, estuda-se o comportamento de ζ termos de μ_o e λ.

As Figs. 16 e 17 ilustram a superfície de desempenho de rastreamento EMSE

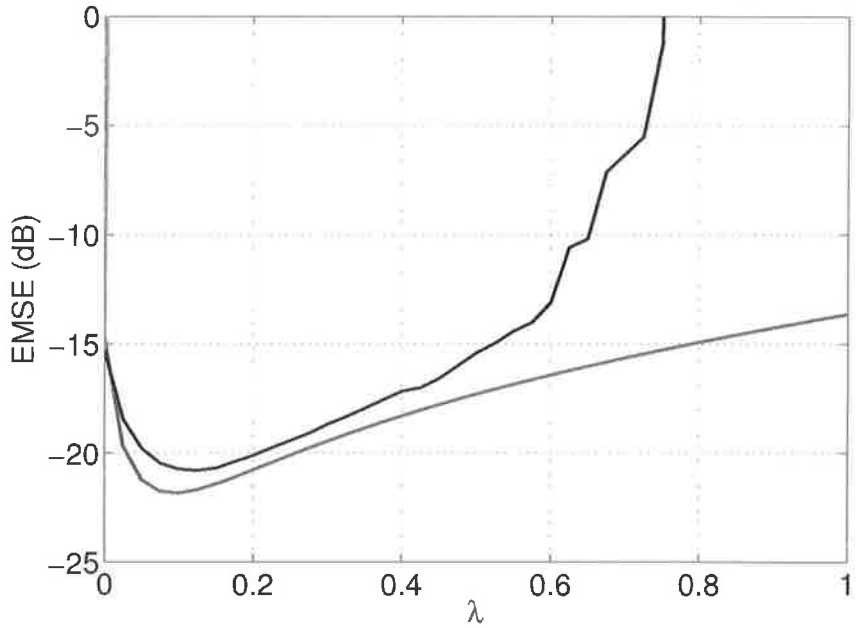
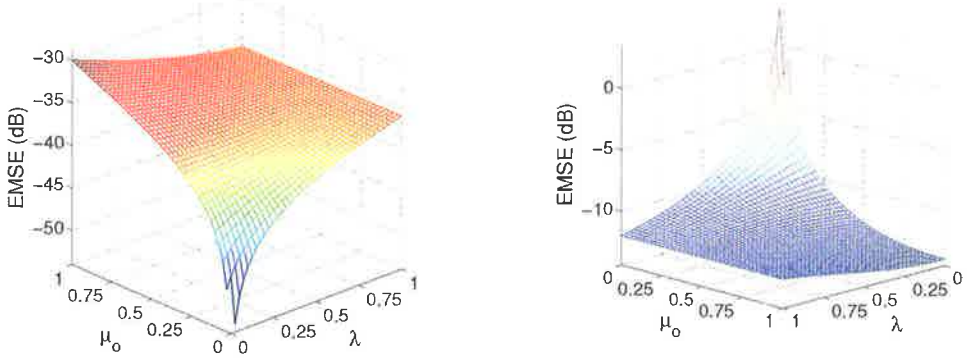


Figura 15: Ábacos EMSE x λ para LMS→LMS: teoria (curva de baixo magenta), simulação (curva do meio preta); a combinação paralela divergiu; $\mu_o = 0.6$, $r_o = 0.01$ e $\sigma_q^2 = 10^{-6}$. A estrutura incremental é universal e estável até $\lambda = 0.6$. A combinação paralela divergiu para todo λ.

em função de μ_o (parâmetro do LMS) e λ (parâmetro do supervisor). São considerados sinais brancos e correlacionados, em ambientes estacionários e não-estacionários.

Os mesmos cenários (quando aplicável) da Seção 3.4 são empregados. O fator de esquecimento do RLS é $\eta = 0.98$, o parâmetro de regularização é $\epsilon = 10^{-5}$ e a ordem do sistema é $M = 20$. As superfícies de desempenho (a) e (b) na Fig. 16 exploram o caso estacionário correlacionado (a) e não-estacionário correlacionado (b), sugerindo uma universalidade trivial, ou seja, conforme varia-se μ_o, o erro é minimizado ou para $\lambda \to 0$ ou para $\lambda \to 1$, indicando que a melhor opção é usar um dos FAs individuais. O caso estacionário branco é muito similar e foi omitido.

Por outro lado, é notável a universalidade pelo formato convexo no caso correlacionado não-estacionário na Fig. 17. Conforme varia-se μ_o, existe um supervisor ótimo não-trivial λ^o. A superfície sugere que aumentar $\mu_o > 1$ leva-



(a) (b)

Figura 16: Combinação RLS→LMS. Casos estacionário correlacionado (a) e não-estacionário branco (b). A estrutura não apresenta universalidade nestes casos.

ria a erros ainda menores; contudo isto pode instabilizar a combinação, dado o alto correlacionamento dos sinais. A superfície também sugere um projeto mais conservativo: diminuindo μ_o não há piora significativa no erro, e tem-se uma estrutura incremental mais estável.

Nos próximos três exemplos são comparados os desempenhos de rastreamento com supervisor fixo convexo dos algoritmos RLS–LMS paralelo e RLS→LMS incremental. As curvas EMSE são geradas para $\mu_o = 0.2, 0.4, 0.9$ e para $\lambda \in [0, 1]$. A planta não-estacionária é inicializada com $\boldsymbol{w}^o_{-1}$ obtido de um processo Gaussiano branco de variância unitária e segue o modelo random walk com $\sigma_q^2 = 10^{-4}$. Os regressores são correlacionados e gerados de acordo com a Seção 3.4, com $\sigma_u^2 = 1$, $\sigma_v^2 = 10^{-3}$, $\eta = 0.98$, $\epsilon = 10^{-5}$.

A Fig. 18 mostra os ábacos comparando o EMSE das estruturas incremental e paralela em regime permanente para $\mu_o = 0.2$; também foi determinado numericamente o supervisor incremental ótimo $\lambda^o = 0.35$. A estrutura incremental é claramente universal e superior à paralela. Esta figura pode ser entendida como uma fatia em $\mu_o = 0.2$ da superfície na Fig. 17. Observa-se boa concordância entre a curva teórica e a simulada. A Fig. 19 apresenta as curvas de aprendi-

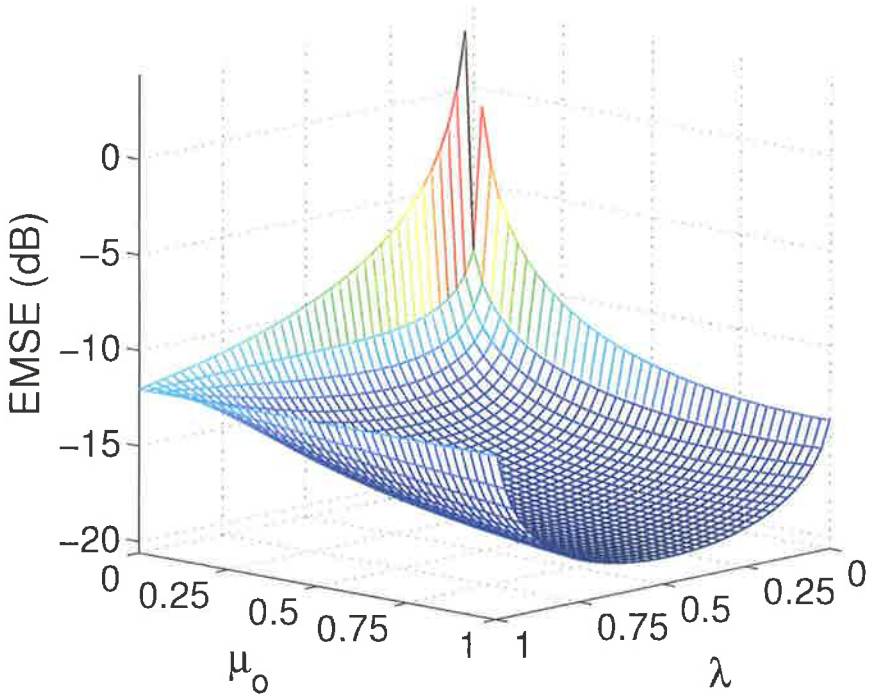


Figura 17: Combinação RLS→LMS. Casos não-estacionário correlacionado. A universalidade é nítida, com o erro diminuindo conforme μ_o aumenta.

zado correspondentes para $\lambda = 0.4$, com uma melhoria de aproximadamente $4dB$ em relação à estrutura paralela. Observa-se boa concordância entre as curvas simulada e teórica.

As Figuras 20 e 21 repetem as simulações anteriores para $\mu_o = 0.4$. Devido ao alto grau de correlacionamento do sinal de entrada, o filtro LMS fica à beira da divergência, o que influencia diretamente a combinação paralela, desestabilizando-a. A estrutura incremental foi capaz de se manter universal e estável para quase toda a faixa de λ; isto ocorre devido à cooperação direta entre os FAs componentes, um FA estabilizou o outro, o que não acontece no caso paralelo. Isto é observado claramente nos ábacos (Fig. 20) e nas curvas de aprendizado geradas para $\lambda = 0.6$ (Fig. 21).

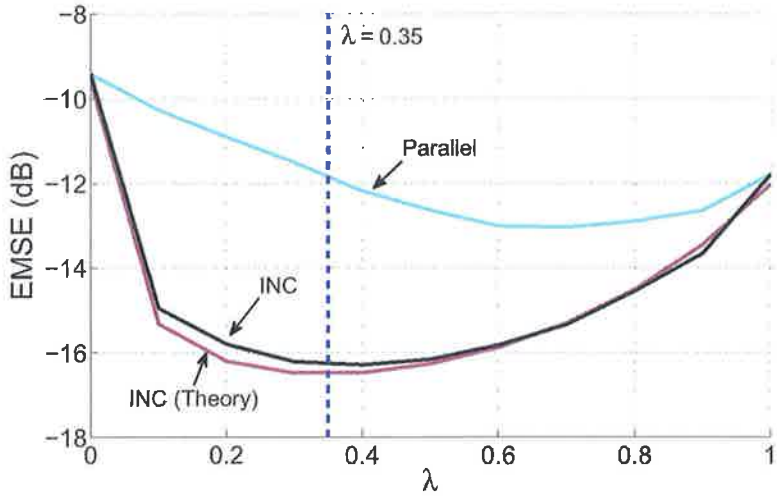


Figura 18: Ábacos EMSE x λ para RLS→LMS (Teoria e simulação) e combinação paralela; $\mu_o = 0.2$ e $\sigma_q^2 = 10^{-4}$. Em azul tracejado o supervisor ótimo λ^o.

As Figs. 22 (a) e (b) abordam o mesmo cenário, apenas aumentando o passo para $\mu_o = 0.9$, enquanto $\lambda = 0.6$. Dos FAa individuais, o RLS converge e o LMS diverge. Por isso, o caso paralelo com supervisor fixo (CVX_1) diverge em todo intervalo $\lambda \in [0, 1]$ e não é mostrado. Também foi simulado a combinação paralela com supervisor *adaptativo* $\lambda(i)$ dado por (3.8) (CVX_2), mas também não houve resposta adequada, mostrando a maior sensibilidade da combinação paralela. Em contrapartida, a estrutura incremental é estável para $\lambda \geq 0.5$, e o desempenho foi melhorado ainda mais em relação aos exemplos anteriores, com uma melhoria de cerca $7dB$ em relação ao único filtro que convergiu, o RLS.

A Fig. 22-(b) é a versão ampliada das curvas convergentes da Fig. 22-(a), e ilustra bem o efeito de estabilização imposto pelo arranjo incremental: Mesmo com um passo grande ($\mu_o = 0.9$), a estrutura incremental é capaz de recuperar a instabilidade em potencial do filtro LMS, além da melhoria da capacidade de rastreamento frente ao FA individual RLS. A combinação divergiria para valores pequenos de λ, porque o componente RLS seria minimizado.

A Fig. 23 apresenta ábacos ilustrando o comportamento universal da estrutura

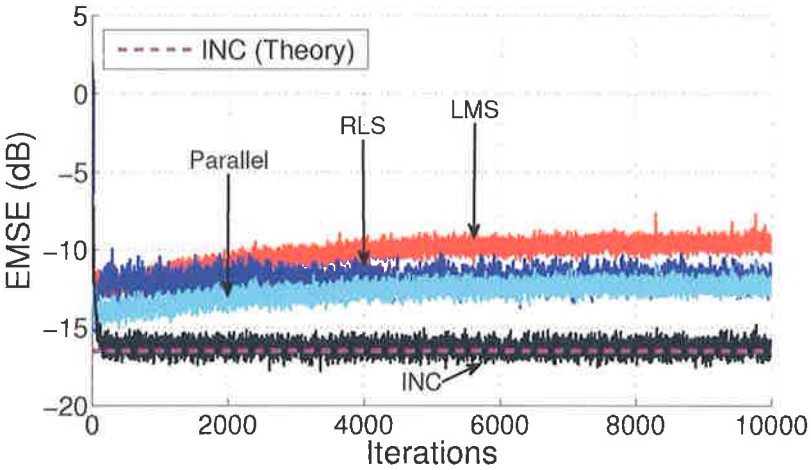


Figura 19: Comparação para $\mu_o = 0.2$ e $\lambda = 0.4$: RLS→LMS simulado (preto) e teoria (tracejado); e combinação paralela. Rodadas 500 realizações.

incremental para diferentes graus de não-estacionaridade (σ_q^2). Observa-se uma melhoria em desempenho da combinação incremental de cerca de $5dB$ sobre os FAs componentes. Isto se mantém para uma ampla faixa de σ_q^2.

Ao contrário do que ocorre com a estrutura LMS→LMS, a combinação híbrida RLS→LMS não pode ser motivada nem como um RLS, nem como um LMS de passo variável. Analisando a natureza da matriz $\boldsymbol{H}_i$, há a parcela $\lambda \boldsymbol{P}_i$ referente ao RLS, mas ponderada pelo supervisor λ. Há ainda a parcela $\mu(1-\lambda)(1-\lambda\|\boldsymbol{u}_i\|^2_{\boldsymbol{P}_i})I$, que vem da perturbação incremental do LMS; esta parcela pode ser entendida como um fator de regularização variável o que, de fato, foi confirmado pelas simulações, em que se observa o efeito estabilizador da combinação incremental sobre os FAs individuais. Contudo, não se trata, apenas, de um algoritmo RLS regularizado; embora possa ser escrita como uma recursão simples, novamente efeito do compartilhamento de dados (ver próximo capítulo), a atualização de w_{i-1} é composta de duas parcelas que não são colineares, como ocorria no caso LMS→LMS. Há a parcela $\lambda \boldsymbol{P}_i u_i^*(d(i) - u_i w_{i-1})$, que não aponta na mesma direção que o gradiente estocástico $u_i^*(d(i) - u_i w_{i-1})$; e há a parcela $\mu(1-\lambda)(1-\lambda\|\boldsymbol{u}_i\|^2_{\boldsymbol{P}_i})u_i^*(d(i) - u_i w_{i-1})$ que está alinhada com o gradiente

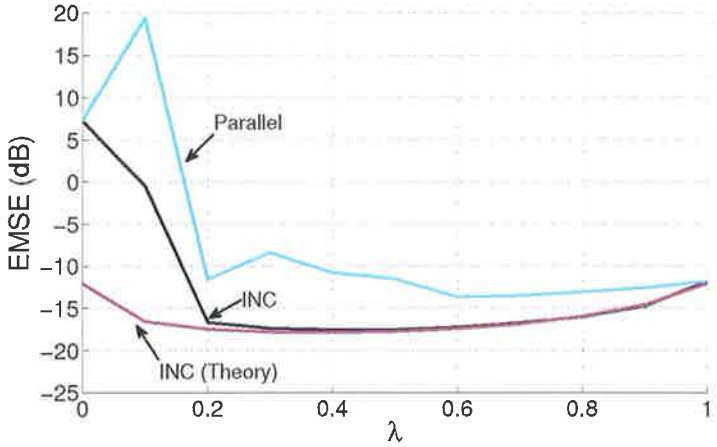


Figura 20: Ábacos EMSE x λ para RLS→LMS (Teoria e simulação) e combinação paralela; $\mu_o = 0.4$ e $\sigma_q^2 = 10^{-4}$.

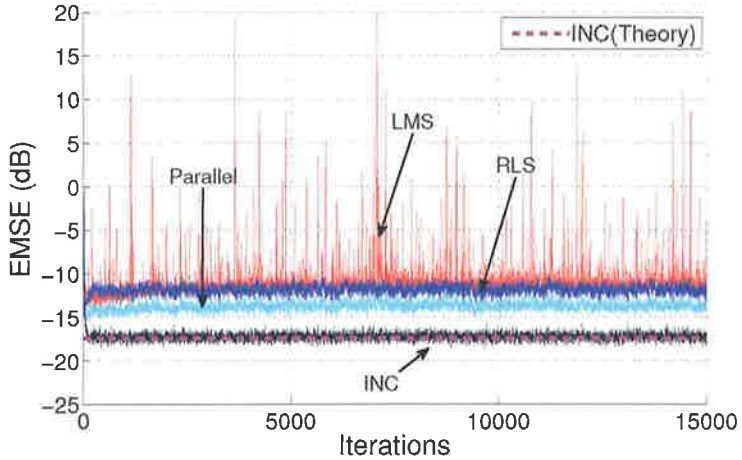


Figura 21: Comparação para $\mu_o = 0.4$ e $\lambda = 0.6$: RLS→LMS simulado (preto) e teoria (tracejado); e combinação paralela. Rodadas 500 realizações.

estocástico. A combinação destas duas parcelas gera um novo algoritmo com características próprias, distintas dos FAs individuais RLS e LMS, e que foi induzido pela combinação incremental híbirda de FAs. É um campo muito promissor: combinação incremental híbrida de filtros adaptativos.

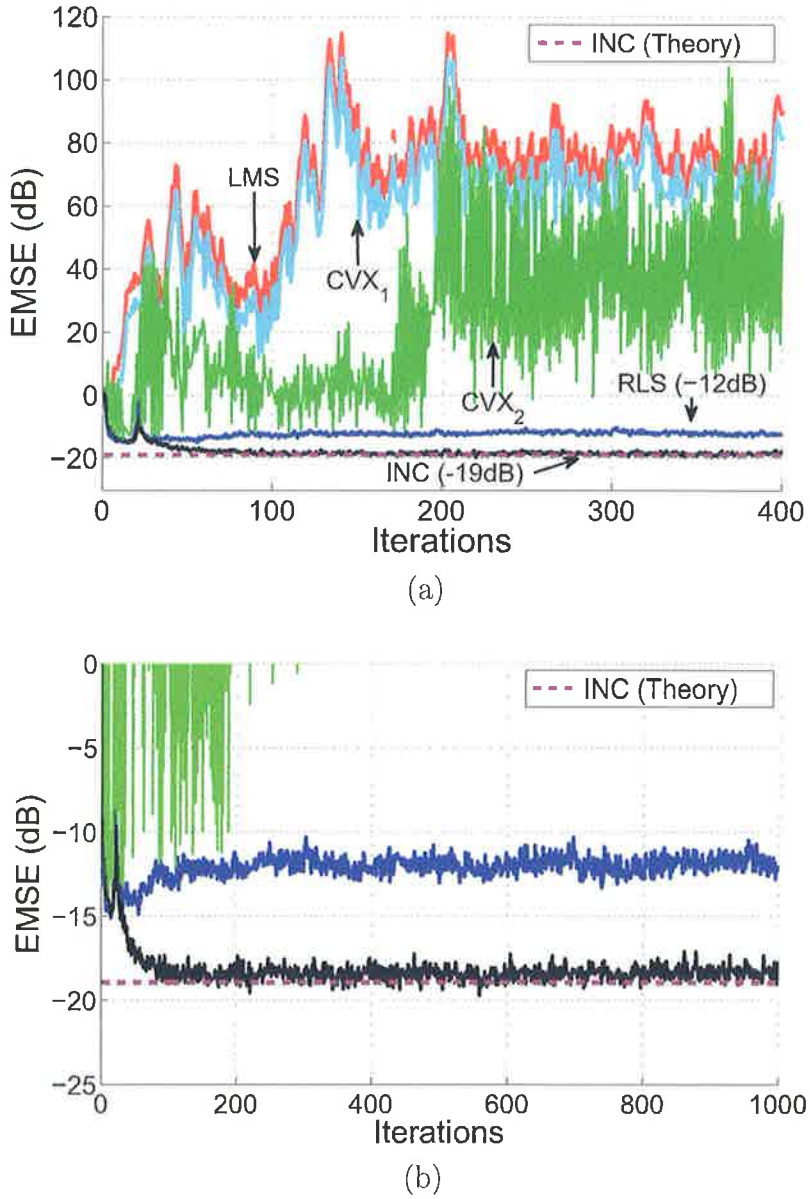


Figura 22: (a) Comparação para $\mu_o = 0.9$, $\lambda = 0.6$ e $\sigma_q^2 = 10^{-4}$: Os FAs individuais LMS e RLS; a estrutura RLS→LMS; a combinação paralela com λ fixo (CVX_1) e com $\lambda(i)$ adaptativo (CVX_2). Sinais correlacionados, 500 realizações. (b) Versão ampliada das curvas convergentes.

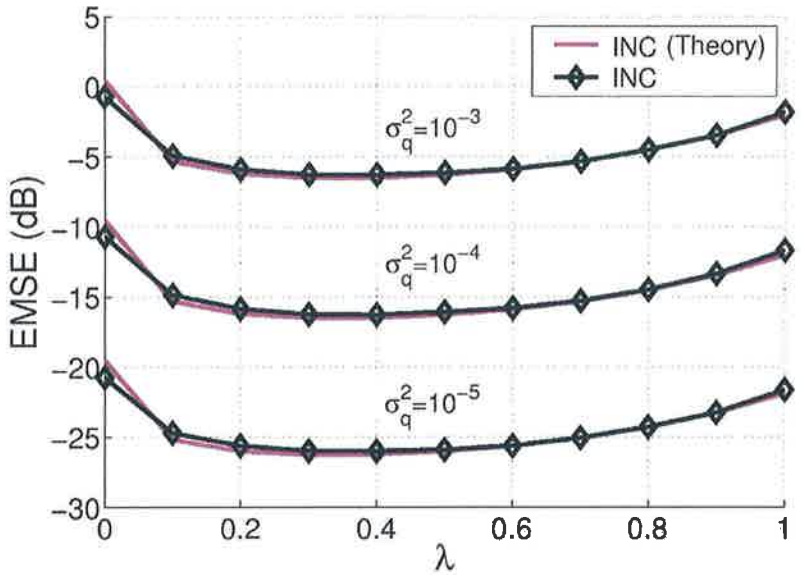


Figura 23: Ábacos para RLS→LMS para diversos valores de σ_q^2 e $\mu_o = 0.2$.

4 REUTILIZAÇÃO DE DADOS EM FORMAS INCREMENTAIS

Reutilização de dados são técnicas bem estabelecidas de melhoria de desempenho de FAs. Consiste em processar várias vezes o par de dados atual, denominado *compartilhamento de dados*, ou inúmeras vezes a massa de dados composta pelos últimos K consecutivos pares de dados, denominado *reuso de dados*. Ambas as técnicas são genericamente referidas neste trabalho como *reutilização de dados* [19], [20], [13, 21–23], [36], [37].

No capítulo anterior foram estudadas cadeias incrementais curtas ($N = 2$) operando no modo de compartilhamento de dados, o que é o modo adotado nos trabalhos de combinações consultados na literatura [1, 3, 4, 7–10], [28–31]. Em verdade, as combinações existentes não abordam explicitamente a questão da reutilização de dados; empregam tacitamente o mesmo par de dados para todos os FAs componentes.

As diferentes regras adaptativas existentes, aliadas aos dois modos de reutilização de dados, levam a uma miríade de combinações de filtros, com desempenho, características e complexidade computacional diferentes.

Neste capítulo coloca-se o problema de reutilização de dados no contexto de combinações de FAs, em particular, estruturas incrementais com cadeias mais longas $N \geq 2$ [11]. Motiva-se o uso de uma massa de dados maior, como uma extensão do modo de compartilhamento de dados, em que cadeias incrementais

com tamanho arbitrário e equipadas com regras de adaptação da família LMS são estudadas. As estruturas resultantes, se bem projetadas, superam em desempenho algoritmos de reuso de dados bastante celebrados na literatura, como o algoritmo de projeções afins (APA) [13, 14], e com complexidade computacional menor.

Continua se explorando o uso de supervisores fixos, que pode ser usado com eficiência no cenário deste capítulo, com cadeias incrementais longas e com reutilização de dados.

4.1 Algoritmo LMS Incremental

A regra de adaptação LMS é frequentemente adotada por sua robustez intrínseca, estabilidade [18] e simplicidade computational [25]. O seu uso dentro da formulação de combinação incremental de filtros, com técnicas de reutilização de dados, leva a resultados bastante promissores.

4.1.1 Técnicas de Reutilização de Dados

Reutilização de dados pode ser implementada usando $K > 1$ vezes o par de dados atual $\{d(i), u_i\}$, denominada *compartilhamento de dados* ("Data Sharing"–DS). Um dos primeiros algoritmos a operar desta maneira, é o Data-Sharing-LMS [37], ou usando a notação de teoria de sistemas, $\{\text{LMS}\}^N$, que implica na utilização de N filtros LMS em cascata incremental

$$w_{0,i} \leftarrow w_{i-1} \tag{4.1}$$

$$w_{k,i} = w_{k-1,i} + \mu_k u_i^*(d(i) - u_i w_{k-1,i}) \tag{4.2}$$

$$w_i \leftarrow w_{K,i} \tag{4.3}$$

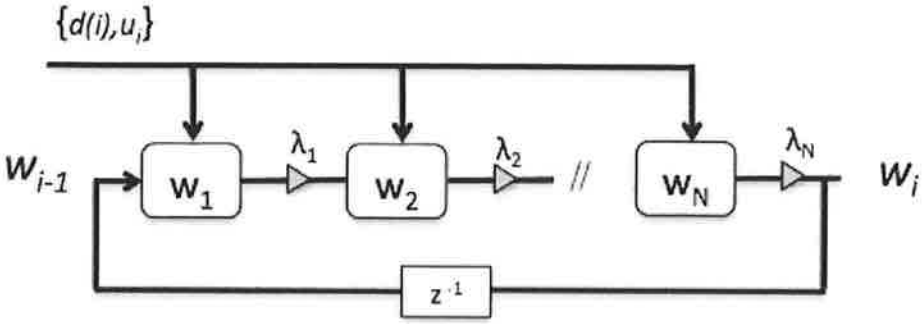


Figura 24: Compartilhamento de dados em estruturas incrementais.

Neste algoritmo, a regra LMS opera $K > 1$ vezes sobre o *mesmo* par de dados atual $\{d(i), u_i\}$. É possível mostrar que, na condição limite $K \to \infty$, este algoritmo tende ao algoritmo NLMS (LMS normalizado) [36].

O outro modo de reutilização de dados opera sobre o conjunto dos últimos K pares de dados $\{d_i, U_i\}$, capturados no vetor $K \times 1$ de sinais desejados e na matriz $K \times M$ de regressores

$$d_i = \text{col}\{\, d(i),\ d(i-1),\ \ldots\ ,\ d(i-K+1)\} \tag{4.4}$$

$$U_i = \text{col}\{\, u_i,\ u_{i-1},\ \ldots\ , u_{i-K+1}\}\ . \tag{4.5}$$

Este modo é denominado *reuso de dados.*

Dois algoritmos de reuso de dados celebrados na literatura são o TRUE–LMS (Cujo nome é derivado de problemas de otimização distribuída [6])

$$w_i = w_{i-1} + \mu U_i^* e_i \tag{4.6}$$

e o ϵ-APA

$$w_i = w_{i-1} + \mu U_i^* \left(\epsilon I + U_i U_i^*\right)^{-1} e_i \tag{4.7}$$

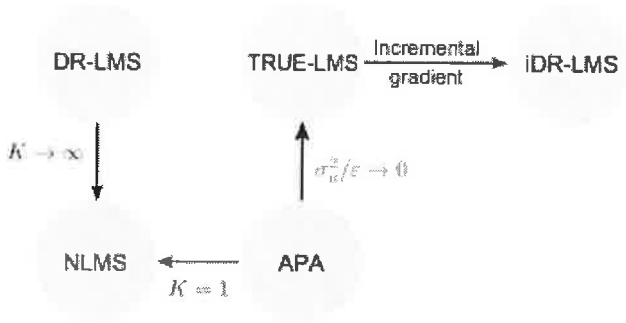


Figura 25: Relação entre as várias estratégias de reutilização de dados.

em que o vetor de erros em ambos os algoritmos é dado por

$$e_i = d_i - U_i w_{i-1} \tag{4.8}$$

É possível mostrar que, conforme $\frac{\sigma_u^2}{\epsilon} \to 0$, o algoritmo ϵ-APA tende ao TRUE-LMS [11]. Apesar de relacionados, o APA apresenta desempenho superior ao TRUE-LMS na maioria dos cenários [12, 13]. A Fig. 25 ilustra a relação entre os algoritmos de reutilização de dados abordados nesta seção, e o novo arranjo incremental desenvolvido na sequência.

4.1.2 Reuso de Dados em Combinações de FAs

Na literatura, as combinações de FAs operam no modo compartilhamento de dados, explorando somente o par de dados atual $\{d(i), u_i\}$. A extensão conceitual consiste em introduzir as técnicas de reuso de dados no arcabouço de combinação de filtros. Ao invés de operar no modo compartilhamento, cada filtro componente n explora o par de dados $\{d(i-n+1), u_{i-n+1}\}$, com $n = 1, \ldots, N$. É possível fazê-lo tanto nas estruturas paralelas, como nas estruturas incrementais.

Revisitando a estrutura paralela com cyclic feedback (3.7), e incorporando

reuso de dados, tem-se (omitiu-se o projeto do supervisor)

$$\begin{aligned} w_{n,i-1} &= \delta_L w_{i-1} + (1-\delta_L) w_{n,i-1} \\ d_n(i) &= d(i-n+1); \quad u_{n,i} = u_{i-n+1} \\ w_{n,i} &= w_{n,i-1} + \mu_n u_{n,i}^* \left(d_n(i) - u_{n,i} w_{n,i-1}\right) \\ w_i &= \sum_{n=1}^{N} \lambda_n(i) w_{n,i} \end{aligned} \tag{4.9}$$

De forma análoga, incorporando reuso de dados na estrutura incremental leva a

$$\begin{aligned} w_{0,i} &\leftarrow w_{i-1} \\ d_n(i) &= d(i-n+1); \quad u_{n,i} = u_{i-n+1} \\ w_{n,i} &= w_{n-1,i} + \lambda_n(i) \mu_n u_{n,i}^* \left(d_n(i) - u_{n,i} w_{n-1,i}\right) \\ w_i &\leftarrow w_{N,i} \end{aligned} \tag{4.10}$$

É interessante notar que é possível recuperar as combinações convencionais de compartilhamento de dados paralela (3.7) e a incremental (3.10), fazendo-se $u_{n,i} = u_i$ e $d_n(i) = d(i)$ em (4.9) e (4.10). No caso paralelo, fazendo-se $L \to \infty$ a estrutura paralela padrão sem realimentação é recuperada; para $N = 2$ recupera-se o algoritmo CLMS (3.6).

Uma consequência importante da formulação (3.7) e (4.10) é que os algoritmos de reutilização de dados apresentados na Subseção 4.1.1 podem ser interpretados como combinações com supervisores *fixos*. Por exemplo, na combinação (3.7) fazendo-se $N = K$, $L = 1$, $\mu_n = \mu$, e $\lambda_n(i) = 1/N$ recupera-se o algoritmo TRUE-LMS (4.6). De maneira similar, na estrutura incremental com $N = K$, $\mu_n = \mu$, $\lambda_n(i) = 1$, recupera-se o algoritmo $\{\text{LMS}\}^N$ (4.1).

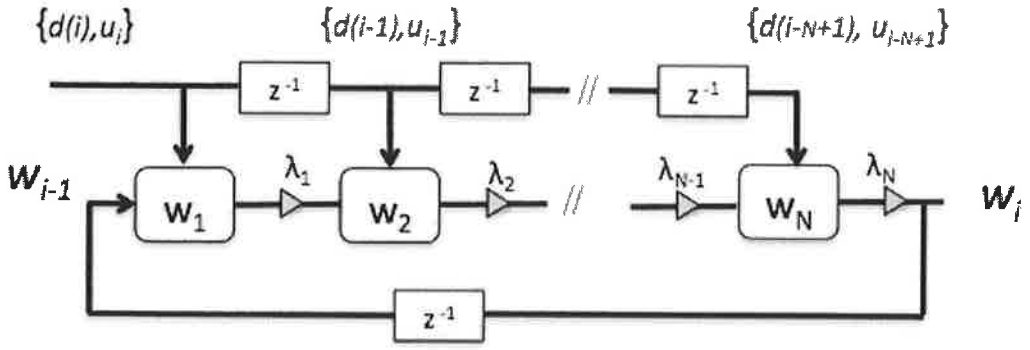


Figura 26: Reuso de dados em estruturas incrementais.

4.1.3 A regra LMS e reuso de dados

A questão subliminar que se põe é: Qual a melhor maneira de se usar a massa de dados disponível? Ou, matematicamente

$$\{d_i, U_i\} \rightarrow p_i \tag{4.11}$$

isto é, como construir uma regra adaptativa $f(\cdot)$ que leve a um vetor de atualização p_i que otimize a adaptação, digamos, em termos de taxa de convergência, sem aumentar proporcionalmente a complexidade computacional? Ou, posto de outra forma: fixando-se a complexidade computacional, como melhorar o desempenho? Nas regras adaptativas clássicas, as duas coisas vem atreladas, não sendo possível aumentar consideravelmente o desempenho sem pagar o preço da (grande) complexidade computacional extra.

A conexão existente entre o APA e o TRUE-LMS sugere uma possibilidade de melhoria deste último algoritmo, na tentativa de torná-lo comparável ao APA [38], [39]. Neste sentido, a função custo do TRUE-LMS é dada por [11]

$$J_{TRUE}(w) = \sum_{k=1}^{K} E|e_k(w)|^2 \tag{4.12}$$

em que $e_k(w) = d(i-k+1) - u_{i-k+1}w$. É um resultado conhecido da teoria

de otimização distribuída que se uma função custo pode ser decomposta como em (4.12), dois métodos de otimização baseados no gradiente-descendente são possíveis.

O método do gradiente total, ou "TRUE-GRADIENT" (que dá origem ao TRUE-LMS) calcula a cada iteração i todos os gradientes intermediários $\nabla J_k(\cdot)$ na mesma estimativa global corrente w_{i-1} [6]. Outro método, conhecido como gradiente incremental, calcula cada gradiente intermediário $\nabla J_k(\cdot)$ em estimativas intermediárias $w_{k-1,i}$, obtidas na função custo intermediária anterior J_{k-1} [5].

No contexto de combinação incremental de FAs, o que se introduz é a possibilidade de balancear desempenho e complexidade computacional, e de forma robusta, com a adoção da regra LMS. Assim, o algoritmo LMS de reuso de dados incremental, denominado DR-$\{\text{LMS}\}^N$, é dado por

$$\begin{aligned} w_{0,i} &\leftarrow w_{i-1} \\ k &= i-(n-1) \mod N \\ w_{n,i} &= w_{n-1,i} + \mu_n u_k^*(d(k) - u_k w_{n-1,i}) \\ w_i &\leftarrow w_{N,i} \end{aligned} \tag{4.13}$$

em que $n = 1, \ldots, N$ e $k = i-(n-1) \mod N$. Nota-se em (4.13) o desacoplamento entre a indexação do FA componente n e o par de dados k. Isto implica que o conjunto de dados pode ser usado várias vezes, fazendo n percorrer a massa de dados novamente quando $n > K$, ciclicamente. Este desacoplamento permite um bom compromisso entre a complexidade computacional permitida, e o desempenho necessário. Permite, também, ajustar o tamanho da janela de observação K, de maneira independente do número de incrementos N, o que é bastante importante em cenários não-estacionários e quando o sistema observa sinais correlacionados.

A recursão em (4.13) envolve o emprego de N filtros LMS, que é um regra ro-

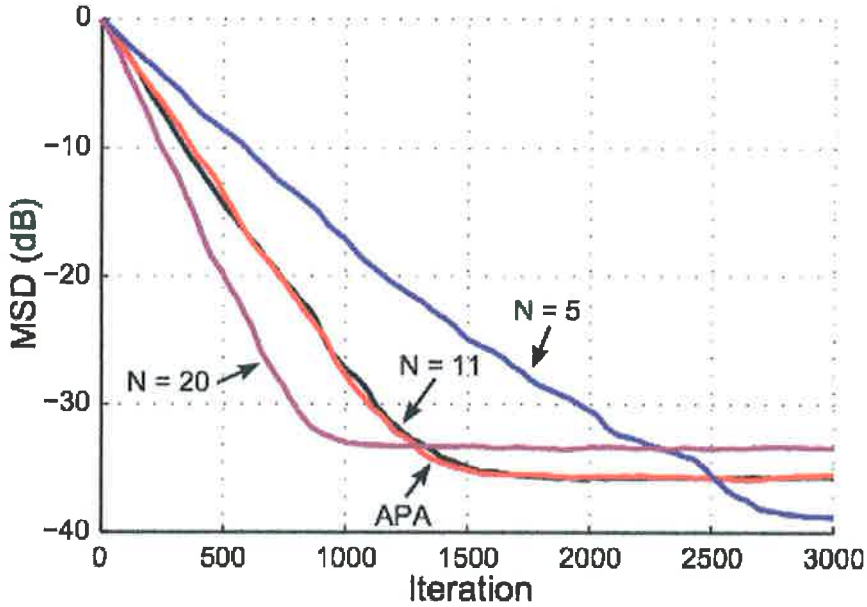


Figura 27: Comparação: APA versus DR-{LMS}^{N}. Sinais brancos em cenário estacionário, com K=10.

busta, levando a uma complexidade computacional global de $O(NM)$. Portanto, requer menos operações que o APA em sua forma tradicional ($O(K^2M)$) [14] ou de versões rápidas do algoritmo ($O(3K^2) + O(KM)$) [40] para uma ampla faixa de N.

Inúmeras simulações mostram que o DR-{LMS}^{N} é capaz de atingir o mesmo desempenho, ou mesmo superar o APA, e operando com menor complexidade computacional. A seguir, o algoritmo DR-{LMS}^{N} é comparado ao APA em diferentes cenários de identificação de sistemas.

As Figuras 27, 28, e 29 apresentam a evolução dos algoritmos em cenários estacionários com $M = 100$, $w^o = \text{col}\{1\}/\sqrt{M}$, $\mu_0 = 0.05$, $\mu_{APA} = \mu_0$, $\mu_n = \mu_0/M\sigma_u^2$ e $\boldsymbol{x}(i)$ e $\boldsymbol{v}(i)$, sequências Gaussianas com média nula e i.i.d., com variâncias $\sigma_x^2 = 1$ e $\sigma_v^2 = 10^{-3}$. Para sinais de entrada brancos, $\boldsymbol{u}(i) = \boldsymbol{x}(i)$, e para entradas correlacionadas, $\boldsymbol{u}(i) = \alpha\,\boldsymbol{u}(i-1) + \sqrt{1-\alpha^2}\boldsymbol{x}(i)$, com $\alpha = 0.95$. A Fig. 30, além do sinal de entrada colorido, também simula um sistema não-estacionário de ordem $M = 60$, modelado como um processo random walk $\boldsymbol{w}_i^o = \boldsymbol{w}_{i-1}^o + \boldsymbol{q}_i$, em que $\boldsymbol{q}_i$ é

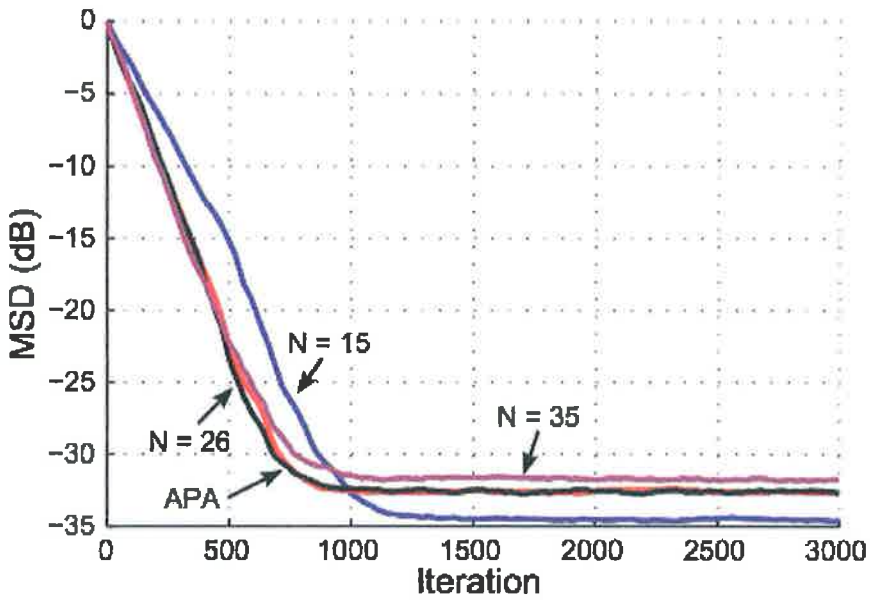


Figura 28: Comparação: APA versus DR-{LMS}^{N}. Sinais brancos em cenário estacionário, com K=20.

Gaussiano, média nula e i.i.d, com matriz de covariância $R_q = \sigma_q^2 I$, e $\sigma_q^2 = 10^{-6}$, conforme já usado no capítulo anterior. Neste caso, $\mu_{APA} = 0.1$ e $\mu_n = 0.003$. Todas as curvas foram geradas promediando 100 experimentos independentes.

O algoritmo DR-{LMS}^{N} é capaz de igualar, ou mesmo superar, o desempenho do APA, e para N relativamente próximo a K, o que significa que sua complexidade é menor que a implementação clássica do APA, ou mesmo que implementações rápidas do mesmo [14, 40].

Outra propriedade interessante do DR-{LMS}^{N} é o balanço entre erro em regime e velocidade de convergência imposto pelo número de FAs componentes N, de forma muito similar ao compromisso existente em termos do passo de cálculo em FAs individuais [14], conforme ilustram as Figs 27 e 28.

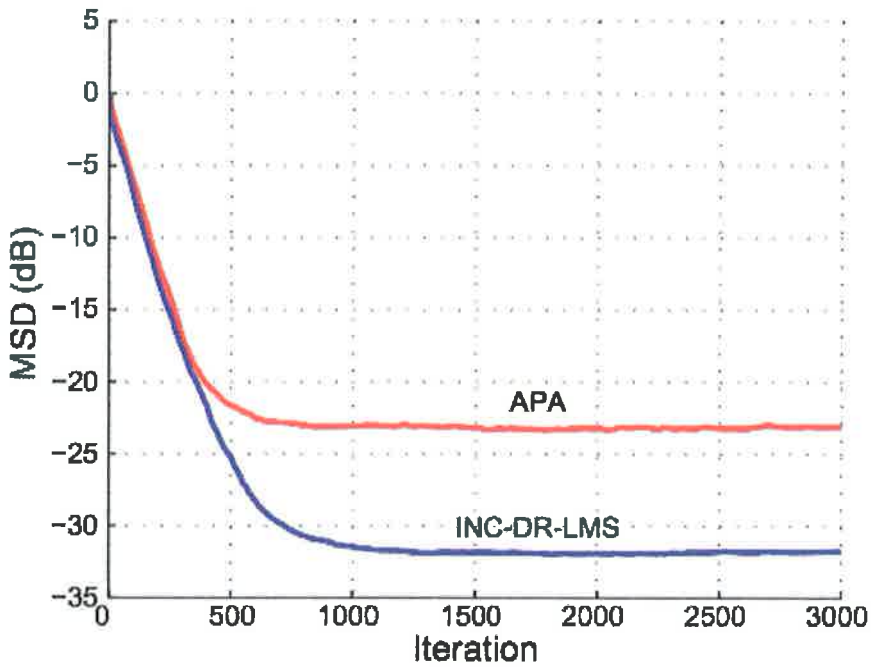


Figura 29: Comparação: APA versus DR-$\{LMS\}^N$ (INC-DR-LMS). Sinais correlacionados em cenário estacionário, K=10, N=20.

4.2 Algoritmo Sign-error LMS Incremental

A seção anterior propõe o algoritmo LMS incremental equipado com reuso de dados como uma maneira computacionalmente mais simples de atingir o nível de desempenho do algoritmo de projeções afins (APA), porém operando uma regra LMS.

Assim, um arranjo incremental de inúmeros FAs LMS observando a mesma massa de dados que o APA, é capaz de apresentar o mesmo (ou melhor) desempenho com um custo computacional menor. Seguindo esta estratégia, coloca-se a questão: Os filtros LMS são os menores quanta de adaptação? "Is there still room at the bottom?" [41] É possível combinar partículas de adaptação menores que o LMS, reduzindo a complexidade, sem prejuízo em desempenho?

Uma solução pioneira no sentido de diminuir a complexidade do já simples LMS, foi o algoritmo sign-error LMS (se-LMS) [14, 15, 42], cuja recursão para

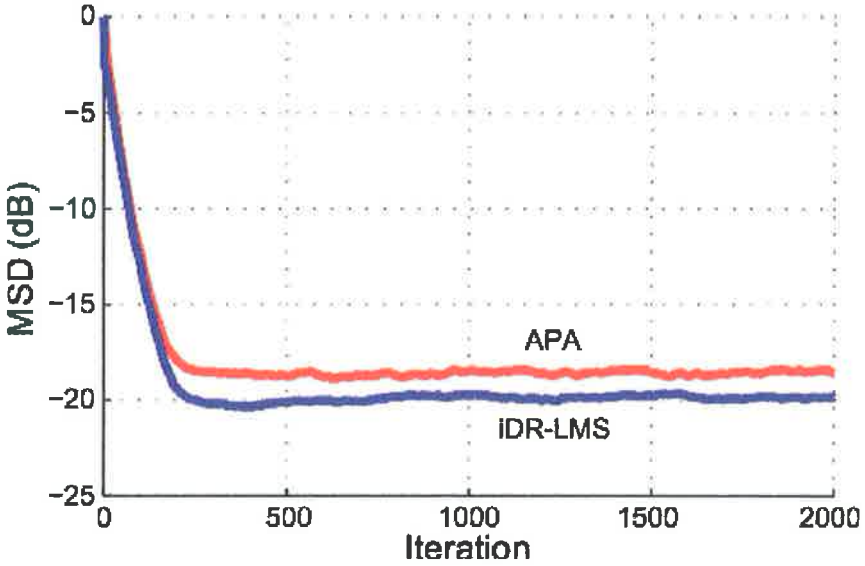


Figura 30: Comparação: APA versus DR-{LMS}^{N} (INC-DR-LMS). Sinais correlacionados em cenário não-estacionário, K=10, N=20.

sinais Reais [1] é

$$w_i = w_{i-1} + \mu u_i^T \operatorname{sign}[e(i)] \tag{4.14}$$

em que $e(i) = d(i) - y(i)$ e $y(i) = u_i w_{i-1}$. O algoritmo se-LMS é capaz de reduzir a complexidade do algoritmo LMS pois elimina a necessidade de se multiplicar o regressor u_i pelo sinal de erro a cada iteração. Além disso, se o passo de cálculo μ for escolhido como uma potência de dois, as multiplicações por μ são implementadas como bit shifts. As únicas multiplicações necessárias são aquelas envolvidas na estimativa $y(i) = u_i w_{i-1}$, que pode ser implementado usando aritmética distribuída [39, 43, 44], tal que apenas uma multiplicação restaria.

Apesar de mais simples, o algoritmo converge de forma mais lenta que o LMS e é enviesado, convergindo para uma região próxima a w^o, dentro de uma hiperesfera cujo raio é proporcional ao passo de cálculo empregado [42]. Em sendo uma variante da regra LMS, o algoritmo se-LMS é um candidato à filtragem incremental. Adotando-se supervisores fixos $\lambda_n = 1$, a recursão incremental para

[1] Nesta seção os sinais são compostos de números Reais.

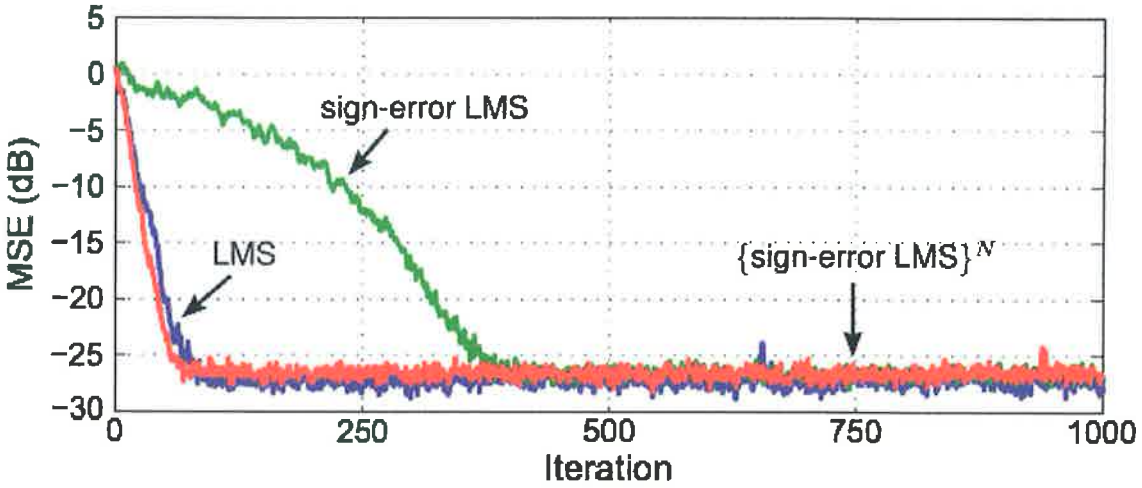


Figura 31: Comparação caso branco estacionário: se-LMS, LMS e $\{\text{se-LMS}\}^N$ em ponto fixo (16 bits). Sign-error: $\mu = 0.004$; LMS: $\mu = 0.083$; e $\{\text{se-LMS}\}^N$: $N = 9$, $\mu_n = 2^{-3-n}$.

N células do algoritmo se-LMS, ou $\{\text{se-LMS}\}^N$, fica

$$\begin{aligned} w_{0,i} &= w_{i-1} \\ e_n(i) &= d(i) - u_i w_{n-1,i} \\ w_{n,i} &= w_{n-1,i} + \mu_n u_i^T \operatorname{sign}[e_n(i)] \\ w_i &= w_{N,i} \end{aligned} \tag{4.15}$$

em que μ_n é uma potência de dois e o algoritmo opera no modo compartilhamento de dados. Implementado desta forma, (4.15) requer NM multiplicações. Contudo, se o regressor u_i apresentar estrutura de deslocamento, como no caso de um filtro transversal, esta combinação pode ser implementada de outra maneira, computacionalmente mais eficiente. Desenvolvendo o erro do $(n+1)$-ésimo FA componente

$$\begin{aligned} e_{n+1}(i) &= d(i) - u_i w_{n,i} \\ &= d(i) - u_i \left\{ w_{n-1,i} + \mu_n u_i^T \operatorname{sign}[e_n(i)] \right\} \\ &= e_n(i) - \mu_n |u_i|^2 \operatorname{sign}[e_n(i)] \end{aligned} \tag{4.16}$$

onde, assumindo estrutura de deslocamento em u_i, tem-se

$$|u_i|^2 = |u_{i-1}|^2 - |u(i-M)|^2 + |u(i)|^2 \tag{4.17}$$

que requer apenas uma multiplicação, uma vez que todos os $|u(j)|^2$, para $j < i$, estão disponíveis de iterações anteriores.

Aplicando-se as recursões (4.16) e (4.17) em (4.15), chega-se a uma recursão computacionalmente mais simples para o algoritmo $\{\text{se-LMS}\}^N$, que requer apenas $M+1$ multiplicações, e este número não depende de N, o número de FAs componentes. O algoritmo completo é reunido no "Algoritmo 1".

Apesar da baixa complexidade, este algoritmo é capaz de atingir o mesmo desempenho do algoritmo LMS, conforme mostra a Fig. 31.

Algorithm 1 $\{\text{se-LMS}\}^N$

$|u_i|^2 = |u_{i-1}|^2 - |u(i-M)|^2 + |u(i)|^2$ ▷ **(1)** ×
$y(i) = u_i w_{i-1}$; $e_1(i) = d(i) - y(i)$ ▷ **(*M*)** ×
$w_{0,i} = w_{i-1}$
for $n = 1, \ldots, N$
  $w_{n,i} = w_{n-1,i} + \mu_n\, u_i^T \operatorname{sign}[e_n(i)]$
  $e_{n+1}(i) = e_n(i) - \mu_n |u_i|^2 \operatorname{sign}[e_n(i)]$
end
$w_i = w_{N,i}$

4.2.1 Uma Implementação Eficiente do Algoritmo NLMS

A análise detalhada do algoritmo $\{\text{se-LMS}\}^N$ é envolvente e ainda encontra-se sob estudo. Contudo, é possível inferir a melhoria em desempenho mostrando que sua recursão se aproxima da recursão de uma algoritmo NLMS.

Agrupando-se as equações em (4.15) chega-se à recursão global do algoritmo

$$w_i = w_{i-1} + u_i^T \sum_{n=1}^{N} \mu_n \operatorname{sign}[e_n(i)]. \tag{4.18}$$

Determinando uma recursão para $\operatorname{sign}[e_n(i)]$ permite colocar a relação acima em uma forma mais conveniente. Assim, aplicando o operador $\operatorname{sign}\{\cdot\}$ em (4.16) e

usando $e_{n-1}(i) = |e_{n-1}(i)| \operatorname{sign}[e_{n-1}(i)]$ leva a

$$\operatorname{sign}[e_n(i)] = \operatorname{sign}[e_{n-1}(i)] \operatorname{sign}\left[\, |e_{n-1}(i)| - \mu_{n-1} \, |u_i|^2 \,\right].$$

Após iterar a equação acima, chega-se a

$$\operatorname{sign}[e_n(i)] = \operatorname{sign}[e(i)] \prod_{k=1}^{n-1} \operatorname{sign}\left[\, |e_k(i)| - \mu_k \, |u_i|^2 \,\right]. \tag{4.19}$$

A Equação (4.19) pode ser usada em (4.18) para obter

$$w_i = w_{i-1} + \bar{\mu}(i) u_i^T \operatorname{sign}[e(i)] \tag{4.20a}$$

$$\bar{\mu}(i) = \mu_1 + \sum_{n=2}^{N} r_n(i) \, \mu_n, \tag{4.20b}$$

em que $r_n(i) = \prod_{k=1}^{n-1} \mathcal{S}_k(i)$ e $\mathcal{S}_k(i) = \operatorname{sign}\left[|e_k(i)| - \mu_k |u_i|^2\right]$.

A recursão global (4.20a) apresenta a forma de um algoritmo se-LMS com um passo variável $\bar{\mu}(i)$ (Variable Step-Size - "VSS"). Assim, está sujeito à mesma condição de convergência que qualquer FA operando sob a regra do sinal do erro do algoritmo LMS [14, 15, 42], ou seja

$$|\widetilde{w}_i|^2 \leq |\widetilde{w}_{i-1}|^2 \Leftrightarrow |e(i)| \geq \bar{\mu}(i) \, |u_i|^2, \tag{4.21}$$

em que $\widetilde{w}_i = w^o - w_i$ é o vetor de erro dos coeficientes. Esta condição pode ser imposta, para algum $\alpha \in (0, 1]$, como

$$\bar{\mu}^o(i) = \alpha \frac{|e(i)|}{|u_i|^2}. \tag{4.22}$$

Substituindo (4.22) em (4.20a) resulta em

$$w_i = w_{i-1} + \alpha \frac{|e(i)|}{|u_i|^2} \, u_i^T \operatorname{sign}[e(i)] = w_{i-1} + \frac{\alpha}{|u_i|^2} \, u_i^T e(i), \tag{4.23}$$

o que é a recursão de um algoritmo NLMS com $\epsilon = 0$ e $\mu = \alpha$. Portanto, o algoritmo VSS sign-error LMS em (4.20a) se torna um algoritmo NLMS se $\bar{\mu}(i) = \bar{\mu}^o(i)$. De fato, é isto que a combinação impõe implicitamente, conforme

mostra subseção a seguir.

4.2.2 Aproximações Sucessivas do Passo de Cálculo

Considerando que o $\{\text{se-LMS}\}^N$ equivale ao VSS sign-error LMS em (4.20), para mostrar que $\{\text{se-LMS}\}^N \rightarrow$ NLMS requer-se apenas mostrar que, para $\bar{\mu}(i)$ dado por (4.20b),

$$|\bar{\mu}(i) - \bar{\mu}^o(i)| \rightarrow 0. \tag{4.24}$$

Então, assumindo, sem perda de generalidade, que $\alpha = 1$ e substituindo (4.20b) e (4.22) em (4.24) leva a

$$\frac{1}{|u_i|^2}\left|\mu_1|u_i|^2 + \sum_{n=2}^{N} r_n(i)\,\mu_n|u_i|^2 - |e(i)|\right|. \tag{4.25}$$

Para $|u_i|^2 \neq 0$, pode-se determinar uma recursão para o valor absoluto em (4.25), o que definindo $r_1(i) = 1$ e para algum k possui a seguinte forma

$$\left|\mu_k|u_i|^2 + r_k(i)\left[\sum_{n=k+1}^{N} r_n(i)\,\mu_n|u_i|^2\right] - |e_k(i)|\right|,$$

o que equivale a

$$\left|r_k(i)\left[\sum_{n=k+1}^{N} r_n(i)\,\mu_n|u_i|^2\right] - \left[|e_k(i)| - \mu_k|u_i|^2\right]\right|. \tag{4.26}$$

Verificando em (4.16) que $|e_{k+1}(i)| = [|e_k(i)| - \mu_k\,|u_i|^2]\,\mathcal{S}_k(i)$ e multiplicando (4.26) por $|\mathcal{S}_k(i)| = 1$, resulta em

$$\left|\mathcal{S}_k(i)r_k(i)\left[\sum_{n=k+1}^{N} r_n(i)\,\mu_n|u_i|^2\right] - |e_{k+1}(i)|\right| \tag{4.27}$$

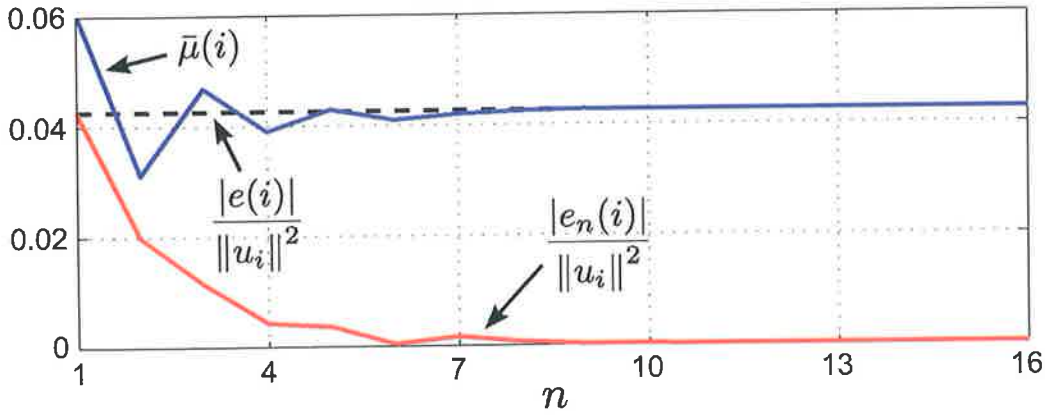


Figura 32: Evolução de uma realização $\bar{\mu}(i)$ e $|e_n(i)|$ durante uma iteração i: $M = 10$, $N = 16$, e $\mu_n = 2^{-4-n}$ (Precisão de MATLAB). $|e_n(i)|$ é normalizado por $|u_i|^2$ para compatibilidade com a escala do gráfico.

Dado que $r_{k+1}(i) = \mathcal{S}_k(i) r_k(i)$ (ver (4.20b)), tem-se

$$\left| \mu_k |u_i|^2 + r_k(i) \left[\sum_{n=k+1}^{N} r_n(i)\, \mu_n |u_i|^2 \right] - |e_k(i)| \right| = \left| \mu_{k+1} |u_i|^2 + r_{k+1}(i) \left[\sum_{n=k+2}^{N} r_n(i)\, \mu_n |u_i|^2 \right] - |e_{k+1}(i)| \right| \quad (4.28)$$

A partir de (4.25) e (4.28) forma-se um processo de indução. É importante notar que para passos μ_k pequenos e decrescentes, vale concluir de (4.16) que $|e_{k+1}| < |e_k|$, tal que $|e_k(i)| \to 0$ conforme se itera (4.28)—isto é, conforme k aumenta—, juntamente com $\mu_k |u_i|^2$. Além disso, a somatória em (4.25) envolve consecutivamente menos termos com o decréscimo de μ_k, de forma que vai para zero. Portanto, (4.24) tende a zero para N grande o suficiente e $\bar{\mu}(i) \to \bar{\mu}^o(i)$.

É possível estender o argumento e mostrar que $\{\text{se-LMS}\}^N$ força assintoticamente o erro de saída a posteriori $e_{N+1}(i) = u_i w_i$ para zero. Portanto, pode ser interpretado como um problema de otimização bem definido, similar ao princípio da perturbação mínima do algoritmo NLMS [14]. Este fenômeno é capturado pela Fig. 32, na qual pode-se constatar que $|e_n(i)|$ tende de fato a zero.

4.2.3 Minimizando o número de componentes

O uso de uma sequência de μ_n pequenos aliado a um número N grande de filtros, conforme abordado na seção anterior, para garantir uma boa aproximação para $\bar{\mu}(i)$ não é prático, do ponto de vista de implementação. Contudo, (4.20b) pode ser visto como a representação truncada de um número real em termos de funções de Rademacher. Desta maneira, encontra-se no espaço gerado pelos passos dos FA componentes. Escolhendo-se μ_n como uma base para o espaço em que $\bar{\mu}(i)$ deve estar minimiza o número de componentes necessários, uma vez que qualquer base é a representação mínima do espaço por ela gerado. No ambiente digital binário, particularmente em sistemas embarcados, uma escolha apropriada seria $\mu_n = 2^{-P-n}$, em que $P \in \mathbb{Z}$, levando a $\bar{\mu}(i) \in (0, 2^{-P+1})$.

Essa estratégia apresenta várias vantagens: (i) Minimiza o número de componentes necessários; (ii) Os μ_n são números em potência de dois, tal que multiplicações são implementadas como deslocamento de bits; e (iii) O fator 2^{-P} age como o fator α in (4.23), o que limita o máximo passo global, aumentando a robustez em relação à representação truncada e ao ruído. A Fig. 32 ilustra a eficácia da estratégia de escolha de μ_n conforme supracitado.

4.2.4 O Algoritmo DR-$\{$LMS$\}^N$ via $\{$se-LMS$\}^N$

Foi mostrado que o algoritmo $\{$se-LMS$\}^N$ é capaz de pelo menos equiparar o desempenho de um FA LMS e com complexidade reduzida. Isto sugere que o *Algoritmo 1* poderia ser concebido como um "módulo" adaptativo simples para construir combinações mais sofisticadas, substituindo os FAs LMS no algoritmo DR-$\{$LMS$\}^N$ (4.13), reduzindo ainda mais a complexidade. Assim, o LMS é substituido por $\{$se-LMS$\}^N$, o qual é implementado L vezes via reuso de dados, gerando o novo algoritmo DR-$\{\{$se-LMS$\}^N\}^L$, em que introduziu-se o parâmetro L para maior flexibilidade. Uma visão sumária do algoritmo resultante é apre-

Algorithm 2 O Algoritmo DR-$\{\{\text{se-LMS}\}^N\}^L$

$|u_i|^2 = |u_{i-1}|^2 - |u(i-M+1)|^2 + |u(i)|^2$ ▷ **(1)** $\times$
$w_{0,i} = w_{i-1}$
for $\ell = 1, \ldots, L$ ▷ DR incremental combination
  $k = (\ell - 1) \mod K$
  $\varepsilon_1(i) = d(i-k) - u_{i-k} w_{\ell-1,i}$ ▷ (M) $\times$
  $w_{\ell,i}^{(0)} = w_{\ell-1,i}$
  for $n = 1, \ldots, N$ ▷ $\{\text{sign-error LMS}\}^N$
    $w_{\ell,i}^{(n)} = w_{\ell,i}^{(n-1)} + \mu_n \, u_{i-k}^T \, \text{sign}[\varepsilon_n(i)]$
    $\varepsilon_{n+1}(i) = \varepsilon_n(i) - \mu_n \, |u_{i-k}|^2 \, \text{sign}[\varepsilon_n(i)]$
  end
  $w_{\ell,i} = w_{\ell,i}^{(N)}$
end
$w_i = w_{L,i}$

sentado em *Algoritmo 2*. Esta implementação requer $LM + 1$ multiplicações.

É possível reduzir a complexidade ainda mais caso se explore a estrutura de deslocamento de u_i, criando buffers para $u(i-p-1)u(i-q-1)$ e $u_{i-p-1}u_{i-q-1}^T$, $p, q < K + M$, e atualizando e_i a cada iteração n. Esta implementação poderia alcançar a complexidade de cerca de $2M + K - 1$ multiplicações; isto está em desenvolvimento no momento. A Tabela 1 compara a complexidade de vários algoritmos adaptativos.

Algumas simulações serão exploradas, comparando a capacidade de rastreamento do algoritmo $\{\text{se-LMS}\}^N$ com o algoritmo se-LMS e o próprio LMS. Também será testado o desempenho do algoritmo DR-$\{\{\text{se-LMS}\}^N\}^L$ e comparado ao APA. Semelhante aos capítulos anteriores, todos os dados para as simulações são obtidos de sequências Gaussianas de média nula independentes e identicamente distribuídas (iid) $\{x(i)\}$ e $\{v(i)\}$, com $\sigma_x^2 = 1$ e $\sigma_v^2 = 10^{-3}$— SNR = 30 dB. Os exemplos que envolvem sinais brancos usam $\boldsymbol{u}(i) = \boldsymbol{x}(i)$, enquanto sinais correlacionados são gerados via $\boldsymbol{u}(i) = \beta \boldsymbol{u}(i-1) + \sqrt{1-\beta^2}\boldsymbol{x}(i)$ com $\beta = 0.95$, o que resulta em sinais altamente correlacionados. Simulações em *ponto fixo* foram realizadas quantizando tanto os dados quanto o resultado de cada operação para uma representação em 16 bits complemento de dois, usando

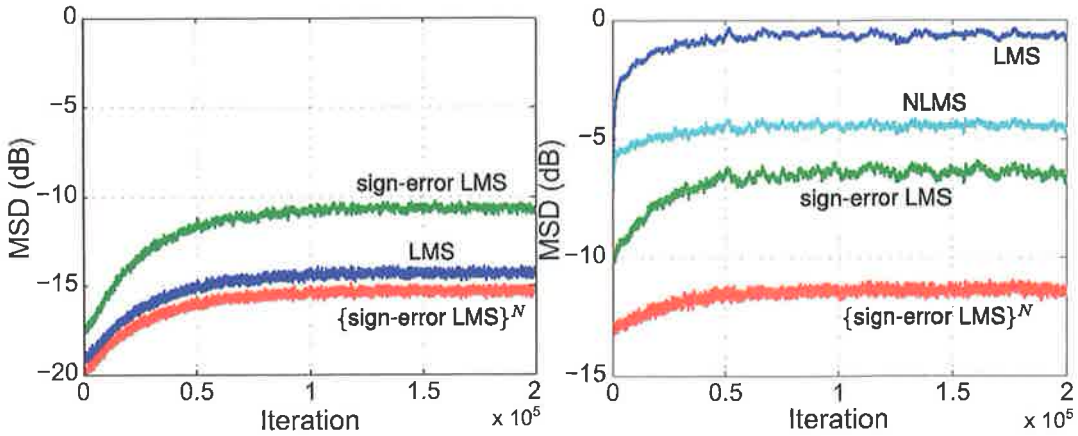
Tabela 1: Multiplicações envolvidas na recursão de diferentes FAs

AF	$\times$
Standard APA	$(K^2+2K)M+K^3+K$
DCD-APA [38]	$M+K^2+3K+2$
DR-$\{\text{LMS}\}^N$ [11]	$(2M+1)N, \quad N \ll K^2$
Algoritmo 2	$LM+1, \quad L \ll K^2$
Algoritmo 2 (Versão Otimizada)	$2M+K-1$

um fator de escalamento de $F = 13$ bits—isto é, 1 bit para o sinal, 2 bits para a parte inteira, e 13 bits para a parte decimal. É empregada saturação para limitar os valores na faixa $[-2^2, 2^2 - 2^{-13}]$. Finalmente, cenários não-estacionários foram simulados usando $w^o(i) = \mathcal{Q}[w^o(i-1) + q_i]$, onde q_i é a realização de um vetor aleatório Gaussiano com matriz de covariância $\sigma_q^2 I$, $\sigma_q^2 = 10^{-4}$, e $\mathcal{Q}$ representa a quantização recém descrita [2]. Curvas para o MSE, o EMSE e o MSD são obtidas pela média estatística de 200 e 800 realizações independentes para cenários estacionários e não-estacionários, respectivamente.

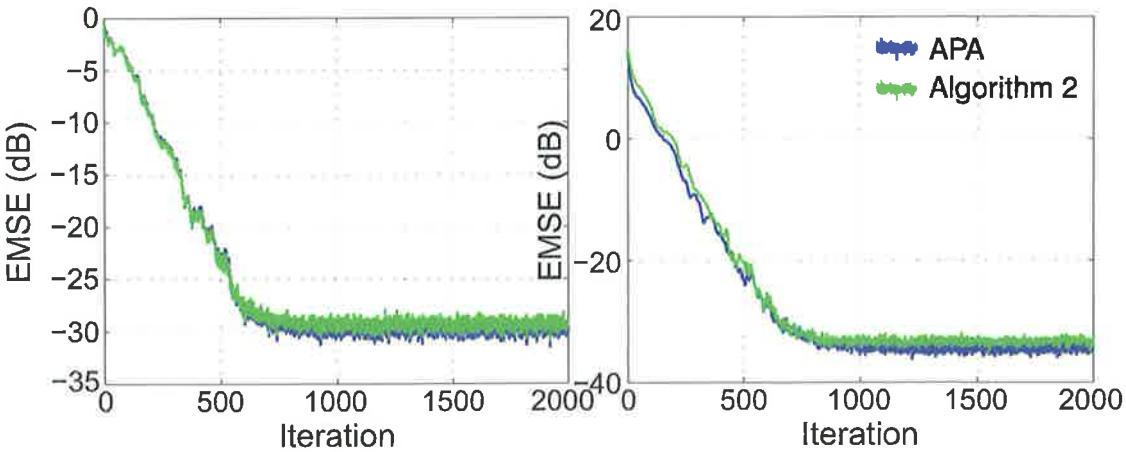
O desempenho do algoritmo $\{\text{se-LMS}\}^N$ em um ambiente estacionário já havia sido mostrado na Fig. 31, que evidenciou a sua capacidade de igualar o algoritmo LMS. Para sistemas não-estacionários, contudo, a combinação supera com grande vantagem outras regras adaptativas, com um ganho de quase 12 dB sobre o LMS para entradas coloridas—Fig. 30. Importante notar que na Fig. 30 o LMS foi projetado com o valor de passo ótimo para cenários não-estacionários [14, Lema 21.1]. Estes resultados claramente justificam o uso desta nova combinação como uma substituta direta para filtros LMS, bem como para combinações mais sofisticadas. De fato, a Fig. 34 mostra que o Algoritmo 2 é capaz de igualar o desempenho do algoritmo APA, e com complexidade menor.

[2] Observa-se o alto grau de não-estacionaridade comparado à faixa de valores usados na literatura: $\sigma_q^2 = [10^{-6}, 10^{-8}]$. w^o também é quantizado para limitar sua variância.



(a) (b)

Figura 33: Cenário não-estacionário com $M = 10$ e quantização ponto fixo (16 bits). (a) Entrada branca. **se-LMS**: $\mu = 2^{-7}$; **LMS**: $\mu = \mu^o = 0.0916$; e **{se-LMS}**^{N}: $N = 5$ and $\mu_n = 2^{-5-n}$. (b) Entrada correlacionada. **se-LMS**: $\mu = 2^{-6}$; **LMS**: $\mu = 0.02$; e **NLMS**: $\mu = 0.09$ e $\epsilon = 2^{-F+1}$; e **{se-LMS}**^{N}: $N = 5$ e $\mu_n = 2^{-5-n}$.



(a) (b)

Figura 34: APA e Algoritmo 2 em cenário estacionário—$M = 100$ e $K = 10$ (simulação em precisão de MATLAB devido à sensibilidade do algoritmo APA a palavras de comprimento reduzido). (a) Entrada branca. **APA**: $\mu = 0.3$ e $\epsilon = 10^{-6}$; e **DR-{{se-LMS}**^{N}**}**^{L}: $L = 10$, $N = 20$, e $\mu_n = 2^{-n}$. (b) Entrada Correlacionada. **APA**: $\mu = 0.06$ e $\epsilon = 10^{-6}$; e **DR-{{se-LMS}**^{N}**}**^{L}: $L = 30$, $N = 10$, and $\mu_n = 2^{-15-n}$.