DANIEL GILIO TIGLEA

Um algoritmo adaptativo de baixo custo computacional para amostragem e censura em redes de difusão

São Paulo
2020

DANIEL GILIO TIGLEA

Um algoritmo adaptativo de baixo custo computacional para amostragem e censura em redes de difusão

Dissertação apresentada à Escola Politécnica da Universidade de São Paulo para obtenção do título de Mestre em Ciências.

São Paulo
2020

DANIEL GILIO TIGLEA

Um algoritmo adaptativo de baixo custo computacional para amostragem e censura em redes de difusão

Dissertação apresentada à Escola Politécnica da Universidade de São Paulo para obtenção do título de Mestre em Ciências.

Área de concentração:

Engenharia de Sistemas Eletrônicos

Orientador:

Prof. Dr. Magno Teófilo Madeira da Silva

Coorientador:

Prof. Dr. Renato Candido

São Paulo
2020

Este exemplar foi revisado e corrigido em relação à versão original, sob responsabilidade única do autor e com a anuência de seu orientador.

São Paulo, 10 de julho de 2020

Assinatura do autor: Daniel Gilio Tiglea

Assinatura do orientador: Magno T. M. Silva

Catalogação-na-publicação

Tiglea, Daniel

Um algoritmo adaptativo de baixo custo computacional para amostragem e censura em redes de difusão / D. Tiglea, R. Candido, M. Silva – versão corr. - São Paulo, 2020.

98 p.

Dissertação (Mestrado) - Escola Politécnica da Universidade de São Paulo. Departamento de Engenharia de Sistemas Eletrônicos.

1.Sistemas distribuídos 2.Filtros elétricos adaptativos 3.Redes de computadores (Sensores) 4.Processamento de sinais (Amostragem) 5.Amostragem em soluções distribuídas I.Universidade de São Paulo. Escola Politécnica. Departamento de Engenharia de Sistemas Eletrônicos II.t. III.Candido, Renato IV.Silva, Magno

Aos meus pais,
Paulo e Virgínia

AGRADECIMENTOS

Agradeço ao meu orientador, Prof. Dr. Magno Teófilo Madeira da Silva, pela constante dedicação, apoio e motivação. Obrigado por estar sempre presente, da concepção deste trabalho à sua conclusão.

Ao meu coorientador, Prof. Dr. Renato Candido, que foi essencial para que este trabalho se tornasse o que é. Obrigado pelas conversas estimulantes, pelas observações sempre pertinentes e por todo o auxílio ao longo do caminho.

Aos membros da banca examinadora, Prof. Dr. Denis Fantinato e Prof. Dr. Wallace Martins, pela leitura cuidadosa da dissertação e pelas contribuições enriquecedoras para este trabalho.

Aos meus pais, Paulo e Virgínia, por todo o amor, dedicação e paciência e por despertarem em mim, desde cedo, o amor por essa coisa que chamamos “ciência”.

À minha namorada, Bruna, por todo o carinho, compreensão e companheirismo, por todos os risos compartilhados e, sobretudo, pelo amor.

A todos os meus amigos, que ajudaram imensamente (e talvez sem saber) na elaboração deste trabalho ao tornarem a vida mais leve.

A todos os professores que foram parte da minha formação, por contribuírem para o meu crescimento pessoal e acadêmico.

À Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) e ao Conselho Nacional de Pesquisa e Desenvolvimento (CNPq), pelo financiamento.

RESUMO

TIGLEA, Daniel Gilio. Um algoritmo adaptativo de baixo custo computacional para amostragem e censura em redes de difusão / D. G. Tiglea. 97f. São Paulo, 2020

Nos últimos anos, redes de difusão adaptativas e filtros adaptativos baseados em grafos se tornaram tópicos de forte interesse na comunidade de processamento de sinais. As redes de difusão adaptativas se consolidaram na literatura como ferramentas interessantes para o processamento distribuído de sinais, apresentando vantagens em relação a soluções centralizadas e a outras técnicas de difusão. Os filtros adaptativos baseados em grafos, por sua vez, vêm ganhando notoriedade por sua capacidade de lidar com situações em que há grandes quantidades de dados relacionados entre si por meio de estruturas irregulares. Em ambos os casos, foram propostas técnicas para reduzir a quantidade de informação medida e transmitida ao longo das redes, o que possibilita reduzir o custo computacional e o consumo energético. Tais técnicas em geral afetam o desempenho das soluções originais, mas são importantes por prolongar a vida útil das redes. Neste trabalho, é proposto um mecanismo adaptativo de amostragem para soluções adaptativas difusas e baseadas em grafos. O algoritmo de amostragem proposto utiliza mais nós quando a magnitude do erro ao longo da rede é elevada e menos nós caso contrário. Dessa forma, alcança-se uma redução significativa em termos de custo computacional ao mesmo tempo em que o impacto no desempenho é mitigado. Mostra-se ainda que, com uma pequena modificação, ele pode ser utilizado para reduzir a quantidade de transmissões entre nós, possibilitando uma economia em termos energéticos. Além disso, é apresentada uma análise teórica acerca do mecanismo proposto, que possibilita uma melhor compreensão do seu funcionamento e permite escolhas mais embasadas para os seus parâmetros.

Palavras-chave: Redes de difusão adaptativas. Estimação distribuída. Processamento de sinais em grafos. Filtragem em grafos. Amostragem em grafos. Eficiência energética. Combinação convexa.

ABSTRACT

TIGLEA, Daniel Gilio. A low-cost algorithm for adaptive sampling and censoring in diffusion networks / D. G. Tiglea. 97f. São Paulo, 2020

In recent years, diffuse adaptive networks and graph adaptive filters have become topics of strong interest in the signal processing community. Diffuse adaptive networks have consolidated themselves in the literature as interesting tools for distributed signal processing, presenting advantages over centralized solutions and other diffusion techniques. Graph adaptive filters, in turn, have attracted attention for their ability to deal with situations in which there are large amounts of data that are related through irregular structures. In both cases, techniques have been proposed to reduce the amount of information measured and transmitted over the networks, which enables a reduction in the computational cost and in the energy consumption. Such techniques usually affect the performance of the original solutions, but are important for extending the network lifetime. In this work, we propose an adaptive sampling mechanism for distributed adaptive solutions and graph adaptive filters. The proposed sampling algorithm uses more nodes when the magnitude of the error throughout the network is high, and less nodes otherwise. Thus, a significant reduction in computational cost is achieved while the impact on performance is mitigated. It is also shown that, with a small modification, the proposed sampling mechanism can be used to reduce the number of transmissions between nodes, enabling a reduction in energy consumption. Furthermore, we conduct a theoretical analysis of the proposed mechanism, which enables a better understanding of its functioning and allows more suitable choices for its parameters.

Keywords: Adaptive networks. Distributed estimation. Graph signal processing. Graph filtering. Sampling on graphs. Energy efficiency. Convex combination.

LISTA DE ABREVIAÇÕES

ACW	*adaptive combination weights* (pesos de combinação adaptativos)
APA	*affine projection algorithm* (algoritmo de projeções afins)
AS-dNLMS	*adaptive-sampling diffuse NLMS* (NLMS difuso com amostragem adaptativa)
ASC-dNLMS	*adaptive-sampling-and-censoring diffuse NLMS* (NLMS difuso com amostragem e censura adaptativas)
ATC	*adapt-then-combine* (adapte e então combine)
CTA	*combine-then-adapt* (combine e então adapte)
dNLMS	*diffuse NLMS* (NLMS difuso)
FIR	*finite impulse response* (resposta ao pulso unitário finita)
GFT	*graph Fourier transform* (transformada de Fourier para grafos)
GSP	*graph signal processing* (processamento de sinais em grafos)
i.i.d.	independente e identicamente distribuído
LMS	*least mean squares*
LS	*least squares* (mínimos quadrados)
MSE	*mean square error* (erro quadrático médio)
MSD	*mean square deviation* (desvio quadrático médio)
NLMS	*normalized least mean squares*
RLS	*recursive least squares* (mínimos quadrados recursivos)
SNR	*signal-to-noise ratio* (razão sinal-ruído)

LISTA DE SÍMBOLOS

Notação. Letras em fontes convencionais denotam escalares, letras minúsculas em negrito denotam vetores e letras maiúsculas em negrito denotam matrizes. O k-ésimo elemento do vetor $\mathbf{x}$ é denotado por $[\mathbf{x}]_k$ ou por x_k, quando isto não causar confusão. Cabe observar que, neste trabalho, consideram-se apenas sinais reais.

Símbolos gerais

$(\cdot)^{\mathrm{T}}$	transposição de vetores ou matrizes
$(\cdot)^{\mathrm{H}}$	transposição Hermitiana de vetores ou matrizes
$\mathrm{E}\{\cdot\}$	operador esperança matemática
$\sum$	somatório
$\|\cdot\|$	norma euclidiana de um vetor
$\lvert\cdot\rvert$	cardinalidade de um conjunto ou módulo de um escalar
$\mathbf{I}$	matriz identidade
∇^2	operador Laplaciano
z^{-1}	operador de atraso unitário
σ_x^2	variância do sinal x
$\widehat{x}$	valor estimado de x
n	instante de tempo
min	mínimo de uma variável ou função
max	máximo de uma variável ou função
Δ	diferença ou variação
ω	frequência angular
φ	fase de um sinal senoidal
$\mathrm{sgm}(\cdot)$	função sigmoidal

$\mathrm{col}\{\cdot\}$	vetor coluna contendo os elementos especificados no seu argumento
$\mathrm{diag}\{\cdot\}$	matriz diagonal contendo os elementos do seu argumento na diagonal principal

Processamento distribuído de sinais e filtragem adaptativa baseada em grafos

e	erro de estimação
d	sinal desejado
y	sinal de saída do filtro
η	ruído aditivo
u	sinal de entrada do filtro
μ	passo de adaptação de algoritmos adaptativos
$\mathbf{w}$	vetor de coeficientes do filtro
$\mathbf{w}^{\mathrm{o}}$	vetor de coeficientes do sistema desconhecido
δ	pequena constante positiva utilizada para evitar divisão por zero
M	número de coeficientes do filtro
J	função custo
V	número de nós de um grafo
$\mathcal{N}_k$	conjunto composto pela união do nó k com a sua vizinhança
$\boldsymbol{\psi}$	estimativa local do algoritmo dNLMS
$\{c_{ik}\}$	pesos de combinação das estimativas locais no nó k
$\mathbf{A}$	matriz de adjacência de um grafo
$\widehat{\sigma}_{jk}$	estimativa calculada pelo nó k da variância do ruído no nó j
ν	constante utilizada pelo algoritmo ACW para filtrar as estimativas $\widehat{\sigma}_{jk}$ para cada $j \in \mathcal{N}_k$
δ_c	pequena constante positiva utilizada para evitar divisão por zero na atualização dos pesos de combinação pelo algoritmo ACW
$\boldsymbol{\Upsilon}$	matriz de grau de um grafo

$\boldsymbol{\mathcal{L}}$	operador Laplaciano para grafos
$\boldsymbol{\Omega}$	matriz de autovetores do operador Laplaciano para grafos
$\boldsymbol{\Lambda}$	matriz de autovalores do operador Laplaciano para grafos
$\mathbf{u}$	sinal definido sobre um grafo
$\mathbf{s}$	transformada de Fourier estendida a grafos de um sinal sobre um grafo
$\mathbf{H}$	filtro linear e invariante ao deslocamento para grafos
L	ordem do filtro $\mathbf{H}$ que processa o sinal sobre o grafo
$\mathbf{h}$	saída de um filtro linear e invariante ao deslocamento
$\mathbf{d}$	vetor com amostras do sinal desejado em cada nó da rede ou grafo
$\overline{t}_k$	variável utilizada para decidir se o nó k deve ser amostrado ou não no instante n
t_k	variável auxiliar utilizada para decidir o valor de $\overline{t}_k$
$J_{t,k}$	função custo utilizada na atualização de t_k
β	parâmetro dos algoritmos AS-dNLMS e ASC-dNLMS
α_k	variável auxiliar utilizada na adaptação de t_k
α^+	maior valor positivo que α_k pode assumir
$\phi(\cdot)$	função que relaciona α_k a t_k
$\phi'(\cdot)$	derivada da função ϕ em relação a α_k
μ_t	passo de adaptação do mecanismo de amostragem do algoritmo AS-dNLMS
ε_k	último valor do sinal de erro medido no nó k
β_r	relação entre β e $\sigma^2_{\max}$
$\sigma^2_{\min}$	menor variância do ruído observada na rede
$\sigma^2_{\max}$	máxima variância do ruído observada na rede
V_t	número de nós amostrados na rede
p_{t_k}	probabilidade do nó k ser amostrado em regime permanente

θ_k	número de iterações por ciclo do algoritmo de amostragem em que o nó k permanece amostrado
$\overline{\theta}_k$	número de iterações por ciclo do algoritmo de amostragem em que o nó k permanece sem ser amostrado
$\oplus$	custo computacional – adição
$\otimes$	custo computacional – multiplicação
$\mathbf{T}$	operador de amostragem
$f_s(\cdot)$	função penalizadora da não-esparsidade
κ	peso da penalização da não-esparsidade na função custo do algoritmo de (LORENZO et al., 2016)
$g_\gamma(\cdot)$	função esparsificadora
γ	parâmetro de esparsificação
$\widetilde{\mathbf{X}}$	matriz de entrada da versão centralizada do algoritmo de (NASSIF et al., 2017)
$\boldsymbol{\zeta}$	estimativa local da transformada de Fourier estendida a grafos do algoritmo de (LORENZO et al., 2017a)
$\boldsymbol{\tau}_k^{\mathrm{H}}$	k-ésima linha da matriz Ω
$\widetilde{\mathbf{x}}_k$	vetor de entrada local da versão distribuída do algoritmo de (NASSIF et al., 2017) no nó k

SUMÁRIO

1 INTRODUÇÃO

Neste capítulo são apresentados o contexto, as contribuições e a estrutura do presente trabalho. Na Seção 1.1, é feita uma introdução qualitativa ao processamento distribuído de sinais e à filtragem adaptativa baseada em grafos. Além disso, são expostas as principais motivações para a realização deste trabalho. Os objetivos e a justificativa são apresentados nas Seções 1.2 e 1.3, respectivamente. Finalmente, as Seções 1.4 e 1.5 expõem, respectivamente, as contribuições e a organização da dissertação.

1.1 Problemática e motivação

Ao longo da última década, as redes de difusão adaptativas se consolidaram como uma ferramenta interessante para o processamento distribuído de sinais. Comparado às abordagens centralizadas, que exigem que uma unidade central receba e processe os dados de toda a rede, esse tipo de solução apresenta melhor escalabilidade, autonomia e flexibilidade (SAYED, 2014; LOPES; SAYED, 2008b; CATTIVELLI; SAYED, 2009; CATTIVELLI; LOPES; SAYED, 2008a; LI; CHAMBERS, 2009). Consequentemente, redes de difusão adaptativas são consideradas soluções eficazes em várias aplicações, como localização e rastreamento de alvos (SAYED, 2014), sensoreamento espectral em redes móveis (LORENZO; BARBAROSSA; SAYED, 2013; SAYED, 2014), aplicações médicas (AKYILDIZ et al., 2002), entre outras.

Essas redes consistem em um conjunto de **agentes** ou **nós** conectados, capazes de coletar dados e realizar cálculos localmente e de se comunicar com outros agentes próximos, chamados de **vizinhos**. O objetivo da rede como um todo é estimar um vetor de parâmetros de interesse. Para isso, cada nó calcula sua própria estimativa **local** na chamada **etapa de adaptação**. Na **etapa de combinação**, por sua vez, os nós vizinhos cooperam para alcançar uma estimativa **global** do vetor de interesse. A ordem em que essas etapas são executadas leva a dois esquemas possíveis: as estratégias do tipo "adaptar e então combinar" (*adapt-then-combine* – ATC) e as do tipo "combinar e então adaptar" (*combine-then-adapt* – CTA). Com essas duas etapas, a ideia consiste em estimar os parâmetros de interesse sem que haja uma unidade central de processamento (SAYED, 2014; LOPES; SAYED, 2008b; CATTIVELLI; SAYED, 2009; CATTIVELLI; LOPES; SAYED, 2008a; LI; CHAMBERS, 2009; LORENZO; BARBAROSSA;

SAYED, 2013; AKYILDIZ et al., 2002; FERNANDEZ-BES et al., 2017; NASSIF et al., 2018; HUA et al., 2018).

Mais recentemente, o processamento de sinais em grafos (*graph signal processing* – GSP) e a filtragem adaptativa baseada em grafos se tornaram tópicos de forte interesse na comunidade de processamento de sinais (SANDRYHAILA; MOURA, 2013b; SHUMAN et al., 2013b; CHEN et al., 2015; ANIS; GADDE; ORTEGA, 2016; TSITSVERO; BARBAROSSA; LORENZO, 2016; LORENZO et al., 2018), particularmente no campo das redes de difusão (NASSIF et al., 2018; HUA et al., 2018; LORENZO et al., 2017a). Em comparação com o processamento distribuído de sinais "clássico", os filtros adaptativos baseados em grafos incorporam informações da topologia da rede na etapa de adaptação, o que é útil em situações em que essa topologia desempenha um papel importante na dinâmica dos sinais de interesse (NASSIF et al., 2018; HUA et al., 2018). Esse é o caso em muitas aplicações que apresentam estruturas em formato de rede e que vêm ganhando proeminência nos últimos anos, como as redes elétricas inteligentes (em inglês, *smart grids*), a internet das coisas, redes de transporte e comunicação, entre outras (NASSIF et al., 2018; HUA et al., 2018; LORENZO et al., 2017a; SANDRYHAILA; MOURA, 2013b; SHUMAN et al., 2013b; CHEN et al., 2015; ANIS; GADDE; ORTEGA, 2016; TSITSVERO; BARBAROSSA; LORENZO, 2016; LORENZO et al., 2018). Nesses casos, grafos surgem como ferramentas de modelagem interessantes, pois são convenientes para a representação de estruturas irregulares como, por exemplo, redes formadas por sensores espalhados de maneira arbitrária em uma determinada região.

Ao se implementar soluções com processamento distribuído, muitas vezes é desejável restringir o número de medições nos sensores e a quantidade de informação transmitida ao longo da rede. Por exemplo, quando essas estratégias são empregadas em redes de sensores sem fio, o consumo de energia é frequentemente a maior restrição ao funcionamento da rede (TAKAHASHI; YAMADA, 2010; ARROYO-VALLES; MALEKI; LEUS, 2013; FERNANDEZ-BES et al., 2015). Consequentemente, foram propostas na literatura várias soluções para reduzir o consumo energético associado à comunicação entre nós. Algumas dessas soluções buscam reduzir a quantidade de informação enviada em cada transmissão (ARABLOUEI et al., 2013; CHOUVARDAS; SLAVAKIS; THEODORIDIS, 2013), enquanto outras desligam conexões de acordo com certas políticas de transmissões seletivas (LOPES; SAYED, 2008a; TAKAHASHI; YAMADA, 2010; ZHAO; SAYED, 2012; XU; DE LAMARE; POOR, 2015). Por fim, existem as chamadas técnicas de censura (*censoring*, em inglês). Esses métodos procuram evitar que

os nós transmitam suas informações para quaisquer vizinhos, permitindo assim que eles desliguem seus transmissores e economizem mais energia (ARROYO-VALLES; MALEKI; LEUS, 2013; GHAREHSHIRAN; KRISHNAMURTHY; YIN, 2013; FERNANDEZ-BES et al., 2015; YANG et al., 2018; BERBERIDIS et al., 2015).

Além disso, em certas situações, o custo de se medir e processar os dados disponíveis em cada nó a cada iteração é proibitivamente alto. Nesses casos, faz-se necessário empregar alguma técnica de amostragem (LORENZO et al., 2018; LORENZO et al., 2017a). A amostragem pode reduzir significativamente o custo computacional e de memória associado ao aprendizado, mas também pode impactar negativamente o desempenho do algoritmo. Para ilustrar isso, na Figura 1 são mostrados resultados de simulações obtidos em um ambiente estacionário. Considera-se uma rede de 20 nós utilizando o algoritmo dNLMS (*diffuse normalized least-mean-squares*) ATC (SAYED, 2014; LOPES; SAYED, 2008b; CATTIVELLI; SAYED, 2009) com uma técnica que, a cada iteração, amostra V_t nós escolhidos aleatoriamente. São apresentados resultados para $V_t \in \{5, 10, 15, 20\}$. O cenário de simulação é descrito com mais detalhes na Seção 3.3. Como indicador de desempenho, adota-se o desvio médio quadrático da rede (*mean-square-deviation* – MSD). Para avaliar o custo computacional, mostra-se o número médio de somas e multiplicações efetuadas por iteração. Os resultados são apresentados como porcentagens dos dados obtidos com todos os $V_t = 20$ nós amostrados. Observa-se que, quanto menos nós amostrados, menor é o custo computacional. Entretanto, há claramente um impacto na taxa de convergência, que se torna cada vez mais lenta conforme se reduz o número de nós amostrados. Isso se deve ao fato de que, ao amostrar menos nós por iteração, reduz-se a taxa com que informações novas são introduzidas na rede. Consequentemente, a convergência durante o transitório é afetada. Entretanto, uma vez que o algoritmo atinge o regime permanente, a introdução de novas informações não leva a uma melhora no desempenho em um ambiente estacionário.

Diante desse experimento simples, a indagação que surge é se seria possível projetar um mecanismo de amostragem mais "inteligente", que amostre mais nós enquanto o erro for alto em magnitude e menos nós caso contrário. Neste trabalho, é apresentado um algoritmo que atende a esses requisitos. Mostra-se que ele é capaz de reduzir significativamente o custo computacional durante o regime permanente, ao mesmo tempo em que preserva a taxa de convergência do algoritmo com todos os nós amostrados. Além disso, com uma ligeira modificação, ele também pode ser empregado como uma técnica de censura. Nesse caso, simulações computacionais

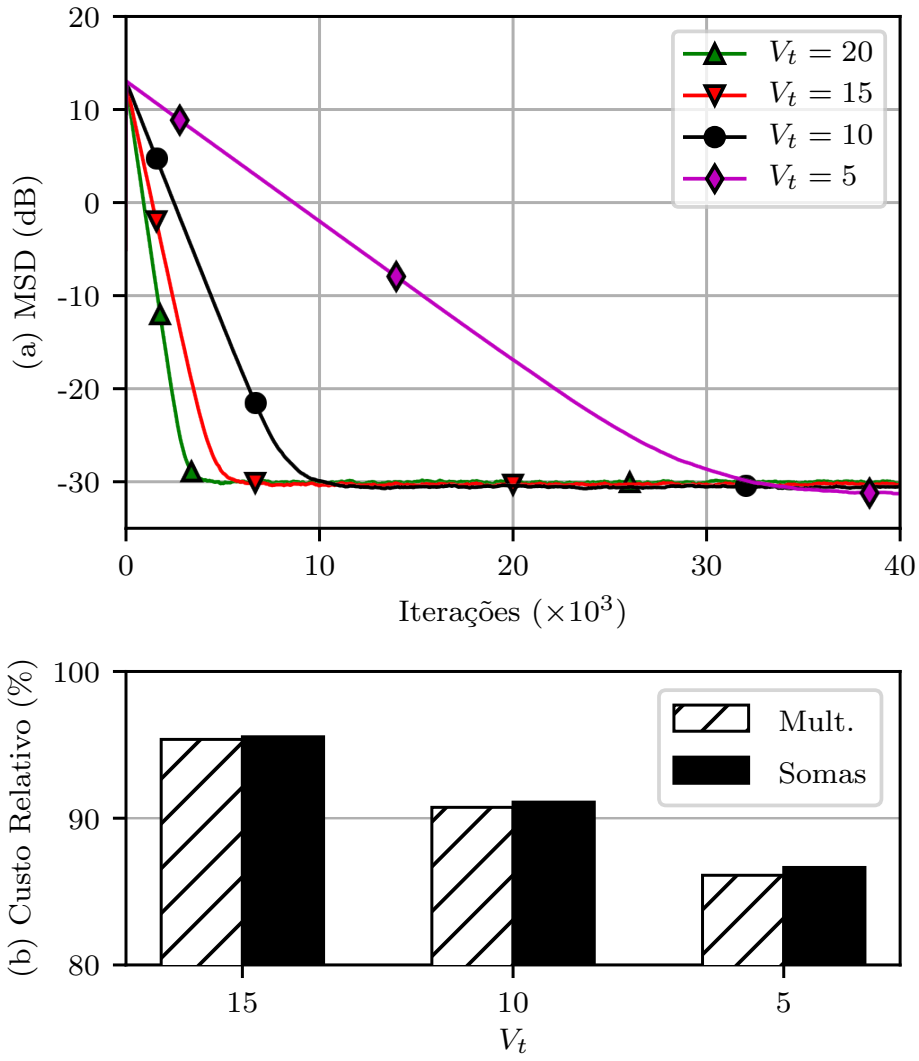


Figura 1: Resultados de simulação para uma rede com 20 nós usando o algoritmo dNLMS ATC com V_t nós amostrados por iteração. O cenário de simulação é descrito na Seção 3.3. (a) Curvas de MSD e (b) Número relativo de somas e multiplicações em comparação com o caso em que todos os $V_t = 20$ nós são amostrados.

Fonte: Autor.

mostram que ele é capaz de economizar mais energia do que outras técnicas apresentadas na literatura, ao mesmo tempo em que apresenta um impacto menor em termos do MSD.

No presente trabalho, o algoritmo proposto é testado em diferentes cenários de simulação. Além disso, é apresentada uma análise teórica que permite uma melhor compreensão do funcionamento do mecanismo de amostragem e/ou censura, ao mesmo tempo em que auxilia na escolha de seus parâmetros.

1.2 Objetivos

Os principais objetivos deste trabalho consistem em:

1. realizar uma revisão bibliográfica acerca das redes de difusão adaptativas que incorporam ou não o processamento de sinas em grafos na etapa de adaptação;

2. propor um mecanismo de amostragem para algoritmos adaptativos difusos, visando a uma redução no custo computacional e/ou no número de transmissões entre nós;

3. testar o mecanismo proposto, considerando diferentes cenários de simulação;

4. realizar uma análise teórica acerca do algoritmo proposto e validá-la por meio de simulações computacionais.

1.3 Justificativa

Devido às suas vantagens em relação às soluções centralizadas e a outros esquemas distribuídos, as redes de difusão adaptativas se consolidaram na literatura como soluções robustas e atrativas para o processamento distribuído de sinais (SAYED, 2014; FERNANDEZ-BES et al., 2017). Similarmente, nos últimos anos, têm sido propostos na literatura algoritmos adaptativos distribuídos que incorporam elementos da teoria dos grafos (LUO et al., 2017; LI et al., 2018; LORENZO; BANELLI; BARBAROSSA, 2017; BUI; RAVI; RAMAVAJJALA, 2017; LORENZO et al., 2016; LORENZO et al., 2018; NIGRIS et al., 2017). Trata-se de uma área nova, com grande potencial para aplicações práticas e em que ainda há muito a ser explorado.

Em ambos os casos, o custo computacional e/ou o consumo energético das redes adaptativas podem ser proibitivamente altos, inviabilizando o uso dessas soluções. Para lidar com isso, mecanismos de amostragem e censura vêm sendo estudados pela comunidade de processamento de sinais. De maneira geral, procura-se obter técnicas que reduzam ao máximo a quantidade de informação coletada e/ou transmitida pela rede e que, ao mesmo tempo, tenham o menor impacto possível no desempenho. Entretanto, verifica-se que as várias abordagens apresentadas na literatura, embora interessantes, geralmente afetam de maneira perceptível a taxa de convergência e/ou o desempenho em regime permanente das soluções para as quais se destinam (ARROYO-VALLES; MALEKI; LEUS, 2013; GHAREHSHIRAN; KRISHNAMURTHY; YIN, 2013; FERNANDEZ-BES et al., 2015; LORENZO et al., 2018; LORENZO et al., 2017a).

A obtenção de uma técnica que leve a uma economia energética e de custo computacional preservando-se o desempenho é, portanto, uma questão relevante. Com isso, seria possível viabilizar a operação das redes sem sacrificar as características que as tornam tão atrativas.

Considerando a crescente disponibilidade de sensores de baixo custo e os avanços constantes na área de redes de comunicação, a tendência é que tais técnicas adquiram importância cada vez maior.

1.4 Contribuições

A partir dos resultados deste trabalho, foram publicados dois artigos nos anais de simpósios nacionais [CN-1, CN-2] e um artigo nos anais de um congresso internacional [CI-1]. Além disso, há dois artigos submetidos que se encontram em fase de revisão: um de congresso internacional [CI-2] e um de periódico internacional [PI]. Esses trabalhos estão listados a seguir.

CN-1 D. G. Tiglea, R. Candido, e M. T. M. Silva. Uma comparação entre técnicas adaptativas baseadas em grafos. In *Anais do Simpósio Brasileiro de Telecomunicações e Processamento de Sinais (SBrT'18)*. Campina Grande, Set. 2018, p. 934-938. DOI 10.14209/sbrt.2018.332.

CN-2 D. G. Tiglea, R. Candido, e M. T. M. Silva. Amostragem adaptativa aplicada a um algoritmo difuso voltado a grafos. In *Anais Simpósio Brasileiro de Telecomunicações e Processamento de Sinais* (SBrT'19). Petrópolis, Set. 2019. DOI 10.14209/SBRT.2019.1570552504.

CI-1 D. G. Tiglea, R. Candido, e M. T. M. Silva. An adaptive sampling technique for graph diffusion LMS Algorithm. In *Proceedings of 2019 European Signal Processing Conference* (EUSIPCO'19). A Coruña, Espanha, Set. 2019. DOI 10.23919/EUSIPCO.2019.8902727.

CI-2 D. G. Tiglea, R. Candido, e M. T. M. Silva. A sampling algorithm for diffusion networks. Submetido ao *2020 European Signal Processing Conference* (EUSIPCO'20). A ser realizado em Amsterdã, Holanda, Jan. 2021.

PI D. G. Tiglea, R. Candido, e M. T. M. Silva. A low-cost algorithm for adaptive sampling and censoring in diffusion networks. Submetido à *IEEE Transactions on Signal Processing* em 27 de março de 2020.

As principais contribuições desta dissertação são:

1. **Algoritmo adaptativo de amostragem e censura para soluções distribuídas**.
   Foi proposto um mecanismo adaptativo de amostragem para soluções adaptativas difusas e baseadas em grafos. Mostra-se que o algoritmo proposto leva a uma redução significativa no custo computacional durante o regime permanente ao mesmo tempo em que preserva a taxa de convergência. Com uma pequena modificação, ele pode ainda ser utilizado como técnica de censura. Assim, reduz-se o número de transmissões entre os nós, o que possibilita uma economia de energia e consequentemente estende a vida útil da rede. O algoritmo é apresentado no Capítulo 3.

2. **Análise teórica do algoritmo proposto**.
   Foram obtidos resultados analíticos acerca do mecanismo de amostragem proposto. Esses resultados permitem um melhor entendimento do seu funcionamento e auxiliam na escolha dos seus parâmetros de adaptação. Se as variâncias mínima e máxima do ruído na rede forem conhecidas, a análise permite ainda estimar limites teóricos para o número de nós amostrados na rede e estabelece uma condição suficiente para que o algoritmo proposto leve a uma redução em termos de custo computacional em regime permanente. Os resultados dessa análise são apresentados no Capítulo 4.

1.5 Organização do trabalho

O presente trabalho está estruturado em cinco capítulos e um apêndice. No Capítulo 2, expõem-se conceitos fundamentais de filtragem adaptativa, de processamento distribuído de sinais e de filtragem adaptativa baseada em grafos.

No Capítulo 3, deriva-se um mecanismo adaptativo de amostragem para redes de difusão. Mostra-se por meio de simulações computacionais que o método proposto é capaz de reduzir o custo computacional sem impactar a taxa de convergência dos algoritmos em que é aplicado. Também é mostrado que ele pode ser utilizado como técnica de censura, o que reduz o consumo energético da rede.

No Capítulo 4, realiza-se uma análise do algoritmo proposto. Nela, obtêm-se condições para o seu bom comportamento em função de seus parâmetros e se estuda a economia gerada em termos de custo computacional.

No Capítulo 5, são apresentadas as conclusões e propostas para trabalhos futuros.

No Apêndice A, são apresentados mais alguns conceitos da teoria de processamento de sinais em grafos. Com isso, pretende-se fornecer um panorama geral dessa área nova, complementando a discussão realizada no Capítulo 2. Também são apresentados alguns algoritmos adaptativos voltados a esse tipo de aplicação, os quais são comparados por meio de simulações computacionais.

2 PROCESSAMENTO DISTRIBUÍDO DE SINAIS

Neste capítulo, são introduzidos conceitos de processamento distribuído de sinais. Primeiramente, realiza-se na Seção 2.1 uma breve revisão acerca dos fundamentos de filtragem adaptativa. Nela, são apresentados os algoritmos LMS (*least-mean squares*) e NLMS (*normalized least-mean squares*), que servem de base para o algoritmo distribuído considerado nesta dissertação. Na Seção 2.2, abordam-se redes de difusão adaptadas com a versão distribuída do algoritmo NLMS. Na Seção 2.3, são apresentados conceitos da teoria de processamento de sinais em grafos e o algoritmo distribuído da Seção 2.2 é então estendido para redes de difusão baseadas em grafos. Por fim, na Seção 2.4, são apresentadas as conclusões do capítulo.

2.1 Conceitos fundamentais de filtragem adaptativa

Filtros adaptativos são comumente empregados em situações nas quais o ambiente muda de tal forma que um sistema fixo não seria capaz de proporcionar um desempenho adequado. Eles vêm sendo utilizados em inúmeras aplicações, tais como sonar, comunicações, engenharia biomédica, predição de séries temporais, controle ativo de ruído, cancelamento de eco acústico, entre outras (HAYKIN, 2014; DINIZ, 2008; SAYED, 2008).

Em geral, considera-se que o filtro adaptativo apresenta resposta ao impulso unitário finita (*finite impulse response* – FIR) de comprimento M. Neste caso, o sinal de saída do filtro no instante n obtido a partir da operação de convolução, dada por

$$y(n) = \sum_{m=0}^{M-1} w_m(n)u(n-m), \tag{2.1}$$

em que $u(n)$ é o sinal de entrada e $w_0(n), \cdots, w_{M-1}(n)$ são os coeficientes do filtro, como mostrado na Figura 2. Para a formulação dos algoritmos adaptativos, é conveniente usar notação vetorial. Dessa forma, definindo o vetor regressor de entrada e o vetor de coeficientes respectivamente como

$$\mathbf{x}(n) = \begin{bmatrix} u(n) & u(n-1) & \cdots & u(n-M+1) \end{bmatrix}^{\mathrm{T}} \tag{2.2}$$

e

$$\mathbf{w}(n) = \begin{bmatrix} w_0(n) & w_1(n) & \cdots & w_{M-1}(n) \end{bmatrix}^{\mathrm{T}}, \tag{2.3}$$

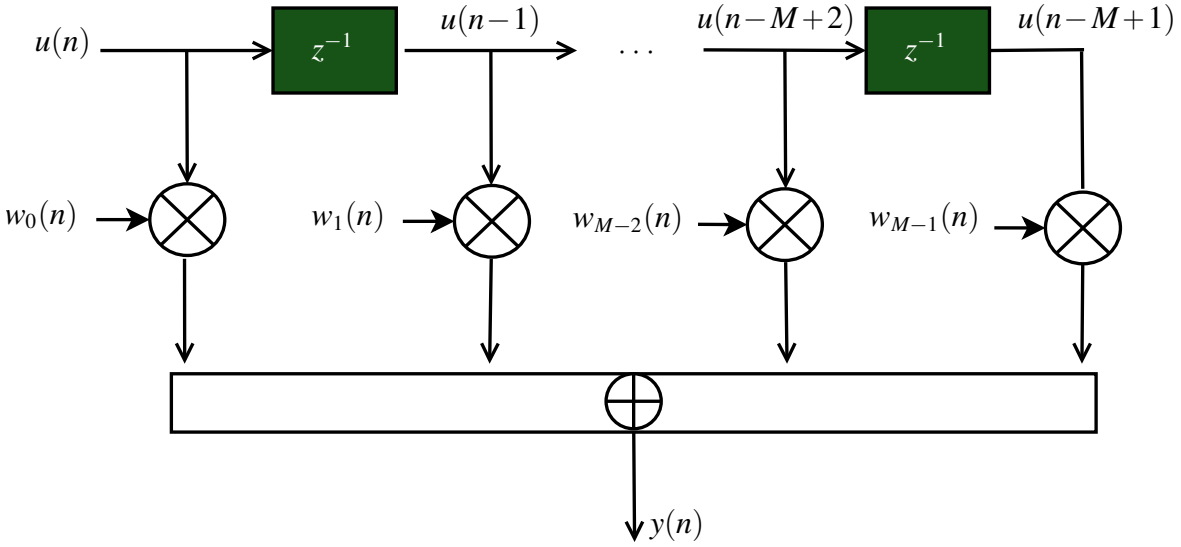


Figura 2: Representação esquemática do cálculo da saída de um filtro adaptativo.
Fonte: Autor.

pode-se calcular a saída como

$$y(n) = \mathbf{x}^{\mathrm{T}}(n)\mathbf{w}(n), \tag{2.4}$$

em que $(\cdot)^T$ representa transposição.

Apesar das particularidades de cada aplicação e das diferenças entre os algoritmos existentes, é possível estabelecer uma formulação comum para os problemas de filtragem adaptativa. Em um contexto supervisionado, a atuação desse tipo de filtro pode ser descrita da seguinte forma: recebe-se como entrada um sinal $u(n)$, a partir do qual procura-se estimar o sinal $d(n)$, denominado **sinal desejado**. O sinal $e(n)$, denominado **sinal de erro** ou **erro de estimação**, é então calculado por

$$e(n) = d(n) - y(n), \tag{2.5}$$

ou seja, pela diferença entre o sinal desejado e a saída do filtro adaptativo $y(n)$. Essa situação é representada de forma esquemática na Figura 3 (HAYKIN, 2014; DINIZ, 2008; SAYED, 2008).

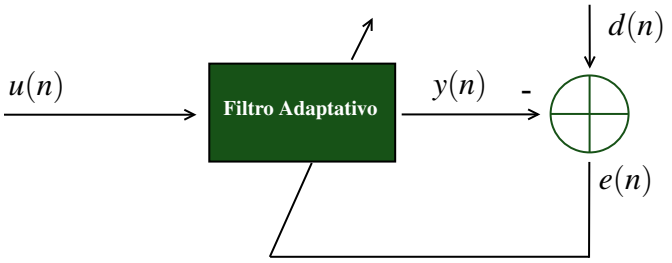


Figura 3: Entradas e saídas do filtro adaptativo.
Fonte: Autor.

Normalmente, o ajuste dos filtros adaptativos é realizado por meio da medição de $e(n)$, no sentido de minimizar a função custo do erro quadrático médio (*mean-square error* – MSE), dada por (HAYKIN, 2014; DINIZ, 2008; SAYED, 2008)

$$J \triangleq \mathrm{E}\{e^2(n)\}, \tag{2.6}$$

em que $\mathrm{E}\{\cdot\}$ denota o operador **esperança matemática**. O algoritmo LMS visa a minimizar uma estimativa instantânea de J, sendo portanto um algoritmo do tipo gradiente estocástico. Dessa forma, obtém-se a seguinte equação de atualização do vetor de pesos $\mathbf{w}(n)$ (HAYKIN, 2014; DINIZ, 2008; SAYED, 2008):

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu e(n)\mathbf{x}(n), \tag{2.7}$$

em que μ é o passo de adaptação, uma constante que o projetista deve escolher e que influencia o comportamento do algoritmo (HAYKIN, 2014). Se μ for muito pequeno, o algoritmo pode atingir um erro quadrático relativamente baixo em regime permanente, mas pode demorar a convergir. Conforme se aumenta μ, o algoritmo converge cada vez mais rapidamente, mas o patamar de erro quadrático atingido em regime permanente se torna cada vez mais elevado, até o ponto em que o algoritmo diverge. É possível mostrar que se o sinal de entrada apresenta potência σ_u^2, o intervalo de μ para que o algoritmo não divirja na média quadrática é (DINIZ, 2008)

$$0 < \mu < \frac{2}{3M\sigma_u^2}. \tag{2.8}$$

Este é um resultado teórico que é obtido quando se fazem algumas hipóteses simplificadoras acerca dos sinais envolvidos. Como nem sempre essas hipóteses valem na prática, é possível que o algoritmo divirja mesmo se escolhendo um valor de μ dentro desse intervalo (HAYKIN, 2014; DINIZ, 2008; SAYED, 2008).

Para facilitar a escolha do passo de adaptação, foi proposto o algoritmo LMS normalizado (NLMS) que utiliza um passo variante no tempo, dado por (HAYKIN, 2014; DINIZ, 2008; SAYED, 2008)

$$\mu(n) = \frac{\widetilde{\mu}}{\delta + \|\mathbf{x}(n)\|^2}, \tag{2.9}$$

em que δ é uma constante positiva usada para evitar divisão por zero, $\widetilde{\mu}$ é um passo de adaptação fixo e $\|\cdot\|$ representa a norma euclidiana. Com esse passo, a atualização do vetor de coeficientes

passa a ser dada por

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \frac{\tilde{\mu}}{\delta + \|\mathbf{x}(n)\|^2}\mathbf{x}(n)e(n). \tag{2.10}$$

É possível mostrar que, para evitar a divergência, $\tilde{\mu}$ deve ser escolhido no intervalo (HAYKIN, 2014; DINIZ, 2008; SAYED, 2008)

$$0 < \tilde{\mu} < 2. \tag{2.11}$$

Diferentemente do LMS, o intervalo do passo de adaptação do NLMS independe da potência do sinal de entrada. Isso torna mais simples a escolha desse parâmetro, já que em algumas aplicações a potência do sinal de entrada pode variar com o tempo (HAYKIN, 2014; DINIZ, 2008; SAYED, 2008).

A área de filtragem adaptativa pode ser considerada consolidada na literatura. A busca por algoritmos com boas propriedades de rastreamento (*tracking*), baixo MSE em regime permanente, velocidade de convergência elevada e baixo custo computacional levaram a diferentes algoritmos desde o surgimento do LMS (HAYKIN, 2014; DINIZ, 2008; SAYED, 2008). Podem-se citar, por exemplo, o algoritmo dos mínimos quadrados recursivos (*recursive least-squares* – RLS), o algoritmo de projeções afins (*affine projection algorithm* – APA), o *set-membership*, entre muitos outros (HAYKIN, 2014; DINIZ, 2008; SAYED, 2008).

2.1.1 Tipos de aplicação

Devido à sua inerente capacidade de se ajustar a diferentes ambientes, filtros adaptativos se consolidaram como ferramentas poderosas em diversas áreas. É comum dividir as aplicações envolvendo filtros adaptativos em quatro grupos (HAYKIN, 2014; DINIZ, 2008; NASCIMENTO; SILVA, 2014):

1. cancelamento de interferência;
2. identificação de sistemas;
3. predição; e
4. identificação inversa de sistemas.

Neste trabalho, a maior parte das simulações realizadas pode ser classificada como identificação de sistemas e, por isso, descreve-se aqui apenas esse tipo de aplicação. Um esquema

de identificação de sistemas é mostrado na Figura 4. Nesse tipo de aplicação, alimenta-se o filtro adaptativo com a mesma entrada que o sistema que se quer identificar. O filtro tenta então reproduzir a saída desse sistema, denotada na Figura 4 por $y^{\text{ótimo}}(n)$. O sinal $\eta(n)$ representa um ruído de medição, geralmente considerado independente dos demais sinais. Ele pode ser usado para modelar ruído térmico e imperfeições dos sensores utilizados, ou pode representar as limitações do modelo adotado para o sistema real (HAYKIN, 2014; DINIZ, 2008; NASCIMENTO; SILVA, 2014). Espera-se que com o tempo $y(n)$ e $e(n)$ se aproximem respectivamente de $y^{\text{ótimo}}(n)$ e de $\eta(n)$ (HAYKIN, 2014; DINIZ, 2008; NASCIMENTO; SILVA, 2014).

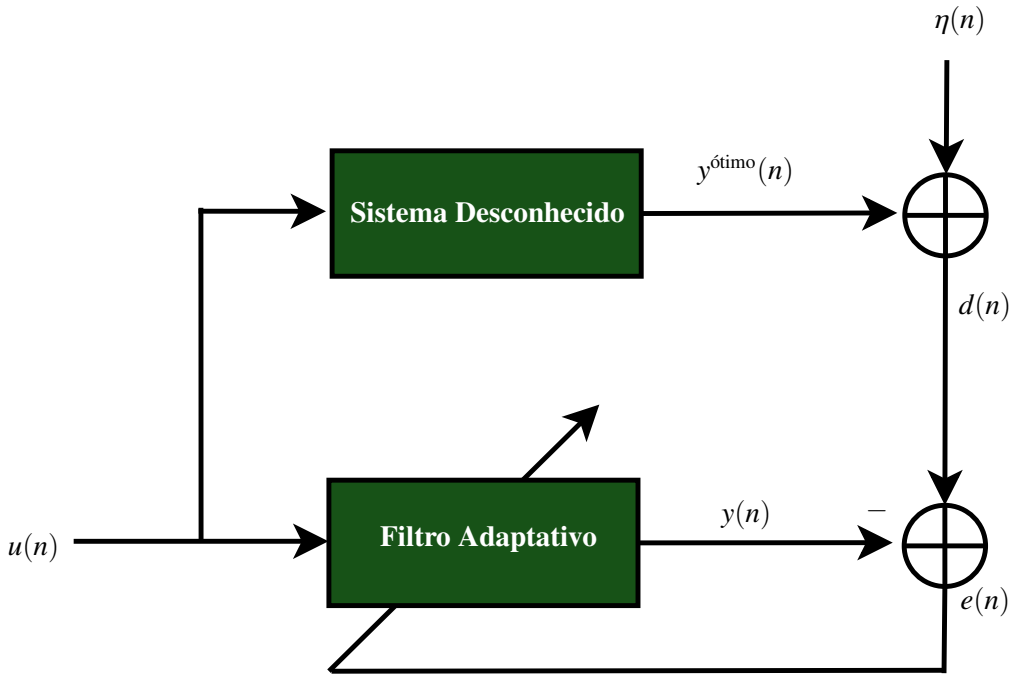


Figura 4: Representação esquemática da aplicação de um filtro adaptativo em identificação de sistemas. Na figura, $y^{\text{ótimo}}(n)$ representa a saída do sistema ótimo que se pretende identificar.
Fonte: Autor.

2.2 Redes de difusão adaptativas "clássicas"

Ao longo das últimas décadas, sensores e processadores de baixo consumo energético se tornaram cada vez mais difundidos e acessíveis, o que levou à popularização de redes de sensores em diversas aplicações (CATTIVELLI; SAYED, 2009). Exemplos incluem a localização e rastreamento de alvos, sensoriamento espectral em telefonia móvel, aplicações médicas, dentre outras (SAYED, 2014). Consequentemente, surgiu um forte interesse por parte da comunidade de processamento de sinais no desenvolvimento de soluções distribuídas. Foi nesse contexto que foram propostas as redes de difusão adaptativas, que têm sido amplamente estudadas na

literatura por serem soluções eficientes em muitas das aplicações citadas (FERNANDEZ-BES et al., 2017).

A formulação matemática dos problemas de estimação distribuída pode ser feita da seguinte forma. Considere uma rede com uma topologia definida e V nós com rótulos $\{1, \cdots, k, \cdots, V\}$. O conjunto de nós com o qual o nó k consegue se comunicar (incluindo o próprio nó k) é denominado **de vizinhança do nó** k e é denotado por $\mathcal{N}_k$, tendo cardinalidade $|\mathcal{N}_k|$. Além disso, como indicado na Figura 5, cada nó tem acesso a um sinal de entrada $u_k(n)$ e a um sinal desejado $d_k(n)$, modelado como

$$d_k(n) = \mathbf{x}_k^{\mathrm{T}}(n)\mathbf{w}^{\mathrm{o}} + \eta_k(n), \tag{2.12}$$

em que $\mathbf{x}_k(n)$ é um vetor regressor local, calculado como em (2.2), $\mathbf{w}^{\mathrm{o}}$ denota o sistema real desconhecido e $\eta_k(n)$ o ruído de medição no nó k. Considera-se que o ruído $\eta_k(n)$ é independente de qualquer outra variável e estacionário no sentido amplo, apresentando média nula e variância $\sigma_{\eta_k}^2$. Além disso, por simplicidade, nesta análise se considera que o vetor de coeficientes $\mathbf{w}^{\mathrm{o}}$ não varia no tempo. O objetivo da rede consiste em obter uma estimativa $\mathbf{w}$ do vetor $\mathbf{w}^{\mathrm{o}}$ de maneira distribuída, resolvendo o problema de otimização (SAYED, 2014; LOPES; SAYED, 2008b; CATTIVELLI; SAYED, 2009)

$$\min_{\mathbf{w}} J(\mathbf{w}) = \min_{\mathbf{w}} \sum_{k=1}^{V} J_k(\mathbf{w}), \tag{2.13}$$

em que $J_k(\mathbf{w})$ são funções custo locais em cada nó k, dadas por

$$J_k(\mathbf{w}) \triangleq \mathrm{E}\{(d_k(n) - \mathbf{x}_k^{\mathrm{T}}(n)\mathbf{w})^2\}. \tag{2.14}$$

Por esse motivo, o sistema que se deseja estimar e o vetor $\mathbf{w}^{\mathrm{o}}$ que o representa são frequentemente chamados de "sistema ótimo" na literatura (SAYED, 2008; SAYED, 2014).

A cada iteração, cada nó k calcula uma estimativa local de $\mathbf{w}^{\mathrm{o}}$ a fim de minimizar sua própria função custo. Essa etapa é realizada utilizando-se apenas as informações disponíveis localmente. Em seguida, os nós se comunicam com os seus vizinhos e a cooperação entre eles leva ao surgimento de uma estimativa global. É possível mostrar que, quando a combinação das estimativas locais é feita adequadamente, elas convergem para uma única solução comum (SAYED, 2014; LOPES; SAYED, 2008b; CATTIVELLI; LOPES; SAYED, 2008b; CATTIVELLI; SAYED, 2009).

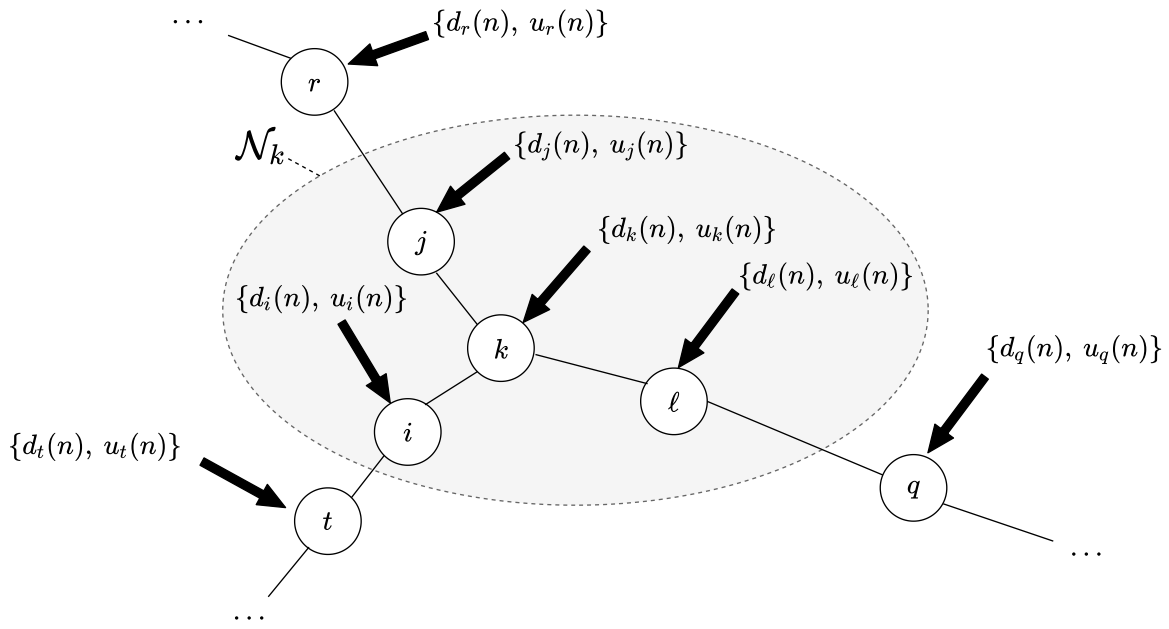


Figura 5: Exemplo de uma rede de difusão. Neste caso, o conjunto $\mathcal{N}_k$ é formado pelos nós i, j, k e ℓ. Fonte: Autor.

Diversas soluções adaptativas foram propostas na literatura para esse problema. Em particular, uma delas consiste em uma versão difusa do algoritmo NLMS (dNLMS). Seguindo uma configuração do tipo ATC, as equações dessa solução são dadas por (SAYED, 2014; LOPES; SAYED, 2008b; CATTIVELLI; SAYED, 2009).

$$\begin{cases} \boldsymbol{\psi}_k(n+1) = \mathbf{w}_k(n) + \mu_k(n)\mathbf{x}_k(n)e_k(n) & \text{(2.15a)} \\ \mathbf{w}_k(n+1) = \displaystyle\sum_{j\in\mathcal{N}_k} c_{jk}\boldsymbol{\psi}_j(n+1), & \text{(2.15b)} \end{cases}$$

em que $\boldsymbol{\psi}_k$ e $\mathbf{w}_k$ representam respectivamente as estimativas locais e combinadas no nó k e

$$e_k(n) = d_k(n) - \mathbf{x}_k^{\mathrm{T}}(n)\mathbf{w}_k(n) \tag{2.16}$$

é o erro de estimação no nó k. Além disso, $\mu_k(n) = \tilde{\mu}_k/[\delta + \|\mathbf{x}_k(n)\|^2]$ é um passo de adaptação normalizado como em (2.9). Por fim, os $\{c_{jk}\}$ são pesos de combinação convexos, satisfazendo (LOPES; SAYED, 2008b; CATTIVELLI; SAYED, 2009)

$$c_{jk} \geqslant 0,\ \sum_{j\in\mathcal{N}_k} c_{jk} = 1, \quad \text{e} \quad c_{jk} = 0 \text{ para } j \notin \mathcal{N}_k. \tag{2.17}$$

É interessante notar a analogia entre as Equações (2.15a) e (2.10). De maneira simplificada, pode-se interpretar que cada nó executa o algoritmo NLMS convencional em paralelo com os demais para atualizar a sua estimativa local $\boldsymbol{\psi}_k$. Entretanto, graças ao cálculo da estimativa

combinada $\mathbf{w}_k$ em (2.15b) e à sua utilização na computação do erro em (2.16) e na atualização da estimativa local em (2.15a), os $\boldsymbol{\psi}_k$ são forçados a convergir para uma única solução global (SAYED, 2014).

Em relação aos pesos $\{c_{jk}\}$, é interessante notar que há diversas possibilidades de escolha para eles. Dentre as estratégias mais utilizadas na literatura, podem-se citar as regras Uniforme (BLONDEL et al., 2005), Laplaciana (XIAO; BOYD, 2004), Metrópolis (LOPES; SAYED, 2008b), do Grau Relativo (CATTIVELLI; LOPES; SAYED, 2008b), entre outras. A literatura também contém algoritmos adaptativos para ajuste dos pesos c_{jk}. Dentre as estratégias adaptativas, destacam-se o algoritmo ACW (*adaptive combination weights*) (TAKAHASHI; YAMADA; SAYED, 2010a; TU; SAYED, 2011), o APA, o algoritmo dos mínimos quadrados (*least-squares* – LS) (FERNANDEZ-BES et al., 2017), entre outros (SAYED, 2014; YU; SAYED, 2013). Por simplicidade, consideram-se neste trabalho apenas as estratégias Uniforme, Metrópolis e ACW descritas a seguir.

Na estratégia Uniforme, cada nó calcula a média aritmética simples das estimativas locais dos seus vizinhos, ou seja,

$$c_{jk} = \begin{cases} \dfrac{1}{|\mathcal{N}_k|}, \text{ se } j \in \mathcal{N}_k \\ 0, \quad \text{caso contrário} \end{cases} . \tag{2.18}$$

Na regra Metrópolis, os pesos são atribuídos de acordo com

$$c_{jk} = \begin{cases} \dfrac{1}{\max\{|\mathcal{N}_k|,|\mathcal{N}_j|\}}, \text{ se } j \in \mathcal{N}_k \text{ e } j \neq k \\ 0, \qquad\qquad \text{se } j \notin \mathcal{N}_k \\ 1 - \sum_{i \in \mathcal{N}_k} c_{ik}, \quad \text{se } j = k \end{cases} . \tag{2.19}$$

Já o ACW incorpora a informação do perfil do ruído ao longo da rede e é obtido ao se resolver um problema de otimização em relação aos pesos $\{c_{jk}\}$ (TAKAHASHI; YAMADA; SAYED, 2010a; TU; SAYED, 2011). Suas equações são dadas por

$$c_{jk}(n) = \begin{cases} \dfrac{\widehat{\sigma}_{jk}^{-2}(n+1)}{\sum_{\ell \in \mathcal{N}_k} \widehat{\sigma}_{\ell k}^{-2}(n+1)}, \text{ se } j \in \mathcal{N}_k \\ 0, \qquad\qquad\qquad \text{caso contrário} \end{cases} , \tag{2.20}$$

em que $\widehat{\sigma}^2_{jk}$ representa a estimativa de $\sigma^2_{\eta_j}$ calculada no nó k e é atualizada com base em

$$\widehat{\sigma}^2_{jk}(n+1) = (1-\nu_k)\widehat{\sigma}^2_{jk}(n) + \nu_k \|\boldsymbol{\psi}_j(n+1) - \mathbf{w}_k(n)\|^2, \tag{2.21}$$

com $\nu_k > 0$ para $k = 1, \cdots, V$. Dessa forma, atribuem-se pesos mais elevados aos nós em que a variância do ruído é menor (TU; SAYED, 2011). É interessante notar que os pesos assim definidos satisfazem a (2.17) a cada iteração. Neste trabalho, para evitar divisão por zero, adota-se uma versão regularizada de (2.20). Para isso, substitui-se $\widehat{\sigma}^{-2}_{jk}(n)$ e $\widehat{\sigma}^{-2}_{\ell k}(n)$ por $[\delta_c + \widehat{\sigma}^2_{jk}(n)]^{-1}$ e $[\delta_c + \widehat{\sigma}^2_{\ell k}(n)]^{-1}$ em (2.20), respectivamente, em que $\delta_c > 0$ é uma constante pequena.

Por fim, cabe notar que também há a versão CTA do algoritmo, em que a ordem das Equações (2.15a) e (2.15b) é trocada (SAYED, 2014; LOPES; SAYED, 2008b; CATTIVELLI; SAYED, 2009). Cabe ainda mencionar que foram propostas outras soluções adaptativas difusas, como versões distribuídas do APA, do algoritmo RLS, entre outras (LI; CHAMBERS, 2009; CATTIVELLI; LOPES; SAYED, 2008b). Entretanto, para efeito de simplicidade, o presente trabalho focará em versões distribuídas do tipo NLMS com estratégia ATC.

2.3 Redes de difusão adaptativas que incorporam a teoria de grafos

Nesta seção, são descritos conceitos fundamentais da teoria de grafos e de processamento de sinais em grafos para então estender o algoritmo dNLMS da Seção 2.2 para redes de difusão que incorporam essa teoria em sua adaptação. Cabe mencionar que o processamento de sinais em grafos conta com um vasto arcabouço teórico. No Apêndice A, apresentam-se conceitos complementares e soluções adaptativas não distribuídas baseadas em grafos.

2.3.1 Grafos: classificações e definições

Grafos são estruturas comumente utilizadas para representar interações entre objetos de interesse, consistindo em um conjunto de pontos, denominados **nós**, e em um conjunto de linhas conectando determinados pares de pontos, denominadas **arestas** (LATOUCHE; ROSSI, 2015). Tipicamente, os nós em um grafo representam objetos e/ou agentes, ao passo que as arestas descrevem relações entre os objetos ou agentes (LATOUCHE; ROSSI, 2015; BONDY; MURTY, 1976). Na Figura 6 é mostrado um exemplo de grafo.

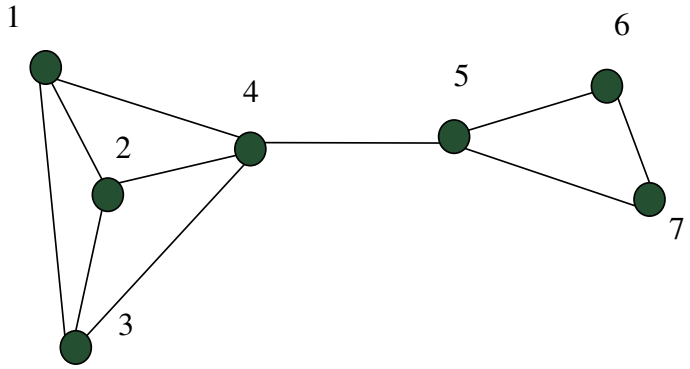


Figura 6: Um grafo, com os seus nós numerados de 1 a 7 e as suas arestas representadas pelas linhas que conectam um nó a outro.

Fonte: Autor.

Nota-se que há uma analogia entre o conceito de grafo e a definição de redes adaptativas apresentada na Seção 2.2. Não se trata de mero acaso. Grafos são abstrações matemáticas capazes de representar uma grande variedade de situações do mundo real (BONDY; MURTY, 1976). Por esse motivo, encontram aplicações nas mais diversas áreas. Exemplos incluem a análise de imagens, engenharia de *software*, inteligência artificial, processamento de linguagem natural, entre outras (FRANCESCONI et al., 1997; KRAHMER; ERK; VERLEG, 2003; MASON; BLAKE, 2001; BARESI; HECKEL, 2002; COLLBERG et al., 2003; RUSSELL; NORVIG, 2016; BUA; GORI; SANTINI, 2002; CRANDALL et al., 2008; BALDI; POLLASTRI, 2003).

Grafos podem ser classificados de acordo com uma série de propriedades de seus nós e de suas arestas. Um grafo é dito **direcionado** quando a cada uma de suas arestas está associada uma dada orientação. Por exemplo, uma aresta conectando os nós i e j pode estar orientada "de i para j" ou "de j para i". Grafos direcionados também podem ser chamados de "grafos dirigidos" ou "grafos orientados". Eles são úteis, por exemplo, em aplicações envolvendo tráfego em que certos caminhos são de "mão única" (BONDY; MURTY, 1976).

Ainda em relação à classificação de grafos, quando a cada uma de suas arestas está associado um número real, denominado "peso" da aresta, diz-se que o grafo é **ponderado** (BONDY; MURTY, 1976). O significado dos pesos depende do que os nós e as arestas de um dado grafo representam. Por exemplo, se em uma determinada situação, os nós de um grafo representam pontos de interesse em um mapa e as arestas a existência de um caminho conectando esses pontos, pesos podem ser utilizados para indicar a intensidade do tráfego por esses caminhos (BONDY; MURTY, 1976).

Diz-se que um grafo é **finito** se os seus conjuntos de nós e de arestas forem finitos. Por fim,

diz-se que um grafo é **simples** se nenhum par de nós está diretamente conectado por mais de uma aresta e se não há arestas conectando um nó a ele mesmo (BONDY; MURTY, 1976).

Uma propriedade relevante dos grafos é que eles admitem representações matriciais, o que os torna manipuláveis computacionalmente (BONDY; MURTY, 1976). Uma das formas de representá-los dessa maneira é por meio da **matriz de adjacência** do grafo. Para um grafo não ponderado $\mathcal{G}$ dotado de V nós, a matriz de adjacência $\mathbf{A}$ é uma matriz $V \times V$ cujo elemento $\mathbf{A}_{i,j}$ é igual ao número de arestas ligando os nós i e j (BONDY; MURTY, 1976). Para grafos simples e ponderados, a definição anterior pode ser estendida considerando-se $\mathbf{A}_{i,j}$ igual ao valor do peso associado à aresta que conecta os nós i e j. É interessante notar que a matriz de adjacência de um grafo não direcionado é simétrica com relação à diagonal principal e que se o grafo for simples e não ponderado, todos os elementos de sua matriz de adjacência são iguais a 0 ou 1. Por exemplo, a matriz de adjacência do grafo não direcionado, simples e não ponderado da Figura 6 é dada por

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 \end{bmatrix}. \tag{2.22}$$

Por fim, três conceitos importantes que ainda devem ser apresentados são os de **distância**, **vizinho** e de **grau**. A distância entre quaisquer dois nós i e j é igual ao menor número de arestas que devem ser percorridas para se chegar ao nó j partindo-se do nó i. Por exemplo, a distância entre os nós 1 e 5 na Figura 6 é igual a dois. Um nó é vizinho de outro se ele está diretamente ligado a ele por meio de uma aresta. O conjunto formado pelos vizinhos de um nó é denominado a **vizinhança aberta**. Pode-se definir ainda a **vizinhança fechada** de um certo nó i como sendo o conjunto formado pela união do próprio nó i com a sua vizinhança aberta. Neste trabalho, por simplicidade, será usado apenas o termo "vizinhança" para se referir à vizinhança fechada de um dado nó i, a qual será denotada por $\mathcal{N}_i$. Cabe observar que este conceito de vizinhança coincide com aquele exposto na Seção 2.2, o que possibilita o uso de uma notação comum. Finalmente, o grau de um nó é igual ao número de arestas que incide sobre ele (BONDY; MURTY, 1976; LATOUCHE; ROSSI, 2015). Por exemplo, para o nó 4

do grafo da Figura 6, tem-se $\mathcal{N}_4 = \{1, 2, 3, 4, 5\}$ e o seu grau é igual a 4. Para grafos não direcionados como o da Figura 6, o grau de cada nó é igual à cardinalidade de sua vizinhança aberta.

2.3.2 Uma breve introdução ao Processamento de sinais em grafos

Inicialmente, convém definir o que é um sinal em um grafo. Para isso, nota-se que essas estruturas podem ser interpretadas como uma coleção finita de amostras, cada uma associada a um de seus nós, ao passo que as suas arestas são determinadas pela física do problema ou inferidas a partir de algum conhecimento sobre os dados. Coletivamente, essas amostras podem ser entendidas como um **sinal definido sobre um grafo** (SHUMAN et al., 2013b). Por brevidade, o termo "sinal no grafo" também é utilizado.

Considere um grafo $\mathcal{G}$ dotado de V nós, com rótulos $k = 1,\ 2,\ \cdots,\ V$ e matriz de adjacência $\mathbf{A}$. Por simplicidade, costuma-se considerar que a topologia do grafo é constante no tempo. Representa-se um sinal definido sobre $\mathcal{G}$ por meio de um vetor $\mathbf{u} \in \mathbb{R}^V$ da forma $\mathbf{u}(n) = [u_1(n)\ u_2(n)\ \cdots\ u_V(n)]$, em que cada elemento $u_i(n)$ é indexado por um nó i (SANDRYHAILA; MOURA, 2013b). Na Figura 7, é mostrado um exemplo baseado no grafo da Figura 6. Considera-se um sinal constante no tempo, dado por

$$\mathbf{u}(n) = [1 \quad 1 \quad 1 \quad 0 \ -1 \ -2 \ -2]^{\mathrm{T}}. \tag{2.23}$$

As arestas são representadas por linhas tracejadas, ao passo que os valores do sinal em cada nó são representados pelas linhas contínuas, estando indicados na figura (valores positivos para cima e negativos para baixo).

A partir dessa definição, também foi proposto na literatura o conceito de **filtro** para sinais sobre grafos (SANDRYHAILA; MOURA, 2013b). Para isso, parte-se de uma analogia entre a matriz de adjacência e o operador atraso unitário da teoria de processamento de sinais de tempo discreto. Considere uma sequência periódica de tempo discreto dotada de V amostras $u_1,\ u_2,\ \cdots,\ u_V$. Essa situação poderia ser representada por um grafo, em que cada amostra u_k está associada a um nó k correspondente. Além disso, há uma aresta direcionada partindo de cada nó k para o seu sucessor, o nó $k+1$, de modo a indicar a evolução da sequência no tempo, como ilustrado na Figura 8.

A matriz de adjacência correspondente a esse grafo é dada por (SANDRYHAILA;

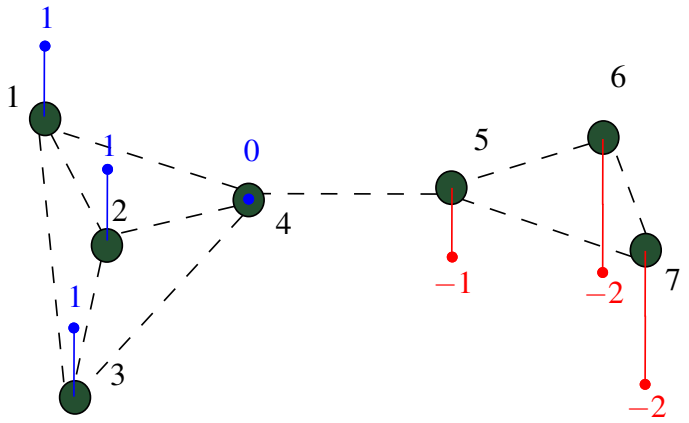


Figura 7: Um exemplo de sinal definido sobre um grafo. Os nós estão numerados de 1 a 7 e as arestas são representadas pelas linhas tracejadas, ao passo que o valor do sinal $\mathbf{u}(n)$ em cada nó i é representado pelas linhas contínuas que se projetam de cada nó (valores positivos para cima e negativos para baixo).
Fonte: Autor.

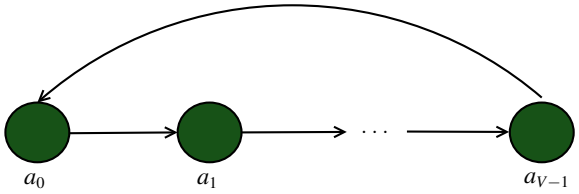


Figura 8: Representação de uma sequência de tempo discreto por um grafo.
Fonte: Autor.

MOURA, 2013b)

$$\mathbf{A} = \begin{bmatrix} 0 & 0 & \cdots & 0 & 1 \\ 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \end{bmatrix}.$$

Agrupando-se as amostras da sequência de tempo discreto em um vetor $\mathbf{u} = [u_0\ u_1\ \cdots\ u_{V-1}]$, nota-se que o produto da matriz $\mathbf{A}$ por $\mathbf{u}$ é o vetor $\bar{\mathbf{u}} = \mathbf{A}\mathbf{u} = [u_{V-1}\ u_0\ \cdots\ u_{V-2}]$. Levando-se em conta que $\mathbf{u}$ é uma representação vetorial de uma sequência de tempo discreto, $\bar{\mathbf{u}}$ pode ser interpretado como uma versão de $\mathbf{u}$ atrasada de uma unidade de tempo. Sob essa perspectiva, a matriz de adjacência cumpre um papel análogo ao operador atraso unitário.

Apesar dessa analogia se basear especificamente no grafo da Figura 8, essa noção é generalizada para qualquer matriz de adjacência $\mathbf{A}$ no contexto de processamento de sinais em grafos. No caso geral, pode-se interpretar que a multiplicação de um sinal $\mathbf{u}$ definido sobre um grafo pela matriz $\mathbf{A}$ representa um deslocamento espacial do sinal ao longo do grafo. Por esse mo-

tivo, a matriz de adjacência é considerada um **operador de deslocamento no grafo** (*graph-shift operator*).

Cabe ressaltar que, ao longo do tempo, propôs-se utilizar outras matrizes como operador de deslocamento no grafo, como é o caso da matriz Laplaciana (NASSIF et al., 2018; HUA et al., 2018). Por simplicidade, neste trabalho apenas a matriz de adjacência será utilizada como operador de deslocamento.

Diante disso, define-se um filtro linear e invariante ao deslocamento no grafo (*linear shift-invariant filter*) como um sistema (SANDRYHAILA; MOURA, 2013b)

$$\mathbf{H} \triangleq \sum_{\ell=0}^{L-1} w_\ell^{\mathrm{o}} \mathbf{A}^\ell, \tag{2.24}$$

em que L é o comprimento do filtro e $\{w_\ell^{\mathrm{o}}\}_{\ell=0}^{L-1}$ denota o conjunto dos seus coeficientes. Dessa forma, se um sinal $\mathbf{u}(n)$ é processado por esse sistema, sua saída $\mathbf{h}(n)$ pode ser descrita por (SANDRYHAILA; MOURA, 2013b)

$$\mathbf{h}(n) = \sum_{\ell=0}^{L-1} w_\ell^{\mathrm{o}} \mathbf{A}^\ell \mathbf{u}(n), \tag{2.25}$$

como esquematizado na Figura 9.

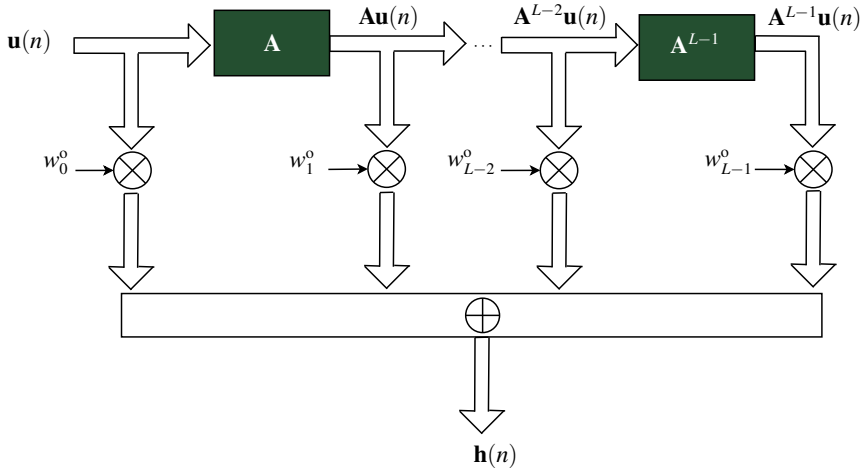


Figura 9: Representação esquemática do funcionamento de um filtro linear e invariante ao deslocamento representado pela Equação (2.25). Cabe notar que a saída do bloco constituído pela matriz $\mathbf{A}$, tendo o vetor $\mathbf{u}(n)$ como entrada, é dada pelo produto de $\mathbf{A}$ por $\mathbf{u}(n)$, ou seja, pelo vetor $\mathbf{A}\mathbf{u}(n)$.
Fonte: Autor.

Em geral, considera-se que não se tem acesso diretamente ao sinal $\mathbf{h}(n)$, mas sim a uma

versão ruidosa do mesmo, que pode ser modelada como (NASSIF et al., 2017)

$$\mathbf{d}(n) = \sum_{\ell=0}^{L-1} w_\ell^{\mathrm{o}} \mathbf{A}^\ell \mathbf{u}(n) + \boldsymbol{\eta}(n), \tag{2.26}$$

em que $\boldsymbol{\eta}(n) = [\eta_1(n), \cdots, \eta_V(n)]^{\mathrm{T}} \in \mathbb{R}^V$ é um ruído estacionário no sentido amplo, com média nula e matriz de covariância diagonal $\mathbf{R}_\eta = \mathrm{diag}\{\sigma_{\eta_k}^2\}_{k=1}^V$, independente e identicamente distribuído (i.i.d.) e independente de qualquer outro sinal (NASSIF et al., 2017).

2.3.3 O algoritmo dNLMS baseado em grafos

Como mencionado na Seção 2.2, redes de difusão adaptativas são úteis em aplicações em que os dados são coletados por uma rede de sensores, levando a medições locais dos sinais de interesse. Como o processamento de sinais sobre grafos foi proposto para lidar exatamente com esse tipo de situação, era natural que a obtenção de algoritmos adaptativos difusos voltados à área se tornasse um tópico de forte interesse. De fato, foram propostas na literatura diversas soluções adaptativas com processamento distribuído (LORENZO et al., 2017a; NASSIF et al., 2017; NASSIF et al., 2018; HUA et al., 2018).

Em particular, em (NASSIF et al., 2018) foi proposta uma extensão do algoritmo dNLMS para a filtragem adaptativa baseada em grafos. Tal extensão tem por objetivo obter uma versão do algoritmo dNLMS que seja capaz de lidar com situações em que os dados coletados pelos sensores estejam fortemente relacionados entre si por meio topologia da rede (NASSIF et al., 2018). Essa situação pode ser modelada pela Equação (2.26). No entanto, em muitas aplicações a informação não se propaga na rede instantaneamente, mas sim ao longo do tempo. Diante disso, é conveniente incorporar o atraso temporal em (2.26), o que leva a

$$\mathbf{d}(n) = \sum_{\ell=0}^{L-1} h_\ell^{\mathrm{o}} \mathbf{A}^\ell \mathbf{u}(n-\ell) + \boldsymbol{\eta}(n). \tag{2.27}$$

Isso é útil, por exemplo, para retratar a chegada gradual de uma frente fria em uma aplicação de meteorologia (HUA et al., 2018; NASSIF et al., 2018).

Definindo-se o vetor $\mathbf{x}_k(n)$ dado por

$$\mathbf{x}_k(n) = \Big[[\mathbf{u}(n)]_k \; [\mathbf{A}\mathbf{u}(n-1)]_k \; \cdots \; [\mathbf{A}^{M-1}\mathbf{u}(n-M+1)]_k \Big]^{\mathrm{T}}, \tag{2.28}$$

em que $[\cdot]_k$ denota o k-ésimo elemento do vetor, pode-se então escrever

$$d_k(n) = \mathbf{x}_k^{\mathrm{T}}(n)\mathbf{w}^{\mathrm{o}} + \eta_k(n),$$

que é idêntica à Equação (2.12) da página 14. A relação entre $\mathbf{x}_k(n)$ e $\mathbf{u}(n)$ em (2.28) é ilustrada na Figura 10 e pode ser interpretada da seguinte forma: $\mathbf{u}(n)$ representa as informações "brutas" disponíveis em cada nó da rede na iteração n, enquanto $\mathbf{x}_k(n)$ modela a disseminação das informações do grafo no nó k. Esse espalhamento é o resultado do deslocamento espacial da informação na rede ao longo do tempo. O vetor de coeficientes $\mathbf{w}^{\mathrm{o}}$ modela de que forma exatamente a topologia do grafo e o tempo afetam a disseminação das informações, ao passo que $d_k(n)$ representa uma medição ruidosa das informações disponíveis no nó k como resultado desse processo (NASSIF et al., 2018; HUA et al., 2018). Cabe notar que há uma analogia entre o diagrama da Figura 10 e a linha de atrasos normalmente encontrada em filtros FIR de tempo discreto da Figura 2 (SAYED, 2008).

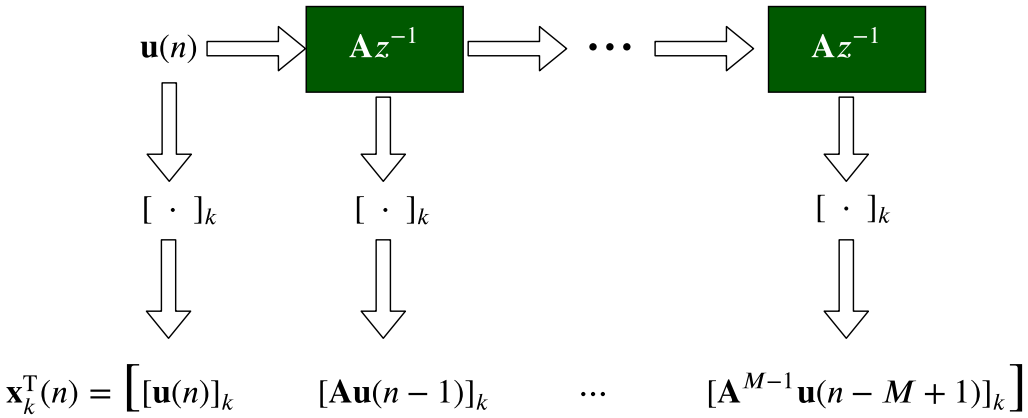


Figura 10: Obtenção de $\mathbf{x}_k(n)$ a partir de $\mathbf{u}(n)$ para cada nó k.
Fonte: Autor

A partir daí, desenvolve-se um algoritmo distribuído do tipo LMS para minimizar a função custo

$$J(\mathbf{w}) = \sum_{k=1}^{V} J_k(\mathbf{w}), \;\; \text{com} \;\; J_k(\mathbf{w}) \triangleq \mathrm{E}\{|d_k(n) - \mathbf{x}_k^{\mathrm{T}}(n)\mathbf{w}|^2\}, \tag{2.29}$$

que coincide com o Critério (2.14) da página 14, usado em redes de difusão adaptativas "clássicas". Calculando-se uma aproximação instantânea para o gradiente de $J(\mathbf{w})$ e seguindo-se uma estratégia de difusão do tipo ATC, obtém-se

$$\begin{cases} \boldsymbol{\psi}_k(n+1) = \mathbf{w}_k(n) + \mu_k(n)\mathbf{x}_k(n)\big[d_k(n) - \mathbf{x}_k^{\mathrm{T}}(n) \cdot \mathbf{w}_k(n)\big] \\ \mathbf{w}_k(n+1) = \sum\limits_{j \in \mathcal{N}_k} c_{jk}\boldsymbol{\psi}_j(n+1), \end{cases}$$

Cabe observar que em (NASSIF et al., 2018), considera-se um passo de adaptação constante para cada nó, o que levou a uma versão LMS do algoritmo, mas neste trabalho optou-se pela versão normalizada.

As equações obtidas coincidem com as de (2.15) e foram repetidas aqui apenas por comodidade. Portanto, a distinção entre o algoritmo dNLMS voltado ao caso clássico e o dNLMS baseado em grafos de (NASSIF et al., 2018) consiste apenas no significado que as entradas adquirem nas duas abordagens. Tal diferença se deve essencialmente aos distintos papeis que o aspecto espacial do problema desempenha em cada caso. No processamento distribuído clássico, a topologia da rede não influencia a dinâmica do sinal desejado. Portanto, $d_k(n)$ depende apenas do sinal $u_k(n)$ e do ruído de medição $\eta_k(n)$ e é independente de $u_i(n)$ para todos os $i \neq k$. Isso ocorre porque as informações não "viajam" de um nó para outro. Já na filtragem adaptativa baseada em grafos, se os nós i e k forem vizinhos imediatos, $d_k(n)$ depende de $u_i(n-1)$, pois as informações de um nó se espalham para seus vizinhos ao longo do tempo. Além disso, se a distância entre os nós j e k é igual a dois, $d_k(n)$ também depende de $u_j(n-2)$ e assim por diante. Portanto, a topologia da rede desempenha um papel importante na forma como o sinal desejado $d_k(n)$ evolui em cada nó k. Isso torna a abordagem baseada em grafos adequada para problemas em que tanto o tempo como o espaço devem ser levados em consideração, como por exemplo, na meteorologia (NASSIF et al., 2018; HUA et al., 2018; SPELTA; MARTINS, 2018; LEWENFUS et al., 2019). Apesar das diferenças conceituais entre as duas abordagens, uma formulação matemática comum pode ser usada para descrevê-los até um certo ponto. Neste trabalho, adota-se esse procedimento para mostrar a generalidade do mecanismo de amostragem proposto, já que ele será empregado tanto em problemas distribuídos clássicos como em filtragem adaptativa baseada em grafos.

2.4 Conclusões

Neste capítulo, foram expostos os conceitos fundamentais de filtragem adaptativa e de processamento distribuído de sinais. Além disso, foi feita uma breve introdução à teoria de grafos e se apresentou o conceito de sinal definido sobre um grafo. Em seguida, expôs-se a extensão do conceito de filtros para sinais definidos sobre grafos. Por fim, foi apresentado o algoritmo dNLMS baseado em grafos. Cabe observar que, após se discutirem as particularidades de cada abordagem, adotou-se uma formulação comum para o algoritmo dNLMS proposto para redes

de difusão adaptativas “clássicas” (SAYED, 2014; LOPES; SAYED, 2008b; CATTIVELLI; LOPES; SAYED, 2008b; CATTIVELLI; SAYED, 2009) e para o dNLMS baseado em grafos (NASSIF et al., 2018). Isso é possível porque, como mostrado na Seção 2.3.3, a diferença entre os dois algoritmos se encontra apenas no significado de suas entradas e saídas.

3 PROPOSTA DE UM ALGORITMO DE AMOSTRAGEM

Neste capítulo, é proposto um algoritmo de amostragem para redes de difusão que regula a quantidade de nós amostrados de acordo com a magnitude do erro. Dessa forma, reduz-se o número de nós amostrados apenas quando tal redução não impacta perceptivelmente o desempenho da rede. Com uma simples modificação, o algoritmo proposto também pode ser usado como um mecanismo de censura, possibilitando dessa forma a redução de transmissão de informação dos nós não amostrados.

O capítulo está dividido da seguinte forma. O mecanismo proposto é derivado na Seção 3.1. Na Seção 3.2, discute-se o seu emprego como técnica de censura e na Seção 3.3 são apresentados resultados de simulação testando-se o algoritmo em diferentes cenários. Por fim, a Seção 3.4 encerra o capítulo com as principais conclusões.

3.1 O algoritmo proposto

A cada iteração, o algoritmo dNLMS estima o vetor de coeficientes $\mathbf{w}^{\mathrm{o}}$ a partir dos dados $\{d_k(n), \mathbf{x}_k(n)\}$. Para incorporar o conceito de amostragem, define-se a variável binária $\bar{t}_k(n)$, que assume os valores um ou zero para determinar se o nó k deve ser amostrado ou não, respectivamente, na iteração n. Assim, pode-se reescrever a etapa de adaptação descrita pela Equação (2.15a) da página 15 como

$$\boxed{\boldsymbol{\psi}_k(n+1) = \mathbf{w}_k(n) + \bar{t}_k(n)\mu_k(n)\mathbf{x}_k(n)e_k(n).} \tag{3.1}$$

Se $\bar{t}_k(n) = 1$, $e_k(n)$ é calculado como na Equação (2.16) e (3.1) coincide com (2.15a). Em contrapartida, se $\bar{t}_k(n) = 0$, $e_k(n)$ não é computado e, portanto, $d_k(n)$ não é amostrado, $\mathbf{x}_k(n)$ e $\mu_k(n)$ não são calculados e $\boldsymbol{\psi}_k(n+1) = \mathbf{w}_k(n)$. Cabe observar que a etapa de combinação descrita pela Equação (2.15b) é mantida sem modificação.

Para obter a variável binária $\bar{t}_k(n) \in \{0, 1\}$ e selecionar os nós que devem ser amostrados, utiliza-se uma variável auxiliar $t_k(n) \in [0, 1]$ tal que

$$\bar{t}_k(n) = \begin{cases} 1, \text{ se } t_k(n) \geqslant 0{,}5, \\ 0, \text{ caso contrário} \end{cases} . \tag{3.2}$$

Procura-se então minimizar a seguinte função custo com relação a $t_k(n)$:

$$J_{t,k}(n) = [t_k(n)]\beta\bar{t}_k(n) + \big[1 - t_k(n)\big]\sum_{i\in\mathcal{N}_k} c_{ik}(n)e_i^2(n), \tag{3.3}$$

em que $\beta > 0$ é um parâmetro introduzido para controlar a taxa de amostragem dos nós. Se o erro quadrático for elevado ou o nó k não estiver sendo amostrado ($\bar{t}_k = 0$), $J_{t,k}(n)$ é minimizada fazendo-se $t_k(n)$ próximo de um, o que leva à amostragem do nó k. Isso assegura que o algoritmo mantenha a amostragem dos nós enquanto o erro for elevado e que a retome nos nós "inativos". Dessa forma, pode-se detectar alterações no ambiente. Em contrapartida, quando o nó k estiver sendo amostrado ($\bar{t}_k = 1$) e o erro quadrático na vizinhança for pequeno em comparação com β, $J_{t,k}(n)$ é minimizada fazendo-se $t_k(n)$ próximo de zero, o que faz com que o algoritmo deixe de amostrar o nó k. O bom comportamento do algoritmo depende de uma escolha adequada para β, o que é estudado no Capítulo 4.

Com base em combinações convexas de filtros adaptativos (ver, e.g.,(ARENAS-GARCIA et al., 2016; LÁZARO-GREDILLA et al., 2010) e suas referências), em vez de atualizar diretamente $t_k(n)$, propõe-se adaptar $\alpha_k(n)$, uma variável relacionada a $t_k(n)$ por meio de (LÁZARO-GREDILLA et al., 2010)

$$t_k(n) = \phi\left[\alpha_k(n)\right](n) \triangleq \frac{\text{sgm}[\alpha_k(n)] - \text{sgm}[-\alpha^+]}{\text{sgm}[\alpha^+] - \text{sgm}[-\alpha^+]}, \tag{3.4}$$

em que $\text{sgm}[x] = (1 + e^{-x})^{-1}$ é a função sigmoidal e α^+ é o maior valor positivo que $\alpha_k(n)$ pode assumir. Um valor comumente adotado na literatura é $\alpha^+ = 4$. Na Figura 11, é mostrado o gráfico de ϕ em função de α_k. Cabe notar que $t_k(n)$ assume os valores 1, 0,5 e 0 para $\alpha_k(n) = \alpha^+$, $\alpha_k(n) = 0$ e $\alpha_k(n) = -\alpha^+$, respectivamente.

Derivando-se (3.3) em relação a $\alpha_k(n)$, obtém-se a regra de atualização

$$\alpha_k(n+1) = \alpha_k(n) + \mu_t\phi'\left[\alpha_k(n)\right]\left[\sum_{i\in\mathcal{N}_k} c_{ik}(n)e_i^2(n) - \beta t_k(n)\right], \tag{3.5}$$

em que $\mu_t > 0$ é um passo de adaptação e

$$\phi'\left[\alpha_k(n)\right] \triangleq \frac{dt_k(n)}{d\alpha_k(n)} = \frac{\text{sgm}[\alpha_k(n)]\{1 - \text{sgm}[\alpha_k(n)]\}}{\text{sgm}[\alpha^+] - \text{sgm}[-\alpha^+]}. \tag{3.6}$$

A Equação (3.5) não pode ser usada em sua forma atual, pois requer o cálculo do erro em todos os nós para decidir se eles devem ser amostrados ou não, o que é contraditório. Por isso,

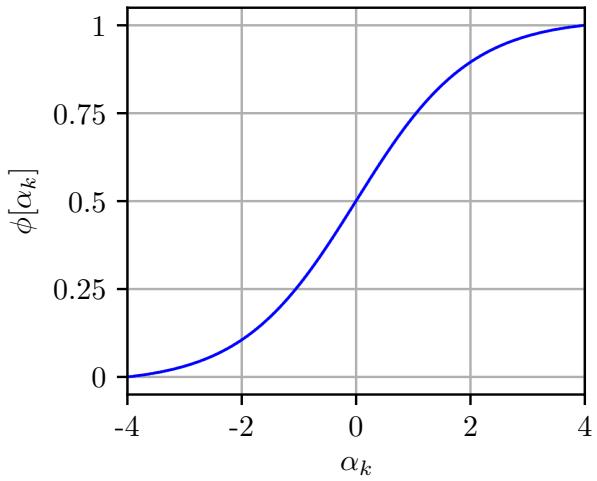


Figura 11: Gráfico de ϕ em função de α_k para $-4 \leqslant \alpha_k \leqslant 4$.
Fonte: Autor.

aplica-se uma modificação em (3.5), substituindo-se $e_i(n)$ por $\varepsilon_i(n)$, que denota a última medida de $e_i(n)$ à qual se tem acesso. Quando o nó i é amostrado, $\varepsilon_i(n) = e_i(n)$. Assim, obtém-se

$$\boxed{\alpha_k(n+1) = \alpha_k(n) + \mu_t \phi'\left[\alpha_k(n)\right]\left[\sum_{i \in \mathcal{N}_k} c_{ik}(n)\varepsilon_i^2(n) - \beta \bar{t}_k(n)\right]}. \tag{3.7}$$

A Equação (3.7) é a base do mecanismo de amostragem proposto. Junto com a Equação (3.1), ela leva a uma versão do algoritmo dNLMS com amostragem adaptativa (*adaptive-sampling* dNLMS – AS-dNLMS). O algoritmo é descrito na forma de pseudo-código na Tabela 1. Como (3.7) depende apenas da última medição do erro de estimação em cada nó, a técnica proposta pode ser estendida para qualquer algoritmo adaptativo com processamento distribuído.

É interessante notar que, apesar de se usar a variável $t_k(n)$ na dedução do algoritmo, ela não precisa ser calculada explicitamente, já que não aparece em (3.1) ou (3.7). Nessas equações, aparecem apenas as variáveis $\bar{t}_k(n)$ e $\dfrac{dt_k(n)}{d\alpha_k(n)}$. A primeira está relacionada a $\alpha_k(n)$ por meio de

$$\bar{t}_k(n) = \begin{cases} 1, \text{ se } \alpha_k(n) \geqslant 0, \\ 0, \text{ caso contrário} \end{cases}, \tag{3.8}$$

como se pode observar a partir de (3.2) e (3.4), enquanto a segunda pode ser armazenada em uma tabela para consultas (*lookup table*), considerando-se uma discretização dos valores de α_k.

O funcionamento do mecanismo de amostragem pode ser explicado da seguinte forma: o al-

Tabela 1: Descrição em pseudo-código do algoritmo AS-dNLMS em sua configuração ATC

% Inicialização

Para cada nó $i = 1, \cdots, V$, faça $\alpha_i(0) \leftarrow \alpha^+$, $\bar{t}_i(0) \leftarrow 1$, $\varepsilon_i(n) \leftarrow 0$,
$\mathbf{x}_i(n) = \mathbf{0}$, $\boldsymbol{\psi}_i(0) \leftarrow \mathbf{0}$, $\mathbf{w}_i(0) \leftarrow \mathbf{0}$.

% A seguir, repita para cada iteração $n \geqslant 0$ e cada nó k:

% Etapa de Adaptação

Se $\alpha_k(n) \geqslant 0$, **faça** :
$\bar{t}_k(n) \leftarrow 1$
Senão, faça:
$\bar{t}_k(n) \leftarrow 0$
Fim
Se $\bar{t}_k(n) = 1$, **faça** :
Atualize $\mathbf{x}_k(n)$ e $\|\mathbf{x}_k(n)\|^2$
$\mu_k(n) \leftarrow \widetilde{\mu}_k / [\delta + \|\mathbf{x}_k(n)\|^2]$
$e_k(n) \leftarrow d_k(n) - \mathbf{x}_k^{\mathrm{T}}(n)\mathbf{w}_k(n)$
$\varepsilon_k(n) \leftarrow e_k(n)$
$\boldsymbol{\psi}_k(n+1) \leftarrow \mathbf{w}_k(n) + \bar{t}_k(n)\mu_k(n)\mathbf{x}_k(n)e_k(n)$
Senão, faça:
$\boldsymbol{\psi}_k(n+1) \leftarrow \mathbf{w}_k(n)$
Fim

% Transmissão

Transmita $\boldsymbol{\psi}_k(n+1)$ e $\varepsilon_k^2(n)$ para cada nó $\in \mathcal{N}_k$

% Etapa de Combinação

$\alpha_k(n+1) \leftarrow \alpha_k(n) + \mu_t \phi'[\alpha_k(n)] \left[\sum_{i \in \mathcal{N}_k} c_{ik}(n) \varepsilon_i^2(n) - \beta \bar{t}_k(n) \right]$
$\mathbf{w}_k(n+1) \leftarrow \sum_{j \in \mathcal{N}_k} c_{jk}(n) \boldsymbol{\psi}_j(n+1)$

Fonte: Autor

goritmo é inicializado com todos os nós amostrados ($\bar{t}_k = 1$) e $\alpha_k(0) = \alpha^+$ para $k \in \{1, \cdots, V\}$. Portanto, (2.15a) é efetuada normalmente e o algoritmo age no sentido de minimizar o MSE. Enquanto a média ponderada do erro quadrático na vizinhança do nó k for maior do que β, a segunda parcela na Equação (3.7) permanece positiva. Como não se permite que a variável $\alpha_k(n)$ se torne maior do que α_k^+, ela se mantém constante. Quando o somatório na Equação (3.7) se torna menor do que β, o termo de correção se torna negativo e consequentemente tem-se $\Delta\alpha_k(n) \triangleq \alpha_k(n+1) - \alpha_k(n) < 0$. Em outras palavras, $\alpha_k(n)$ passa a decrescer. Quando $\alpha_k(n)$ se torna negativo, o nó k deixa de ser amostrado ($\bar{t}_k(n) = 0$), o que por sua vez torna o termo de correção em (3.7) positivo. Consequentemente, $\alpha_k(n)$ começa a aumentar. Quanto maior for a média do erro quadrático na vizinhança do nó k, mais veloz é esse aumento. Após um determinado número de iterações, $\alpha_k(n)$ se torna positivo novamente e o nó k volta a ser amostrado, reiniciando o ciclo. A Figura 12 mostra uma representação esquemática do comportamento descrito.

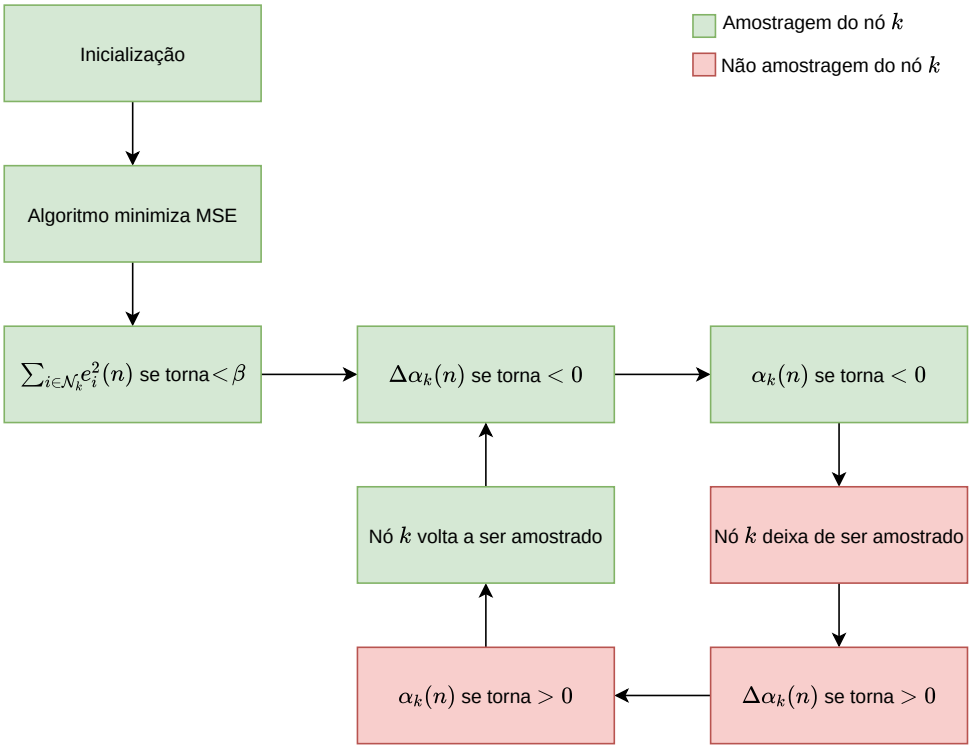


Figura 12: Representação esquemática do funcionamento do mecanismo de amostragem proposto. Fonte: Autor.

Assim, o mecanismo proposto reduz o número de nós amostrados em regime permanente, diminuindo o custo computacional. Se β for escolhido apropriadamente, essa redução não ocorre durante o transitório, preservando a taxa de convergência do dNLMS original. No entanto, cabe notar que há um ligeiro aumento no custo computacional durante o transitório, já que o mecanismo de amostragem requer que se efetue um certo número de somas e multiplicações. Isto será estudado com mais profundidade no Capítulo 4. Além disso, é importante mencionar que, quando o nó i é amostrado, ele deve transmitir $e_i^2(n)$ para os seus vizinhos. Entretanto, essa informação pode ser enviada junto com a estimativa local $\boldsymbol{\psi}_i$ para não aumentar o número de transmissões.

Para finalizar esta seção, é importante mencionar que o algoritmo descrito na Tabela 1 pode ser implementado com qualquer técnica para seleção dos pesos de combinação. Caso eles sejam selecionados por meio de um método adaptativo, como o algoritmo ACW, a atualização dos $\{c_{ik}(n)\}$ deveria ser incluída na Tabela 1. Em relação a isso, é importante mencionar que se o algoritmo ACW for empregado junto com o AS-dNLMS e a amostragem do nó k for interrompida por um longo período de tempo, o mecanismo de amostragem pode prejudicar a atualização dos pesos de combinação. Isso porque, neste caso, $\widehat{\sigma}_{kk}$ poderia tender a zero em (2.21) (página 17) devido ao fato de t_k permanecer igual a zero em (3.1). A fim de evitar

isso, para $j=k$, substitui-se $\boldsymbol{\psi}_j(n+1)$ em (2.21) por

$$\bar{\boldsymbol{\psi}}_k(n+1) \triangleq \bar{t}_k(n)\boldsymbol{\psi}_k(n+1) + [1-\bar{t}_k(n)]\bar{\boldsymbol{\psi}}_k(n).$$

3.2 O mecanismo adaptativo como uma técnica de censura

Com uma simples modificação, o algoritmo adaptativo de amostragem proposto pode ser empregado como uma estratégia de censura. Para obter essa versão alternativa do AS-dNLMS, basta não atualizar $\boldsymbol{\psi}_k$ quando o nó k não é amostrado, o que leva a

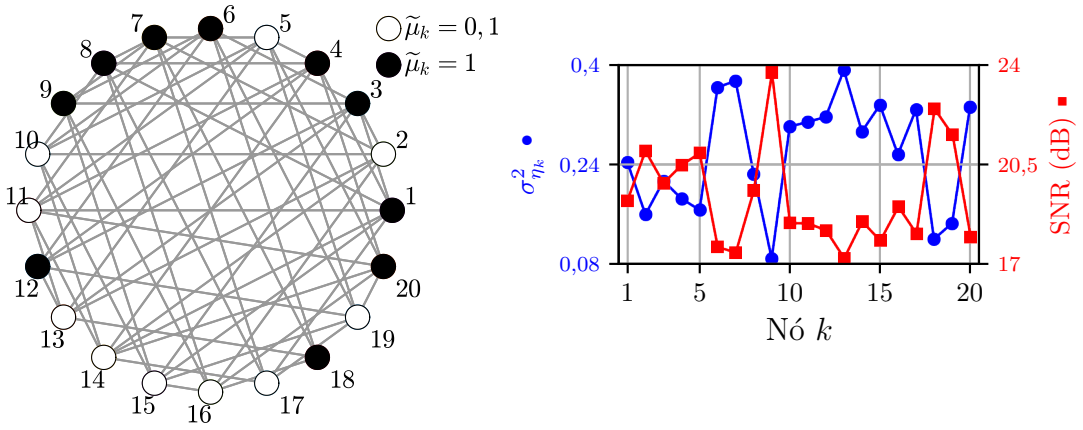
$$\boxed{\boldsymbol{\psi}_k(n+1) = [1-\bar{t}_k(n)]\boldsymbol{\psi}_k(n) + \bar{t}_k(n)\left[\mathbf{w}_k(n)+\mu_k(n)\mathbf{x}_k(n)e_k(n)\right].} \tag{3.9}$$

Considerando que os nós consigam armazenar informações recebidas anteriormente, isso permite diminuir o número de transmissões entre eles, já que neste caso $\boldsymbol{\psi}_k$ e ε_k^2 permanecem constantes quando $\bar{t}_k = 0$ e não há necessidade de enviá-los novamente. Dessa forma, quando o nó k não é amostrado, ele apenas recebe informação e realiza a etapa de combinação descrita pela Equação (2.15b) da página 15, podendo desligar seu transmissor. Esta versão do algoritmo é chamada de dNLMS com amostragem e censura adaptativas (*adaptive-sampling-and-censoring diffusion* NLMS – dNLMS). Ela permite uma redução no consumo de energia, além de diminuir o custo computacional em comparação com o dNLMS original.

3.3 Simulações computacionais

Nesta seção, são apresentados resultados de simulação para ilustrar o comportamento dos mecanismos propostos para amostragem e censura. Os resultados apresentados foram obtidos a partir de uma média de 100 realizações independentes. Para uma melhor visualização, as curvas foram filtradas por um filtro de média móvel com 64 coeficientes.

Nas simulações, considera-se o algoritmo dNLMS ATC e uma rede heterogênea com 20 nós. Dez deles utilizam $\widetilde{\mu}_k = 1$, enquanto os demais usam $\widetilde{\mu}_k = 0{,}1$, como mostrado na Figura 13a. O sinal $u_k(n)$ e o ruído $\eta_k(n)$ são gerados a partir de ruído gaussiano branco. Considera-se $\sigma_{u_k}^2 = 1$ para todo $k \in \{1, \cdots, V\}$, mas cada nó está sujeito a uma variância de ruído $\sigma_{\eta_k}^2$ diferente, como mostrado na Figura 13b, sendo que a mínima e a máxima variância do ruído observadas na rede são, respectivamente, $\sigma_{\min}^2 = 0{,}08$ e $\sigma_{\max}^2 = 0{,}4$. Também é mostrada na figura o valor da razão sinal-ruído (*signal-to-noise ratio* – SNR) em cada nó.



(a) (b)

Figura 13: (a) Rede utilizada nas simulações. Nós representados por círculos preenchidos utilizam o dNLMS ATC com $\widetilde{\mu}_k = 1$, enquanto que nós representados por círculos vazios usam o mesmo algoritmo com $\widetilde{\mu}_k = 0{,}1$. (b) Distribuição da variância $\sigma^2_{\eta_k}$ do ruído para $k \in \{1, \cdots, V\}$.
Fonte: Autor.

Os pesos de combinação são atualizados utilizando-se o algoritmo ACW com $\nu_k = 0{,}2$ para $k = 1, \cdots, V$ (TU; SAYED, 2011). Para os fatores de regularização, foram adotados os valores $\delta = \delta_c = 10^{-5}$. Por conveniência, o valor de β adotado em cada simulação é expresso em função de

$$\beta_r \triangleq \frac{\beta}{\sigma^2_{\max}}. \tag{3.10}$$

A razão para isso ficará clara no Capítulo 4, ao se analisar como escolher β adequadamente. Como indicador de desempenho, utiliza-se o MSD, dado por

$$\mathrm{MSD}(n) = \frac{1}{V}\sum_{k=1}^{V} \mathrm{E}\left\{\|\mathbf{w}^{\mathrm{o}}(n) - \mathbf{w}_k(n)\|^2\right\}. \tag{3.11}$$

Além disso, em algumas situações analisa-se também o MSE, dado por

$$\mathrm{MSE}(n) = \frac{1}{V}\sum_{k=1}^{V} \mathrm{E}\left\{e_k^2(n)\right\}. \tag{3.12}$$

A fim de facilitar o entendimento, esta seção está divida da seguinte forma. Na Subseção 3.3.1, compara-se o algoritmo AS-dNLMS com a técnica de amostragem aleatória da Figura 1 (página 4). Em seguida, uma comparação entre o algoritmo ASC-dNLMS e outras estratégias de censura propostas na literatura é apresentada na Subseção 3.3.2. Por sua vez, na Subseção 3.3.3, estuda-se a capacidade de rastreamento (*tracking*) das técnicas propostas. Finalmente, na Subseção 3.3.2, emprega-se o AS-dNLMS em um contexto de filtragem adaptativa aplicada a grafos.

3.3.1 Comparação com a amostragem aleatória

Nesta subseção, retoma-se a simulação da Figura 1 (página 4) a fim de comparar o comportamento do AS-dNLMS com o do algoritmo dNLMS com V_t nós amostrados aleatoriamente por iteração. Entretanto, dessa vez inverte-se a ordem dos elementos do vetor $\mathbf{w}^{\mathrm{o}}$ no meio de cada realização para simular uma mudança abrupta no ambiente. O algoritmo AS-dNLMS foi ajustado para apresentar aproximadamente o mesmo custo computacional que o dNLMS com $V_t = 5$ nós amostrados. Para isso, adotou-se $\beta_r = 1{,}6$ e $\mu_t = 0{,}06$. Nas Figuras 14a, 14b e 14c são apresentados, respectivamente, as curvas de MSD e o número médio de somas e multiplicações por iteração. Assim como na Figura 1, quanto mais nós são amostrados durante o transitório, mais rápida é a convergência. Como o algoritmo AS-dNLMS manteve a amostragem de todos os nós durante os transitórios, ele convergiu tão rápido quanto o dNLMS com todos os nós amostrados. Além disso, observa-se que ele foi capaz de detectar a mudança abrupta no sistema ótimo. Nas Figuras 14b e 14c observa-se que, durante os transitórios, o custo computacional do AS-dNLMS é ligeiramente maior do que o do algoritmo original, como era de se esperar. Entretanto, o número de operações por iteração cai drasticamente uma vez que se atinge o regime permanente.

3.3.2 Aplicação como técnica de censura

Nesta subseção, o algoritmo ASC-dNLMS é comparado com outras duas técnicas encontradas na literatura de censura: os algoritmos ACW-S (ACW-*Selective*) de (ARROYO-VALLES; MALEKI; LEUS, 2013) e EA-dNLMS (*Energy-Aware* dNLMS) de (GHAREHSHIRAN; KRISHNAMURTHY; YIN, 2013). Considerando que os nós sejam capazes de enviar informações para todos os seus vizinhos com uma única transmissão, apresentam-se na Figura 15a e 15b as curvas de MSD e o número médio de transmissões por iteração, respectivamente.

Os algoritmos foram ajustados para atingir aproximadamente o mesmo nível de MSD em regime permanente. Na Tabela 2 são mostrados os valores adotados para os parâmetros dos algoritmos. Cabe notar que o algoritmo EA-dNLMS apresenta um alto número de parâmetros que devem ser ajustados, o que pode ser difícil de se fazer. Em relação ao EA-dNLMS, considera-se a sua versão que permite a cada nó k receber informação mesmo quando não ele transmite para seus vizinhos (GHAREHSHIRAN; KRISHNAMURTHY; YIN, 2013). Além disso, adota-se para ele um passo normalizado segundo a Equação (2.9) da página 11. Para

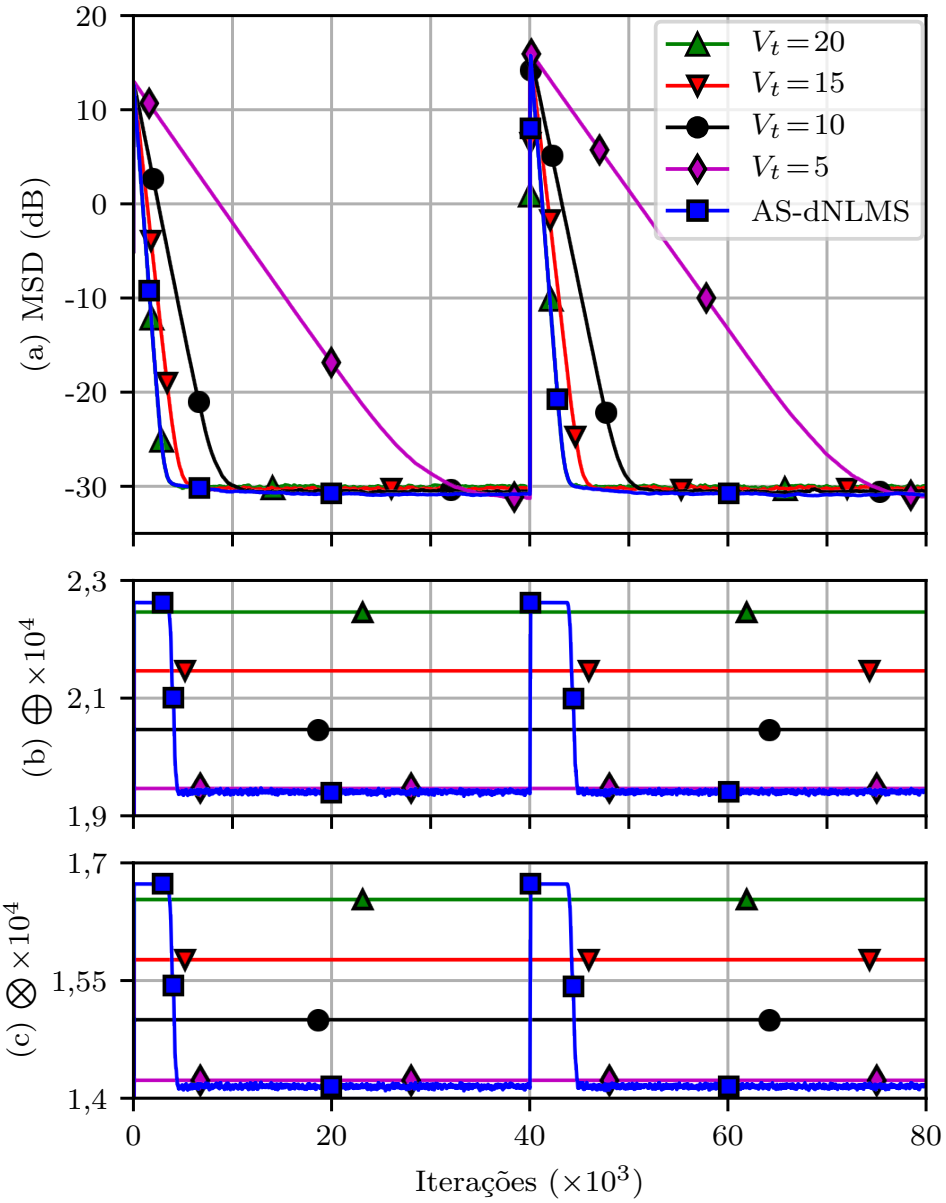


Figura 14: Comparação entre o dNLMS com diferentes quantidades V_t de nós amostrados e o AS-dNLMS ($\beta_r = 1{,}6$, $\mu_t = 0{,}06$). (a) Curvas de MSD, (b) Número médio de somas e (c) multiplicações por iteração.

Fonte: Autor.

efeito de comparação, também são apresentados resultados obtidos com o dNLMS original e com o caso não-cooperativo (i.e., os nós não trocam informação entre si).

Ao contrário do algoritmo AS-dNLMS, que mantém o desempenho em regime permanente do dNLMS, o ASC-dNLMS apresenta um ligeiro aumento no nível de MSD atingido em regime permanente. O mesmo ocorre para o ACW-S e para o EA-dNLMS, como pode ser visto na Figura 15a. É possível observar que o EA-dNLMS apresenta uma taxa de convergência significativamente mais lenta em comparação com os algoritmos ACW-S e ASC-dNLMS, os

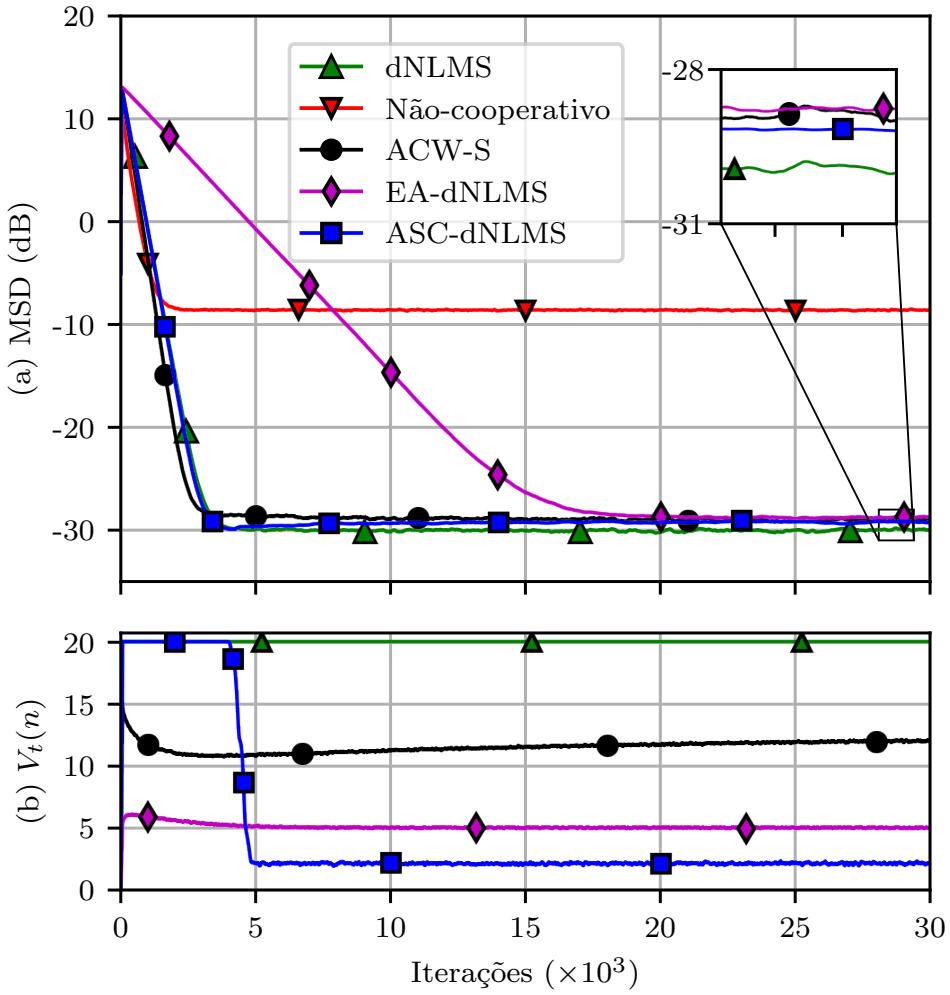


Figura 15: Comparação entre os algoritmos ASC-dNLMS, ACW-S e EA-dNLMS. Os parâmetros adotados são mostrados na Tabela 2. (a) Curvas de MSD. (b) Número médio de transmissões por iteração. Fonte: Autor.

quais convergem aproximadamente com a mesma taxa que o dNLMS original. Em contrapartida, nota-se na Figura 15b que o ACW-S leva a um número comparativamente mais elevado de transmissões e, portanto, a uma menor economia energética. Em regime permanente, tanto o ACW-S como o EA-dNLMS levam a mais transmissões do que o ASC-dNLMS, que mantém o número de comunicações durante o transitório mas reduz drasticamente uma vez atingido o regime permanente. Dessa forma, a técnica proposta leva a uma economia energética maior enquanto preserva a taxa de convergência do dNLMS original.

Tabela 2: Parâmetros dos algoritmos usados na simulação da Figura 15

ACW-S	$E_T = 1$, $E_R = 2$
EA-dNLMS	$E_{\text{Act}} = 33{,}5966 \cdot 10^{-3}$, $E_{\text{Tx}} = 15{,}16 \cdot 10^{-3}$, $K_{\ell,1} = 2$, $K_{\ell,2} = 0{,}5$, $K_g = 2$, $\gamma_g = 2$,$\gamma_\ell = 2$, $\delta = 0{,}5$, $\rho = 0{,}01$, $r = 2$
ASC-dNLMS	$\beta_r = 2{,}1$, $\mu_t = 0{,}0333$

Fonte: Autor.

3.3.3 Rastreamento do tipo "passeio aleatório"

Nesta Subseção, investiga-se o comportamento do algoritmo proposto em ambientes não-estacionários do tipo "passeio-aleatório" (*random-walk*, em inglês), em que o sistema ótimo $\mathbf{w}^{\mathrm{o}}(n)$ varia de acordo com

$$\mathbf{w}^{\mathrm{o}}(n) = \mathbf{w}^{\mathrm{o}}(n-1) + \mathbf{q}(n), \tag{3.13}$$

em que $\mathbf{q}(n)$ é um vetor coluna i.i.d. de média nula, comprimento M e matriz de autocovariância $\mathbf{Q} = \mathrm{E}\{\mathbf{q}(n)\mathbf{q}^{\mathrm{T}}(n)\}$ independente de qualquer outro sinal. Trata-se de um modelo comumente usado na literatura de filtragem adaptativa (HAYKIN, 2014; DINIZ, 2008; SAYED, 2008; FERNANDEZ-BES et al., 2017). Nos experimentos realizados, considera-se uma distribuição gaussiana para $\mathbf{q}(n)$ com $\mathbf{Q} = \sigma_q^2\mathbf{I}$.

Na Figura 16 são apresentados os resultados obtidos com o algoritmo AS-dNLMS e diferentes valores de β em função de $\mathrm{Tr}[\mathbf{Q}]$. Para efeito de comparação, também são mostrados os resultados obtidos com o dNLMS original. Na Figura 16a apresenta-se o nível de MSD atingido em regime permanente, na Figura 16b o número médio de nós amostrados por iteração e na Figura 16c o nível de MSE atingido em regime permanente. Os resultados foram obtidos calculando-se as médias dos dados de interesse durante as últimas 600 iterações de cada realização, após a convergência de cada algoritmo.

Na Figura 16a, pode-se observar que, em ambientes de variação lenta ($\mathrm{Tr}[\mathbf{Q}] = 10^{-8}$), o desempenho do AS-dNLMS é semelhante ao do dNLMS com todos os nós amostrados. No entanto, para $10^{-7} \leqslant \mathrm{Tr}[\mathbf{Q}] \leqslant 10^{-3}$, há uma degradação no desempenho em comparação com o dNLMS. Quanto maior o parâmetro β, mais intensa essa deterioração para um valor fixo de $\mathrm{Tr}[\mathbf{Q}]$. Para $\mathrm{Tr}[\mathbf{Q}] \leqslant 10^{-5}$ e um β fixo, essa deterioração em comparação com o dNLMS se intensifica com o aumento de $\mathrm{Tr}[\mathbf{Q}]$. Por outro lado, para $\mathrm{Tr}[\mathbf{Q}] > 10^{-5}$, a diferença no desempenho começa a diminuir à medida que as variações no sistema ideal se tornam mais rápidas. Isso pode ser explicado analisando as Figuras 16b e 16c. Observa-se que, quando o ambiente varia lenta ou moderadamente, o número de nós amostrados pelo AS-dNLMS não é afetado significativamente pelo aumento de $\mathrm{Tr}[\mathbf{Q}]$. Isso ocorre porque os efeitos das alterações no sistema ótimo são pequenos em comparação com os do ruído de medição para $\mathrm{Tr}[\mathbf{Q}] < 10^{-5}$ e, portanto, o MSE não aumenta visivelmente, como visto na Figura 16c. No entanto, à medida que essas variações se tornam mais rápidas, elas começam a afetar mais intensamente o erro de estimativa e o MSE começa a aumentar para $\mathrm{Tr}[\mathbf{Q}] \geqslant 10^{-5}$, levando a um aumento gradual

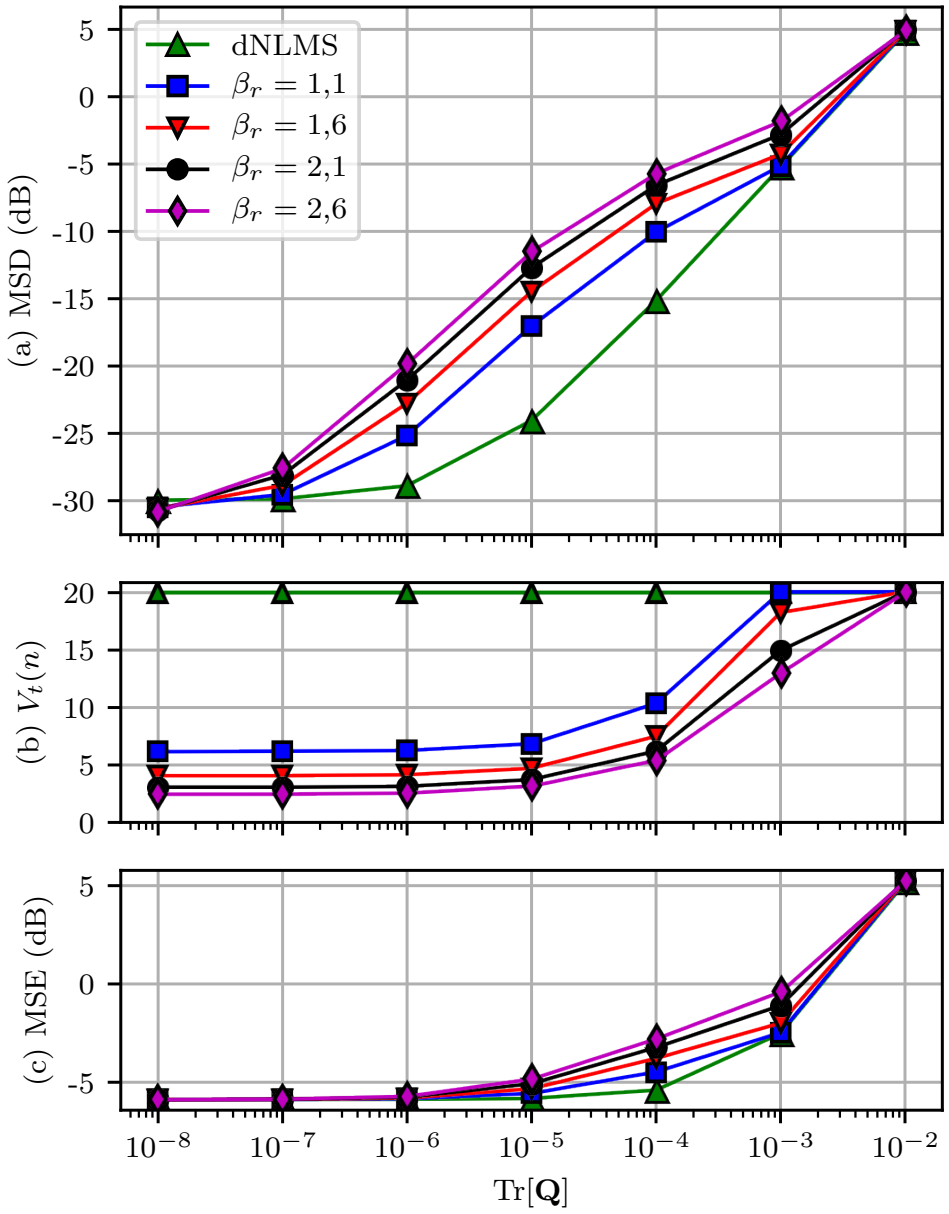


Figura 16: Resultados de simulação em um ambiente modelado pela Equação (3.13). (a) Níveis de MSD em regime permanente, (b) Número médio de nós amostrados, e (c) Níveis de MSE atingidos em regime permanente.

Fonte: Autor.

no número de nós amostrados na Figura 16b. Para $\text{Tr}[\mathbf{Q}] = 10^{-2}$, o algoritmo não deixa de amostrar nenhum dos nós para $\beta_r \leqslant 2{,}6$ e, portanto, seu desempenho corresponde ao do dNLMS.

Em seguida, repetiu-se o experimento da Figura 16 para os algoritmos ASC-dNLMS, ACW-S e EA-dNLMS com os parâmetros da Tabela 2. Os resultados são mostrados na Figura 17. Também são apresentados os resultados obtidos com o ASC-dNLMS com $\beta_r = 1{,}3$ e $\beta_r = 0{,}71$, que foram ajustados respectivamente para levar ao mesmo número de transmissões que os algoritmos EA-dNLMS e ACW-S para $\text{Tr}[\mathbf{Q}] \leqslant 10^{-6}$. Por fim, também são apresentados os

resultados obtidos com o algoritmo dNLMS. Observa-se na Fig. 17a que o ASC-dNLMS com $\beta = 0{,}84$ alcança um desempenho semelhante ao das outras soluções para $\mathrm{Tr}[\mathbf{Q}] = 10^{-8}$ e $\mathrm{Tr}[\mathbf{Q}] = 10^{-7}$. No entanto, para $\mathrm{Tr}[\mathbf{Q}] \geqslant 10^{-6}$, ele alcança níveis comparativamente mais elevados de MSD. Cabe notar que ele leva a menos transmissões do que qualquer outra solução nesses cenários. Com $\beta_r = 1{,}3$, o ASC-dNLMS tem um desempenho melhor em regime permanente do que o EA-dNLMS para $\mathrm{Tr}[\mathbf{Q}] \leqslant 10^{-7}$ e $\mathrm{Tr}[\mathbf{Q}] = 10^{-4}$, embora seu MSD seja mais alto para $\mathrm{Tr}[\mathbf{Q}] = 10^{-6}$ e $\mathrm{Tr}[\mathbf{Q}] = 10^{-5}$. Com $\beta_r = 0{,}71$, o ASC-dNLMS tem um desempenho melhor em regime permanente do que o ACW-S para $\mathrm{Tr}[\mathbf{Q}] \leqslant 10^{-7}$, enquanto o oposto ocorre para outros valores de $\mathrm{Tr}[\mathbf{Q}]$. Os resultados sugerem que o ASC-dNLMS geralmente supera o ACW-S e o EA-dNLMS em ambientes estacionários ou de variação lenta enquanto utiliza o mesmo número de transmissões. Além disso, nesses casos, ele pode obter um desempenho semelhante transmitindo comparativamente menos. No entanto, o ASC-dNLMS deve ser empregado com cautela em cenários nos quais o sistema ótimo varia rapidamente. Por fim, observa-se que é possível controlar o compromisso entre economia de energia e desempenho ajustando adequadamente o parâmetro β.

3.3.4 Aplicação em filtragem adaptativa baseada em grafos

Nesta subseção, aplica-se a técnica de amostragem proposta a um algoritmo adaptativo voltado a grafos. Considera-se ainda a rede da Figura 13a e se utiliza a sua matriz de adjacência não ponderada, normalizada pelo seu maior autovalor, como operador de deslocamento no grafo (SANDRYHAILA; MOURA, 2013b). Além disso, considera-se uma versão escalonada da distribuição de $\sigma^2_{\eta_k}$ mostrada na Figura 13b, a fim de se manter aproximadamente a mesma razão sinal-ruído das Subseções 3.3.1 a 3.3.3. Para isso, adotou-se $0{,}0018 \leqslant \sigma^2_{\eta_k} \leqslant 0{,}009$ para $k = 1, \cdots, V$. Para o sistema ótimo $\mathbf{w}^{\mathrm{o}}$, considera-se um vetor aleatório com $M = 10$ coeficientes uniformemente distribuídos no intervalo $[-1,1]$. Assim como foi feito na Subseção 3.3.1, no meio de cada realização inverte-se a ordem dos elementos do vetor para simular uma mudança no ambiente. Além disso, o AS-dNLMS foi ajustado para apresentar aproximadamente o mesmo custo computacional que o dNLMS com $V_t = 5$ nós amostrados. Para isso, adotou-se $\beta_r = 1{,}8$ e $\mu_t = 2$. As Figuras 18a, 18b e 18c apresentam respectivamente as curvas de MSD e o número médio de somas e multiplicações por iteração. Novamente, pode-se observar que o algoritmo AS-dNLMS detecta a mudança no sistema ótimo e converge com a mesma taxa que o dNLMS com todos os nós amostrados, já que mantém a amostragem de todos os nós durante

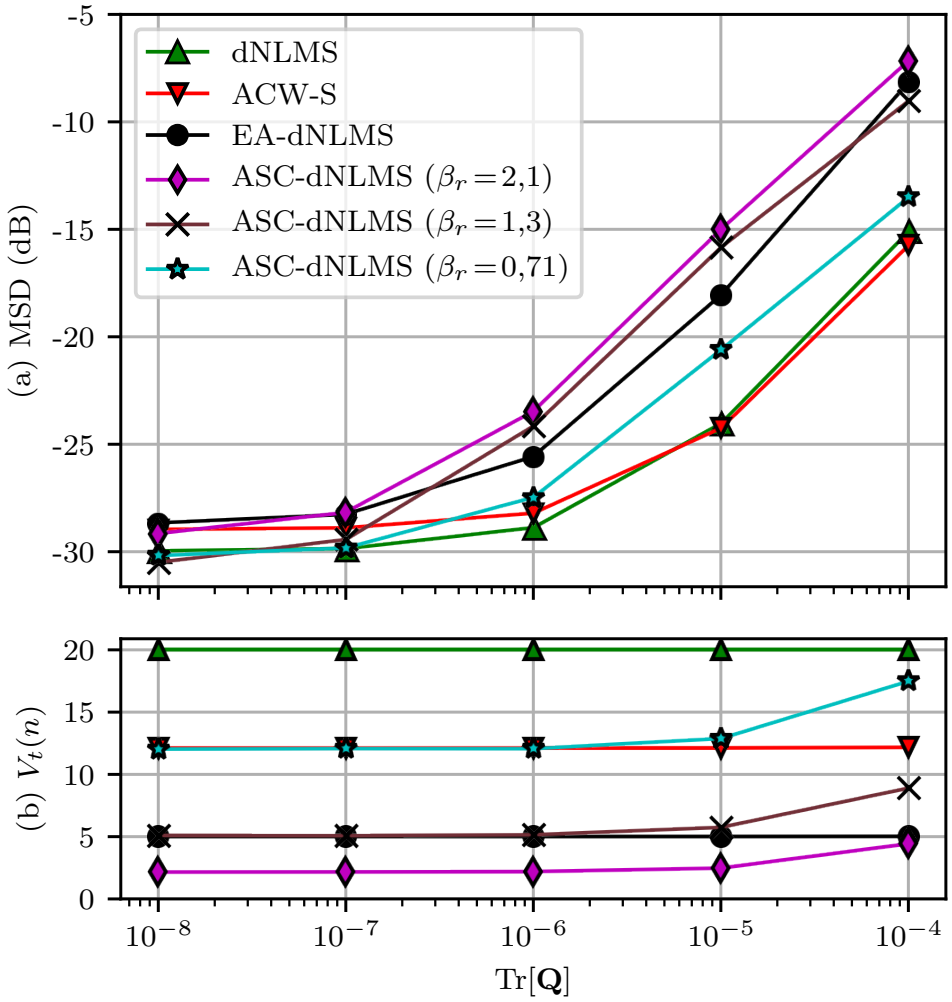


Figura 17: Resultados de simulação obtidos em um ambiente modelado pela Equação (3.13) pelos algoritmos mostrados na Tabela 2. (a) Níveis de MSD em regime permanente, (b) Número médio de transmissões.

Fonte: Autor.

os transitórios. Nota-se nas Figuras 18b e 18c que o custo computacional do AS-dNLMS é ligeiramente maior do que o do dNLMS original durante os transitórios, mas cai drasticamente uma vez que se atinge o regime permanente. Comparando-se as Figuras 14 e 18, nota-se que o mecanismo proposto se comporta de maneira semelhante quando aplicado à filtragem adaptativa voltada a grafos ou a redes de difusão "clássicas", uma vez que os resultados são análogos aos da Seção 3.3.1.

3.4 Conclusões

Mecanismos de amostragem e censura vêm sendo estudados pela comunidade de processamento de sinais desde a consolidação das redes de difusão adaptativas em aplicações envolvendo

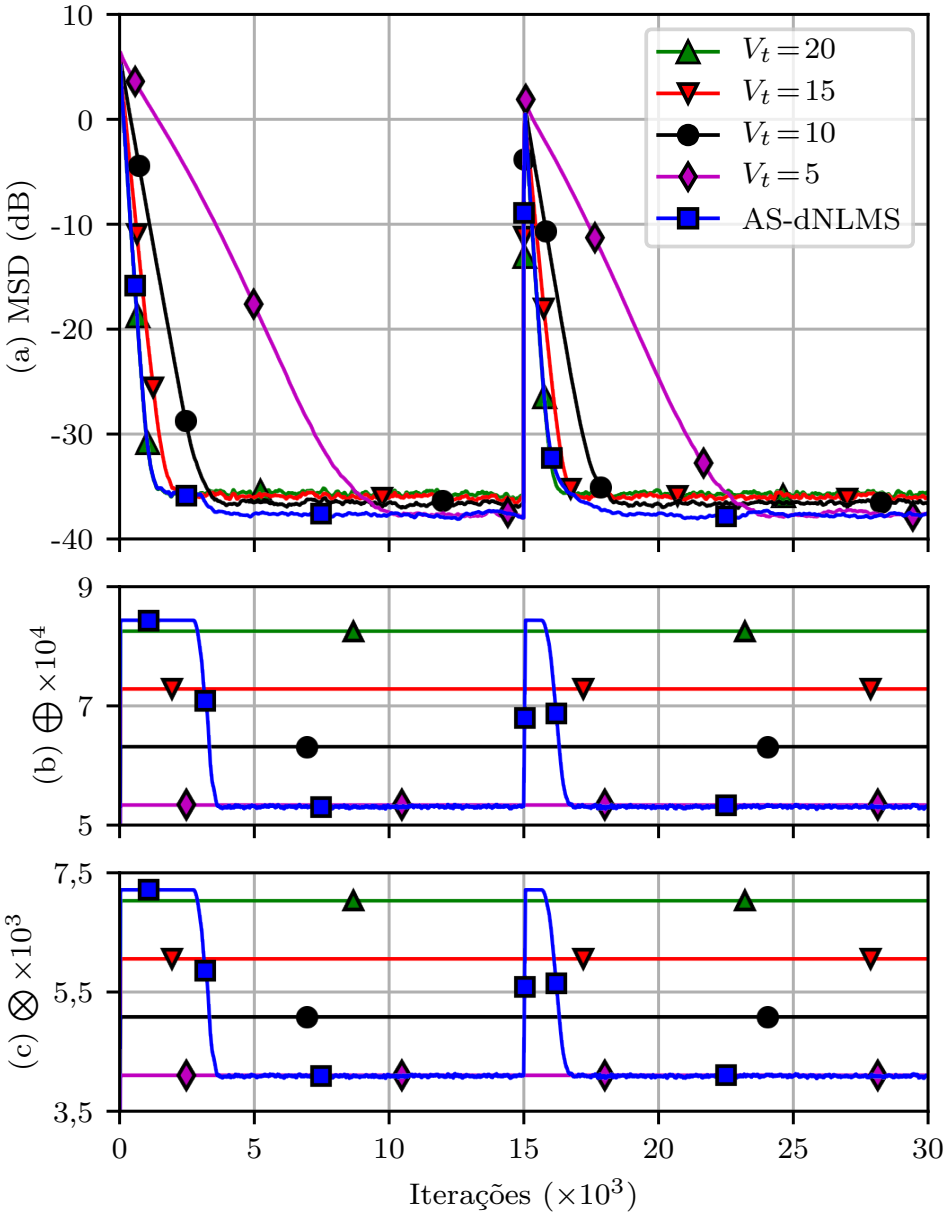


Figura 18: Comparação entre o dNLMS voltado a grafos com V_t nós amostrados aleatoriamente a cada iteração e o AS-dNLMS voltado a grafos ($\beta_r = 1{,}8$, $\mu_t = 2$). (a) Curvas de MSD, (b) Número médio de somas e (c) multiplicações por iteração.

Fonte: Autor.

redes de sensores. A procura por técnicas que estendam a vida útil das redes e viabilizem o seu uso sem comprometer excessivamente o seu desempenho vem motivando diferentes abordagens nessas áreas (ARABLOUEI et al., 2013; CHOUVARDAS; SLAVAKIS; THEODORIDIS, 2013; LOPES; SAYED, 2008a; TAKAHASHI; YAMADA, 2010; ZHAO; SAYED, 2012; XU; DE LAMARE; POOR, 2015; ARROYO-VALLES; MALEKI; LEUS, 2013; GHAREHSHIRAN; KRISHNAMURTHY; YIN, 2013; FERNANDEZ-BES et al., 2015; YANG et al., 2018; BERBERIDIS et al., 2015; LORENZO et al., 2018; LORENZO et al., 2017a).

Neste capítulo, foram propostos mecanismos adaptativos para amostragem e censura em redes de difusão. Os algoritmos obtidos, respectivamente denominados AS-dNLMS e ASC-dNLMS, usam as informações de mais nós quando o erro na rede é alto e de menos nós caso contrário. Os resultados de simulação da Figura 14 mostram que o AS-dNLMS consegue preservar o desempenho do algoritmo dNLMS original, enquanto reduz significativamente o custo computacional em regime permanente. Além disso, ele pode ser empregado tanto em redes de difusão adaptativas "clássicas", como nas baseadas em grafos, como visto na Subseção 3.3.4. Na Subseção 3.3.2, foi demonstrado que o ASC-dNLMS é capaz de economizar mais energia do que outras técnicas de censura encontradas na literatura, ao mesmo tempo em que obtém desempenho semelhante no regime permanente e preserva a taxa de convergência do dNLMS. É importante notar que as técnicas propostas devem ser empregadas com certa cautela em ambientes que variam rapidamente, pois seu desempenho pode se deteriorar em comparação com o dNLMS ou com as outras técnicas testadas, como visto na Subseção 3.3.3. No entanto, resultados encorajadores para ambientes com variações lentas indicam que os algoritmos AS-dNLMS e ASC-dNLMS são as soluções mais adequadas para esses casos, já que eles superam técnicas semelhantes nessas situações.

4 ANÁLISE DO ALGORITMO PROPOSTO

O bom desempenho do algoritmo proposto no Capítulo 3 depende de uma escolha apropriada para os seus parâmetros β e μ_t. Por esse motivo, neste capítulo é apresentada uma análise teórica para determinar como escolher β e μ_t adequadamente. Na Seção 4.1, estuda-se o impacto do parâmetro β no funcionamento do algoritmo e se obtêm condições necessárias e suficientes para garantir que os nós deixem de ser amostrados. Em seguida, na Seção 4.2, obtém-se um modelo para prever o número esperado de nós amostrados em função de β. Na Seção 4.3, analisa-se o efeito do parâmetro μ_t e se obtém uma regra que auxilia a sua escolha pelo projetista do filtro. Na Seção 4.4, investiga-se a economia em termos de custo computacional proporcionada pelos algoritmos propostos para censura e amostragem. Por sua vez, na Seção 4.5 são apresentados resultados de simulação para a validação da análise realizada. Finalmente, a Seção 4.6 encerra o capítulo com as suas principais conclusões.

4.1 O parâmetro β e seus efeitos

O parâmetro β desempenha um papel essencial no comportamento do algoritmo proposto. Ele influencia diretamente o número esperado de nós amostrados em regime permanente e o momento em que o mecanismo de amostragem começa a agir. Por este motivo, nesta seção os seus efeitos sobre o algoritmo são analisados e se obtêm regras para uma escolha embasada desse parâmetro.

Primeiramente, realiza-se uma análise do comportamento de $\alpha_k(n)$ enquanto o nó k é amostrado. Neste caso, pode-se substituir $\varepsilon_i^2(n)$ e $\beta\bar{t}_k(n)$ na Equação (3.7) da página 29 por $e_i^2(n)$ e β, respectivamente. Fazendo-se essas substituições, subtraindo $\alpha_k(n)$ de ambos os lados e tomando as esperanças, obtém-se

$$\mathrm{E}\{\Delta\alpha_k(n)\} = \mu_t \mathrm{E}\left\{\phi'\left[\alpha_k(n)\right]\left[\sum_{i\in\mathcal{N}_k} c_{ik}(n)e_i^2(n) - \beta\right]\right\}, \tag{4.1}$$

em que $\Delta\alpha_k(n) \triangleq \alpha_k(n+1) - \alpha_k(n)$. Para tornar a análise mais tratável, considera-se que os pesos $\{c_{ik}\}$ são constantes no tempo e determinísticos. Assim, as variáveis aleatórias presentes na Equação (4.1) são $\Delta\alpha_k(n)$, $\alpha_k(n)$ e $e_i^2(n)$. Por simplicidade, considera-se ainda que $\phi'\left[\alpha_k(n)\right]$ e

os erros de estimação $e_i(n)$ em (4.1) são estatisticamente independentes para $i \in \mathcal{N}_k$. Resultados de simulação mostram que essas hipóteses são razoáveis e que as expressões obtidas permanecem válidas para o caso de pesos adaptativos, como será visto na Seção 4.5. Dessa forma, (4.1) pode ser reescrita como

$$\mathrm{E}\{\Delta\alpha_k(n)\} = \mu_t \mathrm{E}\{\phi'\left[\alpha_k(n)\right]\}\left[\sum_{i\in\mathcal{N}_k} c_{ik}\mathrm{E}\{e_i^2(n)\} - \beta\right]. \tag{4.2}$$

Para que o nó k deixe de ser amostrado, α_k deve decrescer ao longo das iterações até se tornar negativo. Como $\mu_t \mathrm{E}\{\phi'\left[\alpha_k(n)\right]\}$ é sempre positivo, para garantir que $\Delta\alpha_k(n)$ seja negativo na média, β deve satisfazer

$$\beta > \sum_{i\in\mathcal{N}_k} c_{ik}\mathrm{E}\{e_i^2(n)\}. \tag{4.3}$$

Considerando que o algoritmo estime apropriadamente o filtro ótimo, é razoável assumir que, em regime permanente, $\mathrm{E}\{e_i^2(n)\} \approx \sigma_{\eta_i}^2$, de modo que se tem

$$\sigma_{\min}^2 \leqslant \sum_{i\in\mathcal{N}_k} c_{ik}\mathrm{E}\{e_i^2(n)\} \leqslant \sigma_{\max}^2, \tag{4.4}$$

em que $\sigma_{\min}^2 \triangleq \min_i \sigma_{\eta_i}^2$ e $\sigma_{\max}^2 \triangleq \max_i \sigma_{\eta_i}^2$, $i = 1, \cdots, V$. Portanto, a condição

$$\boxed{\beta > \sigma_{\min}^2} \tag{4.5}$$

é **necessária mas não suficiente** para que os nós deixem de ser amostrados na média em regime permanente. Similarmente, a condição

$$\boxed{\beta > \sigma_{\max}^2} \tag{4.6}$$

é **suficiente mas não necessária** para garantir que os nós deixem de ser amostrados na média em regime permanente. Além disso, essa condição garante que todo nó deixe de ser amostrado em alguma iteração durante o regime permanente. Quando $\sigma_{\min}^2 = \sigma_{\max}^2$, ou seja, todos os nós estão sujeitos à mesma variância de ruído, as Condições (4.5) e (4.6) coincidem e formam uma condição necessária e suficiente.

A condição estabelecida pela Expressão (4.6) é a razão pela qual é conveniente expressar os valores escolhidos para β em função de $\sigma_{\max}^2$, como feito nas simulações do Capítulo 3. Cabe ressaltar ainda que a Condição (4.5) foi satisfeita em todas as simulações desse capítulo. O menor valor de $\beta_r = \beta/\sigma_{\max}^2$ utilizado foi $\beta_r = 0{,}71$, o que corresponde a um β maior do que

σ^2_{min}, pois, na Figura 13b (página 33), observa-se que $\sigma^2_{\text{min}} = 0{,}2\sigma^2_{\text{max}}$.

Além disso, dado um determinado valor de β, pode-se analisar quando o mecanismo de amostragem começará a agir em termos do MSE. Em (4.2), observa-se que $\text{E}\{\Delta\alpha_k(n)\} \geqslant 0$ contanto que $\sum_{i\in\mathcal{N}_k} c_{ik}(n)\text{E}\{e_i^2(n)\} \geqslant \beta$. Como não se permite que $\alpha_k(n)$ se torne maior que α^+, conclui-se que $\text{E}\{\alpha_k(n)\} = \alpha^+$ por $k = 1, \cdots, V$ contanto que $\text{MSE}_{\text{min}}(n) > \beta$, em que $\text{MSE}_{\text{min}}(n) \triangleq \min_i \text{E}\{e_i^2(n)\}$, $i = 1, \cdots, V$. Em outras palavras, na média, o mecanismo de amostragem não age enquanto $\text{MSE}_{\text{min}}(n)$ permanecer maior que β. Consequentemente, nenhum nó deixará de ser amostrado na média durante esse período. Além disso, quanto maior o β, mais cedo $\text{E}\{\alpha_k(n)\}$ começa a diminuir para $k = 1, \cdots, V$ e mais cedo os nós deixam de ser amostrados.

4.2 Número esperado de nós amostrados em regime permanente

Considerando que (4.6) seja satisfeita, é possível estimar um limite inferior e um limite superior para o número esperado $\text{E}\{V_t\}$ de nós amostrados em regime permanente. Para isso, considera-se que, durante o regime permanente, $\overline{t}_k(n)$ é uma variável aleatória de Bernoulli que pode assumir o valores 1 com probabilidade p_{t_k} ou 0 com probabilidade $1 - p_{t_k}$, $0 \leqslant p_{t_k} \leqslant 1$. Assim, pode-se escrever

$$V p_{t_{\text{min}}} \leqslant \text{E}\{V_t\} \leqslant V p_{t_{\text{max}}}, \tag{4.7}$$

em que $p_{t_{\text{min}}}$ e $p_{t_{\text{max}}}$ são respectivamente os limites inferior e superior para p_{t_k}, $k \in \{1, \cdots, V\}$.

Convém recordar que, para cada nó k da rede, o mecanismo de amostragem apresenta uma natureza cíclica durante o regime permanente. Dessa forma, pode-se aproximar p_{t_k} pelo "ciclo de trabalho esperado" do mecanismo no nó k. Em outras palavras,

$$\widehat{p}_{t_k} = \frac{\theta_k}{\theta_k + \overline{\theta}_k}, \tag{4.8}$$

em que θ_k denota o número esperado de iterações por ciclo em que o nó k é amostrado e $\overline{\theta}_k$ representa o número esperado de iterações por ciclo em que isso não ocorre.

Como o objetivo é apenas obter aproximações para $p_{t_{\text{min}}}$ e $p_{t_{\text{max}}}$, não é necessário estimar (4.8) para todo nó k da rede. Em vez disso, basta obter estimativas para os limites superior e inferior de θ_k e $\overline{\theta}_k$, denotados respectivamente por θ_{max}, θ_{min}, $\overline{\theta}_{\text{max}}$ e $\overline{\theta}_{\text{min}}$. Para estimar θ_{max} e θ_{min}, deve-se compreender sob quais circunstâncias o nó k passa o maior (ou menor) número de

iterações sendo amostrado na média. Fazendo-se a mesma análise para o caso em que o nó k não é amostrado, pode-se estimar $\overline{\theta}_{\max}$ e $\overline{\theta}_{\min}$.

Uma das maneiras de se identificar tais circunstâncias consiste em analisar quais são os valores máximo e mínimo que $\mathrm{E}\{\alpha_k(n)\}$ e $\mathrm{E}\{\Delta\alpha_k(n)\}$ podem atingir durante o regime permanente. Para isso, suponha inicialmente que em uma dada iteração n, $\alpha_k(n)$ seja negativo e próximo de zero. Nesse caso, fazendo-se $\alpha_k(n) = 0$ na Equação (3.7) da página 29 e tomando-se as esperanças, obtém-se

$$\mathrm{E}\{\alpha_k(n+1)|\alpha_k(n) \lessapprox 0\} = \mu_t \phi_0' \sum_{i \in \mathcal{N}} c_{ik} \mathrm{E}\{\varepsilon_i^2(n)\}, \tag{4.9}$$

em que $\phi_0' = \phi'[0]$. Assim, $\mathrm{E}\{\alpha_k(n+1)|\alpha_k(n) \lessapprox 0\} > 0$ e a amostragem do nó k é retomada na iteração $n+1$, o que acarreta $\mathrm{E}\{\Delta\alpha_k(n+1)|\alpha_k(n) \lessapprox 0\} < 0$ por conta de (4.2) e (4.6). Portanto, a partir do instante $n+1$, α_k passa a decrescer até se tornar negativo novamente, implicando que (4.9) representa o maior valor que α_k pode atingir na média em regime permanente. Além disso, considerando-se que $\sigma_{\min}^2 \leqslant \mathrm{E}\{\varepsilon_i^2(n)\} \leqslant \sigma_{\max}^2$ para $i = 1, \cdots, V$, a Equação (4.9) fornece um valor diferente para cada nó k dentro do intervalo

$$\mu_t \phi_0' \sigma_{\min}^2 \leqslant \alpha_{k_{\max}}^{\text{s.s.}} \leqslant \mu_t \phi_0' \sigma_{\max}^2, \tag{4.10}$$

em que $\alpha_{k_{\max}}^{\text{s.s.}}$ denota o maior valor que $\alpha_k(n)$ pode atingir em regime permanente.

Analogamente, suponha agora que em uma dada iteração n, $\alpha_k(n)$ seja positivo e próximo de zero. Nesse caso, fazendo-se $\alpha_k(n) = 0$ em (3.7) e tomando-se as esperanças, obtém-se

$$\mathrm{E}\{\alpha_k(n+1)|\alpha_k(n) \gtrapprox 0\} = \mu_t \phi_0' \mathrm{E}\left\{ \sum_{i \in \mathcal{N}_k} c_{ik} \varepsilon_i^2(n) - \beta \right\}. \tag{4.11}$$

Portanto, na iteração $n+1$ o nó k deixa de ser amostrado. Dessa forma, tem-se que $\mathrm{E}\{\alpha_k(n+1)|\alpha_k(n) \gtrapprox 0\} < 0$ e $\mathrm{E}\{\Delta\alpha_k(n)|\alpha_k(n+1) \gtrapprox 0\} > 0$. Consequentemente, (4.11) fornece o menor valor que α_k pode assumir na média em regime permanente. Além disso, para cada nó k, (4.11) fornece um valor diferente dentro do intervalo

$$\mu_t \phi_0' (\sigma_{\min}^2 - \beta) \leqslant \alpha_{k_{\min}}^{\text{s.s.}} \leqslant \mu_t \phi_0' (\sigma_{\max}^2 - \beta), \tag{4.12}$$

em que $\alpha_{k_{\min}}^{\text{s.s.}}$ denota o menor valor que $\mathrm{E}\{\alpha_k(n)\}$ pode assumir em regime permanente.

Após estimar os valores máximo e mínimo que $\alpha_k(n)$ pode atingir na média em regime permanente, resta repetir o procedimento para $\Delta\alpha_k(n)$. Constata-se a partir de (4.10) e (4.12)

que $\mathrm{E}\{\alpha_k(n)\}$ oscila em torno de $\mathrm{E}\{\alpha_k(n)\} = 0$ durante o regime permanente, o que se deve à natureza cíclica do mecanismo de amostragem. Por isso, substitui-se $\phi'[\alpha_k(n)]$ em (4.2) pela sua aproximação de Taylor de primeira ordem ao redor desse ponto, que é simplesmente igual à constante ϕ'_0. Com isso, quando o nó k é amostrado ($\bar{t}_k(n) = 1$), subtraindo-se $\alpha_k(n)$ de ambos os lados em (4.2) e tomando-se as esperanças, obtém-se

$$-\mu_t\phi'_0(\beta-\sigma^2_{\min}) \leqslant \mathrm{E}\{\Delta\alpha_k(n)\} \leqslant -\mu_t\phi'_0(\beta-\sigma^2_{\max}) < 0. \tag{4.13}$$

Analogamente, quando o nó não é amostrado ($\bar{t}_k(n) = 0$),

$$\mu_t\phi'_0\sigma^2_{\min} \leqslant \mathrm{E}\{\Delta\alpha_k(n)\} \leqslant \mu_t\phi'_0\sigma^2_{\max}. \tag{4.14}$$

Portanto, em ambos os casos há um limite inferior e um limite superior para $\mathrm{E}\{\Delta\alpha_k(n)\}$ em regime permanente.

Com base em (4.13) e (4.14) e considerando-se $\mathrm{E}\{\Delta\alpha_k(n)\}$ constante no tempo, é possível chegar a valores extremos para θ_k e $\overline{\theta}_k$ em regime permanente. Assim, dada uma certa iteração n_0 durante o regime permanente, considera-se que vale o modelo

$$\mathrm{E}\{\alpha_k(n_0 + \theta_k)\} = \mathrm{E}\{\alpha_k(n_0)\} + \theta_k\mathrm{E}\{\Delta\alpha_k(n)\}. \tag{4.15}$$

Para estimar o limite superior $\theta_{\max}$ para θ_k, considera-se que $\mathrm{E}\{\alpha_k(n_0)\} = \mathrm{E}\{\alpha^{\text{s.s.}}_{k_{\max}}\}$ e calcula-se o número esperado de iterações para que $\mathrm{E}\{\alpha_k(n)\}$ se torne negativo no caso em que o nó é amostrado pelo maior número de iterações. Isso ocorre quando $\mathrm{E}\{\alpha_k(n_0)\} = \mu_t\phi'_0\sigma^2_{\max}$, que é o limite superior para $\mathrm{E}\{\alpha^{\text{s.s.}}_{k_{\max}}\}$ e $\mathrm{E}\{\Delta\alpha_k(n)\} = -\mu_t\phi'_0(\beta - \sigma^2_{\max})$, que é o valor menos negativo para $\mathrm{E}\{\Delta\alpha_k(n)\}$ quando $\bar{t}_k(n) = 1$ de acordo com (4.13).

Fazendo-se $\theta_k = \theta_{\max}$ e $\mathrm{E}\{\alpha_k(n_0 + \theta_{\max})\} = 0$ em (4.15), após algumas manipulações algébricas obtém-se

$$\theta_{\max} = \max\left\{\frac{\sigma^2_{\max}}{\beta - \sigma^2_{\max}},\ 1\right\}, \tag{4.16}$$

em que se leva em conta o fato de que o nó deve ser amostrado em pelo menos uma iteração por ciclo. Analogamente, pode-se aplicar (4.15) para o limite inferior $\theta_k = \theta_{\min}$, obtendo-se

$$\theta_{\min} = \max\left\{\frac{\sigma^2_{\min}}{\beta - \sigma^2_{\min}},\ 1\right\}. \tag{4.17}$$

Para $\overline{\theta}_k$, substitui-se θ_k em (4.15) por $\overline{\theta}_k$ e considera-se que em uma dada iteração n_0,

$\mathrm{E}\{\alpha_k(n_0)\} = \mathrm{E}\{\alpha^{\text{s.s.}}_{k_{\min}}\}$. Dessa forma, o limite superior $\overline{\theta}_{\max}$ para $\overline{\theta}_k$ pode ser obtido fazendo-se $\mathrm{E}\{\alpha_k(n_0)\} = \mu_t\phi'_0\sigma^2_{\min}$, que é o limite inferior para $\mathrm{E}\{\alpha^{\text{s.s.}}_{k_{\min}}\}$ e $\mathrm{E}\{\Delta\alpha_k(n)\} = \mu_t\phi'_0\sigma^2_{\min}$, que é o menor valor para $\mathrm{E}\{\Delta\alpha_k(n)\}$ quando $\overline{t}_k(n) = 0$ segundo (4.14). Obtém-se então

$$\overline{\theta}_{\max} = \max\left\{\frac{\beta - \sigma^2_{\min}}{\sigma^2_{\min}},\ 1\right\}. \tag{4.18}$$

Analogamente, obtém-se

$$\overline{\theta}_{\min} = \max\left\{\frac{\beta - \sigma^2_{\max}}{\sigma^2_{\max}},\ 1\right\}. \tag{4.19}$$

como estimativa para o limite inferior $\overline{\theta}_{\min}$ de $\overline{\theta}_k$.

Utilizando-se (4.8), pode-se então estimar $p_{t_{\min}}$ e $p_{t_{\max}}$ respectivamente como

$$\widehat{p}_{t_{\min}} = \frac{\theta_{\min}}{\theta_{\min} + \overline{\theta}_{\max}} \tag{4.20}$$

e

$$\widehat{p}_{t_{\max}} = \frac{\theta_{\max}}{\theta_{\max} + \overline{\theta}_{\min}}. \tag{4.21}$$

Se $\beta < 2\sigma^2_{\min}$, as Equações (4.17) e (4.18) fornecem, respectivamente, $\theta_{\min} = \sigma^2_{\min}/(\beta - \sigma^2_{\min})$ e $\overline{\theta}_{\max} = 1$. Em contrapartida, para $\beta \geqslant 2\sigma^2_{\min}$, (4.17) e (4.18) resultam em $\theta_{\min} = 1$ e $\overline{\theta}_{\max} = (\beta - \sigma^2_{\min})/\sigma^2_{\min}$, respectivamente. Em ambos os casos, ao se fazer essas substituições em (4.20), obtém-se

$$\widehat{p}_{t_{\min}} = \frac{\sigma^2_{\min}}{\beta}. \tag{4.22}$$

Analogamente, com base em (4.16), (4.19) e (4.21), obtém-se

$$\widehat{p}_{t_{\max}} = \frac{\sigma^2_{\max}}{\beta}. \tag{4.23}$$

Assim, substituindo-se (4.22) e (4.23) em (4.7), finalmente se obtém

$$V\frac{\sigma^2_{\min}}{\beta} \leqslant \mathrm{E}\{V_t\} \leqslant V\frac{\sigma^2_{\max}}{\beta}. \tag{4.24}$$

Para $\beta < \sigma^2_{\max}$, (4.24) estima um limite superior para $\mathrm{E}\{V_t\}$ que é maior do que o número total V de nós na rede, o que não é conveniente. Entretanto, esse resultado pode ser generalizado para qualquer valor de $\beta > 0$ ao ser reescrito como

$$\boxed{V \cdot \min\left\{1,\ \frac{\sigma^2_{\min}}{\beta}\right\} \leqslant \mathrm{E}\{V_t\} \leqslant V \cdot \min\left\{1,\ \frac{\sigma^2_{\max}}{\beta}\right\}.} \tag{4.25}$$

Substituindo-se $\beta < \sigma^2_{\min}$ em (4.25), conclui-se que $\mathrm{E}\{V_t\} = V$, o que está de acordo com o fato de (4.5) ser uma condição necessária para a redução do número de nós amostrados. Analogamente, substituindo-se $\beta > \sigma^2_{\max}$, conclui-se que $\mathrm{E}\{V_t\} < V$, que está de acordo com (4.6) ser uma condição suficiente. Além disso, observa-se que quanto maior o parâmetro β, menor a quantidade de nós amostrados na média durante o regime permanente, como esperado. Uma vez que há um compromisso entre a capacidade de rastreamento e os ganhos em termos de custo computacional fornecidos pelo mecanismo de amostragem, deve-se tomar cuidado para não escolher valores excessivamente altos para β, pois eles podem deteriorar o desempenho do algoritmo em ambientes não estacionários. Resultados de simulação sugerem que, se $\beta \leqslant 5\sigma^2_{\max}$, o bom comportamento do algoritmo é mantido. Além disso, cabe notar que os limites superior e inferior coincidem quando $\sigma^2_{\min} = \sigma^2_{\max}$. Por fim, é interessante ressaltar que o passo μ_t não afeta o número de nós amostrados contanto que $\mu_t > 0$. Cabe ressaltar que a adoção de passos elevados pode fazer com que os nós deixem de ser amostrados enquanto o algoritmo ainda não atingiu o regime permanente em termos de MSD e/ou MSE, o que pode impactar a taxa de convergência e, consequentemente, a quantidade de nós amostrados por iteração durante esse período. Entretanto, como a Expressão (4.25) foi obtida considerando-se o algoritmo em regime permanente, isso não afeta o resultado analítico obtido.

4.3 Escolha do passo μ_t

Nesta seção, procura-se analisar como escolher o passo μ_t apropriadamente. Para isso, estuda-se o quão rapidamente os nós deixam de ser amostrados com base no valor escolhido para esse parâmetro. Matematicamente, isso se traduz em analisar o quão rápido se chega a $\mathrm{E}\{\alpha_k(n)\} \leqslant 0$ partindo-se da inicialização $\alpha_k(0) = \alpha^+$, $k = 1, \cdots, V$.

Usando (4.2) e (4.4), obtém-se

$$\mathrm{E}\{\Delta\alpha_k(n)\} \leqslant \mu_t \mathrm{E}\left\{\phi'\left[\alpha_k(n)\right]\right\}(\sigma^2_{\max} - \beta). \tag{4.26}$$

Para simplificar (4.26), aproxima-se $\phi'\left[\alpha_k(n)\right]$ no intervalo $[0, \alpha^+]$ por uma reta que cruza os pontos $(0, \phi'_0)$ e $(\alpha^+, \phi'_{\alpha^+})$, em que ϕ'_{α^+} denota o valor de $\phi'\left[\alpha_k(n)\right]$ para $\alpha_k = \alpha^+$. Esta aproximação é dada por

$$\phi'\left[\alpha_k(n)\right] \approx \xi\alpha_k(n) + \phi'_0, \tag{4.27}$$

em que $\xi = [\phi'_{\alpha^+} - \phi'_0]/\alpha^+$. Para $\alpha^+ = 4$, esta é uma boa aproximação já que o erro quadrático

médio no intervalo $[0, \alpha^+]$ é da ordem de

$$\frac{1}{\alpha^+}\int_0^{\alpha^+}\left[\phi'(\alpha_k)-(\xi\alpha_k+\phi'_0)\right]^2 d\alpha_k \approx 5\times 10^{-4}.$$

Na Figura 19, é mostrada a comparação entre ϕ'_{α_k} e a aproximação utilizada.

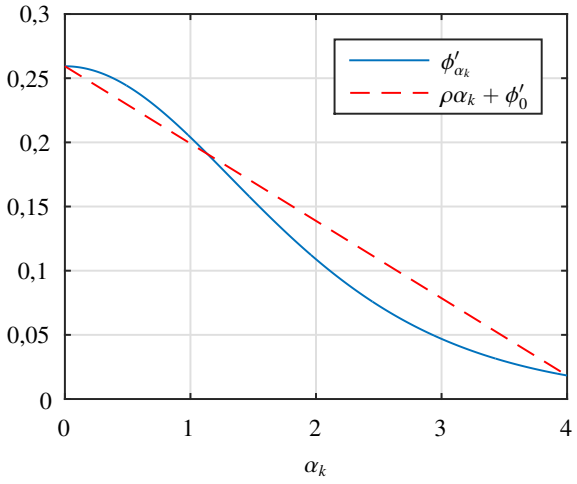


Figura 19: Comparação gráfica entre ϕ'_{α_k} e a aproximação da Equação (4.27).
Fonte: Autor.

Substituindo-se (4.27) em (4.26), obtém-se

$$\mathrm{E}\{\alpha_k(n+1)\} \lessapprox \mathrm{E}\{\alpha_k(n)\}(1+\xi\rho)+\phi'_0\rho, \tag{4.28}$$

em que $\rho = \mu_t(\sigma^2_{\max}-\beta)$. Como foi assumido que $\mathrm{E}\{\alpha_k(n)\}\approx\alpha^+$ durante o transitório, pode-se denotar a primeira iteração do regime permanente por n_0 e definir $n_0+\Delta n \triangleq n+1$. Então, considerando $\mathrm{E}\{\alpha_k(n_0)\}\approx\alpha^+$ em (4.28) e aplicando essa equação recursivamente, obtém-se

$$\mathrm{E}\{\alpha_k(n_0+\Delta n)\} \lessapprox \alpha^+(1+\xi\rho)^{\Delta n}+\phi'_0\rho\sum_{\ell=0}^{\Delta n-1}(1+\xi\rho)^{\ell}. \tag{4.29}$$

Após algumas manipulações algébricas, chega-se a

$$\mathrm{E}\{\alpha_k(n_0+\Delta n)\} \lessapprox [(\xi\alpha^+ + \phi'_0)(1+\xi\rho)^{\Delta n}-\phi'_0]/\xi. \tag{4.30}$$

Fazendo $\mathrm{E}\{\alpha_k(n_0+\Delta n)\}$ igual a zero em (4.30), obtém-se

$$\boxed{\mu_t \lessapprox \frac{\alpha^+}{(\beta-\sigma^2_{\max})(\phi'_0-\phi'_{\alpha^+})}\left[\left(\frac{\phi'_o}{\phi'_{\alpha^+}}\right)^{\frac{1}{\Delta n}}-1\right].} \tag{4.31}$$

Observa-se que quanto menor o valor escolhido para Δn, mais elevado se torna o limitante

inferior para μ_t em (4.31), o que é razoável. Conforme β se aproxima de $\sigma^2_{\max}$, a Expressão (4.31) passa a fornecer valores cada vez maiores. Como (4.6) é uma condição suficiente, os nós podem deixar de ser amostrados mesmo para $\beta \leqslant \sigma^2_{\max}$. Se $\beta \approx \sigma^2_{\max}$ e $\sigma^2_{\min} < \sigma^2_{\max}$, (4.31) pode superestimar o valor de μ_t necessário para que os nós deixem de ser amostrados em até Δn iterações. Entretanto, isso não invalida o resultado obtido, já que o objetivo consiste simplesmente em obter um passo de adaptação que garanta a interrupção da amostragem em no máximo Δn iterações.

4.4 Custo computacional

Como visto na Seção 4.2, se (4.6) for satisfeita, o mecanismo proposto leva a uma redução no número esperado de nós amostrados. No entanto, isso não garante necessariamente uma vantagem em termos de custo computacional, uma vez que o algoritmo de amostragem também requer um certo número de operações. Analisando a Tabela 1, vê-se que o mecanismo de amostragem requer $|\mathcal{N}_k| + 1 + \bar{t}_k(n)$ somas, $|\mathcal{N}_k| + 1$ multiplicações e duas comparações por iteração para cada nó k amostrado na rede. No entanto, quando o nó k não é amostrado, o AS-dNLMS não precisa calcular $\mathbf{x}_k(n)$, $\mathbf{w}_k(n)$, $e_k(n)$ e $\mu_k(n)$, exigindo $2M - \mathcal{N}_k$ e multiplicações, $2M - |\mathcal{N}_k| + 1$ somas e uma divisão a menos que o dNLMS original. Esses resultados estão resumidos na Tabela 3 para ambos os algoritmos com ACW aplicados ao processamento distribuído de sinais clássico. Cabe notar que aqui se considera uma implementação de $\phi'[\alpha_k(n)]$ através de uma tabela de consulta, o que não é levado em consideração na Tabela 3.

Tabela 3: Custo computacional dos algoritmos dNLMS e AS-dNLMS para o caso clássico com pesos ACW: número de operações por nó k

Operação	dNLMS	AS-dNLMS
Multiplicações ($\bigotimes$)	$M(3 + \lvert\mathcal{N}_k\rvert) + 4$	$\bar{t}_k(n)(2M + 2) + M(1 + \lvert\mathcal{N}_k\rvert) + \lvert\mathcal{N}_k\rvert + 4$
Somas ($\bigoplus$)	$M(3 + \lvert\mathcal{N}_k\rvert) + 3$	$\bar{t}_k(n)(2M + 2) + M(\lvert\mathcal{N}_k\rvert + 1) + \lvert\mathcal{N}_k\rvert + 2$
Divisões	$\lvert\mathcal{N}_k\rvert$	$\lvert\mathcal{N}_k\rvert + \bar{t}_k(n) - 1$
Comparações	0	2

Fonte: Autor.

Nesta seção, analisa-se quais as condições que devem ser satisfeitas para garantir que o custo computacional do AS-dNLMS seja menor do que o do dNLMS. A análise apresentada aqui foca no número de multiplicações ($\bigotimes$). Resultados análogos podem ser obtidos para o número de somas, mas como eles são menos restritivos para o AS-dNLMS, eles não serão apresentados aqui.

Primeiramente, subtrai-se o número de multiplicações na segunda coluna da Tabela 3 da primeira, obtendo-se

$$\Delta\otimes_k = 2M - 2(M+1)\bar{t}_k(n) - |\mathcal{N}_k|, \tag{4.32}$$

em que $\Delta\otimes_k$ representa a diferença no número de multiplicações entre os algoritmos dNLMS e AS-dNLMS.

Somando-se os $\Delta\otimes_k$ para todo $k \in \{1, \cdots, V\}$ e tomando-se a esperança, obtém-se para a rede como um todo

$$\mathrm{E}\{\Delta\otimes\} = 2VM - \sum_{k=1}^{V}\left[2(M+1)\mathrm{E}\{\bar{t}_k(n)\} + |\mathcal{N}_k|\right], \tag{4.33}$$

sendo $\Delta\otimes \triangleq \sum_{k=1}^{V}\Delta\otimes_k$.

O algoritmo AS-dNLMS é vantajoso em termos de custo computacional quando $\mathrm{E}\{\Delta\otimes\} > 0$, ou seja, quando o algoritmo original requer mais operações. Considerando novamente que, em regime permanente, $\bar{t}_k(n)$ possa ser encarado como uma variável aleatória de Bernoulli, tem-se que $\mathrm{E}\{\bar{t}_k(n)\} = p_{t_k}$. Neste caso, o cenário do pior caso ocorre quando se considera $\mathrm{E}\{\bar{t}_k(n)\} = p_{t_{\max}}$, pois nessa situação $\mathrm{E}\{\Delta\otimes\}$ é minimizado. Fazendo-se essa substituição, obtém-se

$$\mathrm{E}\{\Delta\otimes_{\min}\} = 2VM - \left[2(M+1)Vp_{t_{max}} + \sum_{k=1}^{V}|\mathcal{N}_k|\right]. \tag{4.34}$$

Fazendo-se $\mathrm{E}\{\Delta\otimes\} > 0$, conclui-se que, para que o AS-dNLMS exija menos multiplicações do que o dNLMS, deve-se ter

$$p_{t_{\max}} < \frac{2VM - \sum_{k=1}^{V}|\mathcal{N}_k|}{2V(M+1)}. \tag{4.35}$$

Substituindo-se $p_{t_{\max}}$ pelo $\widehat{p}_{t_{\max}}$ de (4.23) em (4.35), finalmente se obtém

$$\boxed{\beta > \frac{2V(M+1)}{2VM - \sum_{k=1}^{V}|\mathcal{N}_k|}\cdot\sigma^2_{\max}.} \tag{4.36}$$

É importante observar que (4.36) é uma condição **suficiente mas não necessária** para garantir que o AS-dNLMS apresente um custo computacional menor que o dNLMS. Cabe ainda notar que o fator que multiplica $\sigma^2_{\max}$ em (4.36) é sempre maior que um, o que é esperado. Além disso, o lado direito de (4.36) diminui e se aproxima de $\sigma^2_{\max}$ à medida que M cresce. Assim, quanto maior a ordem do filtro, maior a redução de custo computacional do AS-dNLMS em comparação com o dNLMS para um β fixo. Além disso, só se pode garantir uma redução no

custo computacional se

$$\boxed{M > \frac{\sum_{k=1}^{V} |\mathcal{N}_k|}{2V}.} \tag{4.37}$$

Se a Condição (4.37) não for satisfeita, não há um valor finito para $\beta > 0$ que possa satisfazer à condição suficiente (4.36), pois isso implicaria $p_{t_{\max}} \leqslant 0$ em (4.35). Cabe ainda observar que uma expressão diferente para β seria obtida caso se considerassem outros algoritmos difusos (CATTIVELLI; LOPES; SAYED, 2008a; LI; CHAMBERS, 2009) e outras regras para a seleção dos pesos de combinação (SAYED, 2014; FERNANDEZ-BES et al., 2017; YU; SAYED, 2013).

A análise realizada anteriormente pode ser estendida ao algoritmo dNLMS baseado em grafos (NASSIF et al., 2018). Diferentemente da versão voltada a redes de difusão adaptativas "clássicas", em que o vetor regressor $\mathbf{x}_k(n)$ é atualizado simplesmente deslocando os seus elementos a cada iteração, essa versão do algoritmo dNLMS requer que cada nó k amostrado calcule o seu vetor $\mathbf{x}_k(n)$ por meio da Equação (2.28) (página 23). Considerando que esse cálculo seja feito localmente com as informações transmitidas pelos nós vizinhos, obtêm-se as expressões para o custo computacional mostradas na Tabela 4, em que $\bar{\mathcal{N}}_k^m$ denota o conjunto de nós que podem ser acessados pelo nó k em m "passos" através das arestas da rede. Por exemplo, os conjuntos $\bar{\mathcal{N}}_k^0$ e $\bar{\mathcal{N}}_k^1$ consistem, respectivamente, no próprio nó k e em sua vizinhança aberta, tal como definido na Seção 2.3.1 (página 17).

Tabela 4: Custo computacional dos algoritmos dNLMS e ASdNLMS baseados em grafo com pesos ACW: número de operações por nó k.

Operação	dNLMS	ASdNLMS
Mult. ($\otimes$)	$M(3+\lvert\mathcal{N}_k\rvert)+4+\sum_{m=0}^{M-1}\lvert\bar{\mathcal{N}}_k^m\rvert$	$\bar{t}_k(n)\left(2M+2+\sum_{m=0}^{M-1}\lvert\bar{\mathcal{N}}_k^m\rvert\right)+M(1+\lvert\mathcal{N}_k\rvert)+\lvert\mathcal{N}_k\rvert+4$
Somas ($\oplus$)	$M(2+\lvert\mathcal{N}_k\rvert)+3+\sum_{m=0}^{M-1}\lvert\bar{\mathcal{N}}_k^m\rvert$	$\bar{t}_k(n)\left(M+2+\sum_{m=0}^{M-1}\lvert\bar{\mathcal{N}}_k^m\rvert\right)+M(\lvert\mathcal{N}_k\rvert+1)+\lvert\mathcal{N}_k\rvert+2$
Divisões	$\lvert\mathcal{N}_k\rvert$	$\lvert\mathcal{N}_k\rvert+\bar{t}_k(n)-1$
Comparações	0	2

Fonte: Autor.

Repetindo-se o procedimento descrito pelas Equações (4.32) a (4.35), conclui-se que a condição

$$\boxed{\beta > \frac{V(2M+2)+\sum_{k=1}^{V}\left(\sum_{m=0}^{M-1}|\bar{\mathcal{N}}_k^m|\right)}{2VM+\sum_{k=1}^{V}\left(-|\mathcal{N}_k|+\sum_{m=0}^{M-1}|\bar{\mathcal{N}}_k^m|\right)}\cdot\sigma_{\max}^2} \tag{4.38}$$

é **suficiente mas não necessária** para garantir que o AS-dNLMS baseado em grafos apre-

sente um custo computacional menor que o dNLMS correspondente. Assim como na Expressão (4.38), o termo que multiplica $\sigma^2_{\max}$ é sempre maior que um. Entretanto, neste caso é mais difícil estabelecer se a economia em termos de custo computacional cresce ou diminui com o aumento de M. Isso se deve ao termo $\sum_{k=1}^{V}\left(\sum_{m=0}^{M-1}|\bar{\mathcal{N}}_k^m|\right)$, que depende tanto da ordem do filtro como da topologia da rede. É possível que, dependendo do número de conexões no grafo, comportamentos distintos sejam observados com o aumento no valor de M.

Analogamente ao caso anterior, só se pode garantir uma redução no custo computacional se

$$M > \frac{\sum_{k=1}^{V}\left(|\mathcal{N}_k| - \sum_{m=0}^{M-1}|\bar{\mathcal{N}}_k^m|\right)}{2V}. \tag{4.39}$$

Se (4.39) não for satisfeita, não existe $\beta > 0$ finito que possa satisfazer a condição suficiente (4.38). Como $\mathcal{N}_k = \bar{\mathcal{N}}_k^0 \cup \bar{\mathcal{N}}_k^1$, o lado direito de (4.39) se anula para $M = 2$ e se torna negativo para $M > 2$. Consequentemente, é possível obter um valor de β que garanta uma redução no custo computacional para

$$\boxed{M \geqslant 2.} \tag{4.40}$$

É interessante ressaltar que essa condição não depende de qualquer característica da rede, ao contrário da Expressão (4.37).

Para a rede da Figura 13a (página 33), a Expressão (4.37) indica que se deve ter $M > 5{,}4$ para as redes de difusão adaptativas "clássicas". Como nas simulações do Capítulo 3 utilizou-se $M \geqslant 10$, as Condições (4.37) e (4.40) foram satisfeitas. Nesse cenário, a aplicação da Expressão (4.36) para $M = 50$ fornece aproximadamente $\beta > 1{,}1\sigma^2_{\max}$ para o caso clássico, ao passo que a Expressão (4.38) fornece $\beta > 1{,}02\sigma^2_{\max}$ para $M = 10$ no caso baseado em grafos.

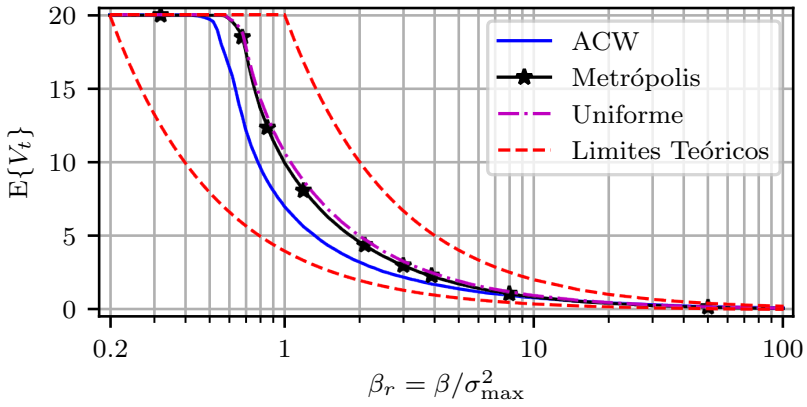
4.5 Validação da análise

Nesta seção, são apresentados resultados de simulação para validar a análise apresentada nas Seções 4.1 a 4.4. Nas simulações apresentadas, considera-se o mesmo cenário descrito na Seção 3.3 (página 32).

A fim de validar a Expressão (4.25), testou-se o algoritmo AS-dNLMS em um ambiente estacionário com diferentes valores de $\beta \geqslant \sigma^2_{\min}$ e três métodos para a seleção dos pesos de combinação: as regras Uniforme e Metrópolis (SAYED, 2014) e o algoritmo ACW (TU; SAYED, 2011). Dois cenários foram considerados: um com a potência do ruído na rede distribuída como

na Figura 13b e outro em que $\sigma^2_{\eta_k} = 0{,}4$ para $k \in \{1, \cdots, V\}$. Os resultados são mostrados nas Figuras 20a e 20b, respectivamente. Para facilitar a visualização, eles são apresentados em termos de β_r. Junto com os dados experimentais, os limites inferior e superior teóricos $\widehat{V}_{t_{\max}}$ e $\widehat{V}_{t_{\min}}$ são apresentados para cada β_r usando linhas tracejadas. Cabe notar que esses limites coincidem na Figura 20b, já que $\sigma^2_{\min} = \sigma^2_{\max}$ nesse caso. Além disso, na Figura 20a, o limite superior permanece fixo em $V_{t_{\max}} = V = 20$ para $\beta \leqslant \sigma^2_{\max}$. Também se observa na Figura 20 que quanto maior o β, menos nós são amostrados nos dois cenários, como esperado. Além disso, os dados experimentais permanecem entre os limites teóricos para todas as regras de combinação e para todos os valores de β_r na Figura 20a. Por outro lado, na 20b, nota-se que o modelo teórico superestima ligeiramente o número de nós amostrados para $1 < \beta_r \leqslant 20$. Nos dois casos, a adoção do algoritmo ACW levou a um número menor de nós amostrados em comparação com as regras Uniforme e Metrópolis.

(a) $\sigma^2_{\eta_k}$ como na Figura 20(b)



(b) $\sigma^2_{\eta_k} = 0{,}4,\ k = 1, \cdots, V$

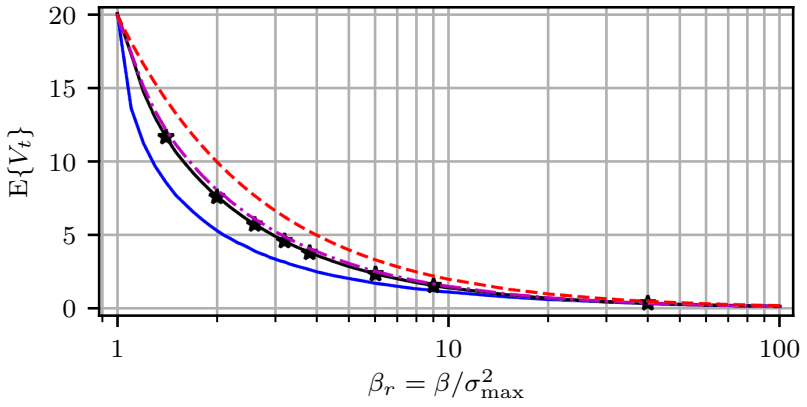


Figura 20: Limites teóricos e número médio de nós amostrados pelo AS-dNLMS com três regras de combinação em função de $\beta \geqslant \sigma^2_{\min}$. (a) $\sigma^2_{\eta_k}$ como na Figura 13. (b) $\sigma^2_{\eta_k} = 0{,}4$ para $k = 1, \cdots, V$.
Fonte: Autor.

Na Figura 21, a Expressão (4.31) é empregada para determinar o passo de adaptação μ_t

para diferentes valores de β com $\Delta n = 3000$. Na Figura 21a são apresentadas as curvas de MSD, na Figura 21b o número de nós amostrados por iteração e na Figura 21c o MSE. Observa-se nas Figuras 21b e 21 (c) que, antes da mudança abrupta no sistema ótimo, o número de nós amostrados estabiliza aproximadamente ao mesmo tempo para $\beta_r > 1{,}1$. Para $\beta_r = 1{,}1$, nota-se que a Expressão (4.31) superestima levemente o valor de μ_t. Isso é esperado para $\beta_r \gtrapprox 1$, conforme discutido na Seção 4.3. Nesse caso, o AS-dNLMS deixou de amostrar os nós antes de atingir o estado estacionário em termos do MSD, o que comprometeu a taxa de convergência. Isso ilustra a importância de uma escolha adequada para μ_t e β. No entanto, como a amostragem dos nós cessou em menos de Δn iterações após o início do estado estacionário em termos do MSE, os resultados obtidos corroboram a validade de (4.31). No entanto, isso mostra que se deve ter uma certa cautela ao usar (4.31) para $\beta \gtrapprox \sigma^2_{\max}$.

Na Figura 22, repetiram-se os experimentos da Figura 21 com valores mais altos de β_r. Nota-se que, antes da mudança abrupta no sistema, o número de nós amostrados estabiliza quase simultaneamente para todos os valores de β_r e o desempenho do AS-dNLMS é mantido. No entanto, após a alteração, o MSD é significativamente afetado nos casos em que $\beta_r \geqslant 8$. Quanto maior o valor de β, mais intensa a deterioração no desempenho. A diferença no comportamento do algoritmo antes e depois da alteração no sistema ótimo pode ser explicada pela inicialização com $\alpha_k(0) = \alpha^+$ para $k \in \{1, \cdots, V\}$. Em contrapartida, logo antes da mudança abrupta, tem-se $\alpha_k(n) \ll \alpha^+$. Assim, o algoritmo deixa de amostrar os nós mais cedo neste caso, como pode ser visto na Fig. 22b. Diante disso, conjectura-se que $\beta \leqslant 5\sigma^2_{\max}$ parece ser um intervalo seguro para a escolha de β.

4.6 Conclusões

Neste capítulo, foram obtidas expressões analíticas que ajudam a entender o papel dos parâmetros β e μ_t nos algoritmos propostos, bem como seus efeitos em termos de desempenho, redução de custos computacionais e economia de energia.

Na Seção 4.1, foram obtidas condições para possibilitar uma redução no número de nós amostrados em função de β. Na Seção 4.2, estudou-se o efeito do parâmetro β sobre o número de nós amostrados em regime permanente. As expressões analíticas obtidas estão relacionadas essencialmente à variância do ruído na rede e indicam que o aumento de β acarreta uma redução na quantidade de nós amostrados, como esperado. Tais resultados foram corroborados

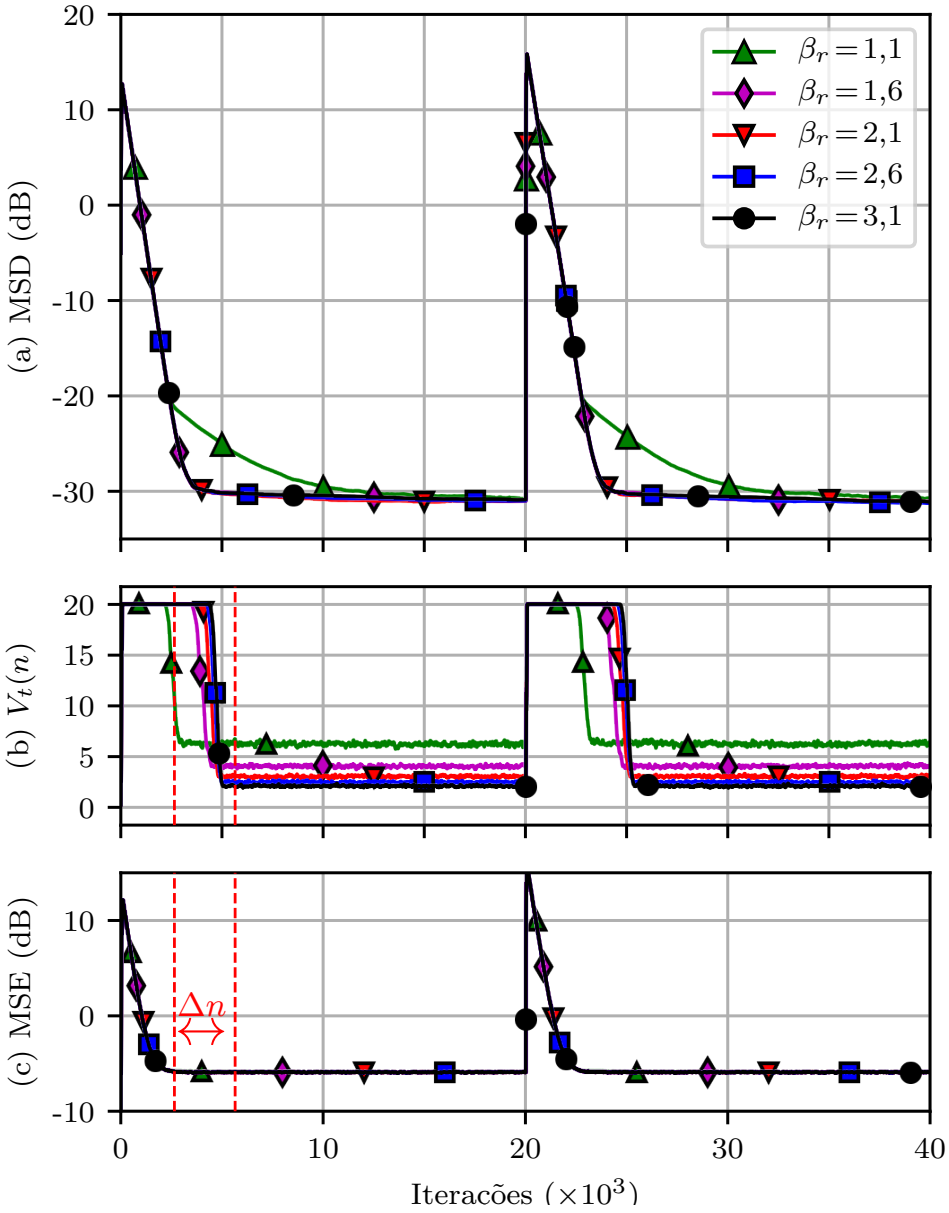


Figura 21: Resultados de simulação obtidos com $1{,}1\sigma^2_{\max} \leqslant \beta \leqslant 3{,}1\sigma^2_{\max}$ e μ_t ajustado por (4.31) para cada caso. (a) Curvas de MSD, (b) Número de nós amostrados por iteração, e (c) Curvas de MSE.
Fonte: Autor.

pelas simulações da Figura 20, embora se tenha constatado que a expressão obtida superestima o número de nós amostrados quando todos os nós da rede estão sujeitos à mesma variância de ruído. Os resultados de simulação indicam ainda que o número de nós amostrados é influenciado pela regra de seleção dos pesos de combinação. Em particular, os resultados obtidos com o algoritmo ACW indicam uma quantidade menor de nós amostrados do que aqueles obtidos com as regras Uniforme e Metrópolis.

Em relação ao passo de adaptação μ_t, estudou-se na Seção 4.3 como escolhê-lo de modo a controlar o número Δn de iterações entre o início do regime permanente em termos do MSE e

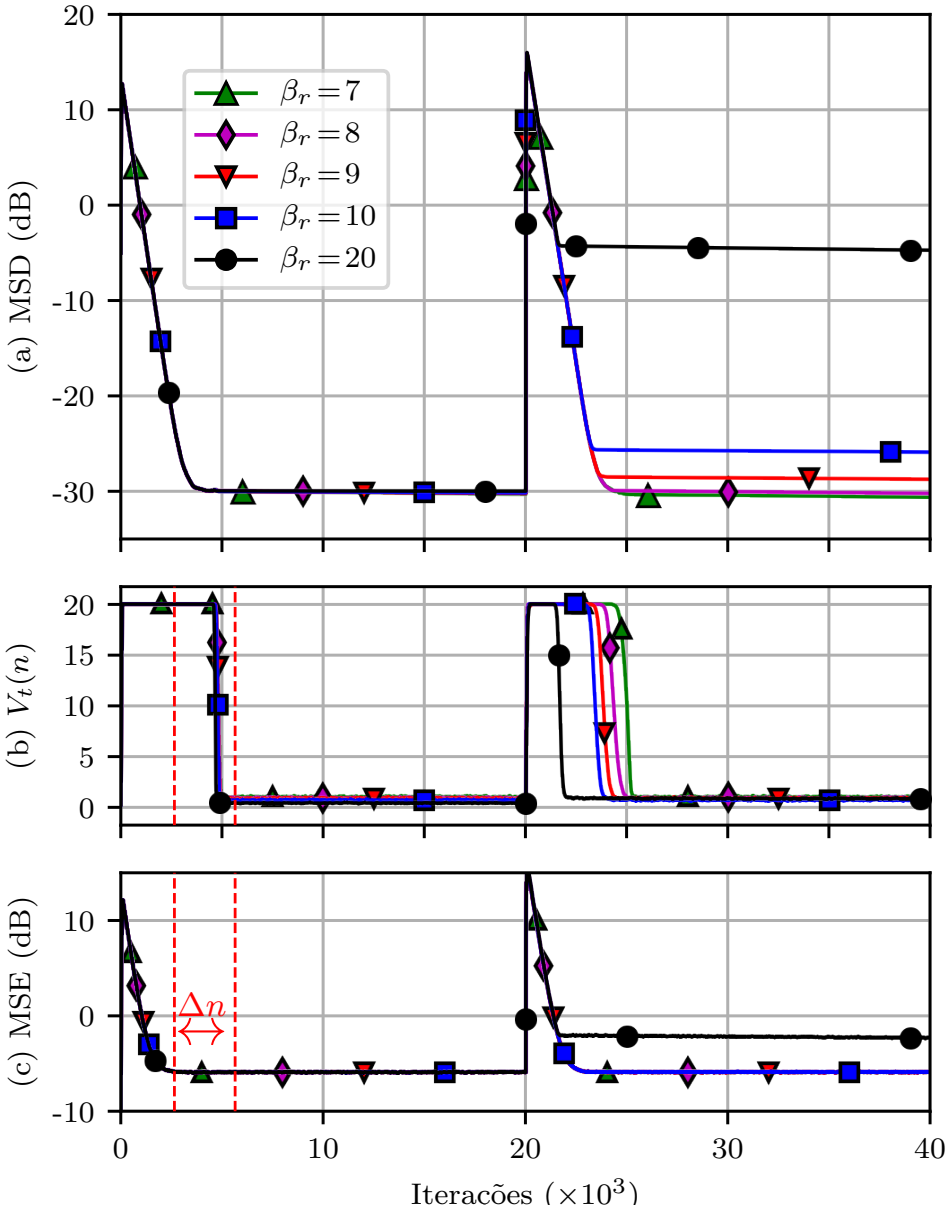


Figura 22: Resultados de Simulação obtidos com $7\sigma^2_{\max} \leqslant \beta \leqslant 20\sigma^2_{\max}$ e μ_t ajustado por (4.31) para cada caso. (a) Curvas de MSD, (b) Número de nós amostrados por iteração, e (c) Curvas de MSE.
Fonte: Autor.

a interrupção da amostragem dos nós. Com isso, obteve-se uma expressão para o ajuste desse parâmetro, que foi validada pelos resultados mostrados nas Figuras 21 e 22. Cabe ressaltar, entretanto, que a expressão obtida deve ser utilizada com cuidado para valores $\beta \gtrapprox \sigma^2_{\max}$, pois ela tende a superestimar o valor de μ_t necessário para obter um certo Δn nessas situações. Esses resultados teóricos permitem uma escolha mais embasada para os valores desses parâmetros e foram validados pelos resultados da simulação. É interessante observar também que os resultados das Figuras 21 e 22 ilustram o possível impacto de um valor inapropriadamente elevado para μ_t na taxa de convergência do algoritmo.

Finalmente, foram estabelecidas na Seção 4.4 condições para garantir que os algoritmos AS-dNLMS e ASC-dNLMS sejam vantajosos em relação ao dNLMS original em termos de custo computacional em função de β, da ordem M do filtro e do número de conexões na rede. Verifica-se que valores elevados de M tendem a favorecer os algoritmos propostos, já que se economizam mais operações por iteração. No entanto, quanto mais conectada for a rede, menor tende a ser a vantagem dos algoritmos apresentados. Isso se deve ao fato de que eles baseiam as suas ações em cada nó k em informações coletadas em toda a vizinhança, o que tende a prejudicá-los em termos de custo computacional quando o número médio de vizinhos por nó aumenta. Por fim, obteve-se uma condição para garantir que os algoritmos AS-dNLMS e ASC-dNLMS possam ser vantajosos em relação ao dNLMS original para algum valor de β. É importante ressaltar que essa condição é respeitada pelos cenários de simulação considerados neste capítulo e no Capítulo 3.

5 CONCLUSÕES

As redes de difusão adaptativas se consolidaram na literatura como soluções interessantes para o processamento distribuído de sinais devido às suas vantagens em relação a outros esquemas de difusão (SAYED, 2014; FERNANDEZ-BES et al., 2017). Em comparação com soluções centralizadas, essas técnicas se destacam por sua melhor escalabilidade e flexibilidade (SAYED, 2014; LOPES; SAYED, 2008b; CATTIVELLI; SAYED, 2009; CATTIVELLI; LOPES; SAYED, 2008a; LI; CHAMBERS, 2009). Além disso, nos últimos anos, foram propostos na literatura algoritmos adaptativos baseados em grafos com processamento distribuído (LUO et al., 2017; LI et al., 2018; LORENZO; BANELLI; BARBAROSSA, 2017; BUI; RAVI; RAMAVAJJALA, 2017; LORENZO et al., 2016; LORENZO et al., 2018; NIGRIS et al., 2017). Tais soluções já despontam como ferramentas interessantes em uma série de aplicações em que a topologia da rede desempenha um papel importante na dinâmica dos sinais envolvidos, como meteorologia, *smart grids*, a internet das coisas, entre outras (NASSIF et al., 2018; HUA et al., 2018; LORENZO et al., 2017a; SANDRYHAILA; MOURA, 2013b; SHUMAN et al., 2013b; CHEN et al., 2015; ANIS; GADDE; ORTEGA, 2016; TSITSVERO; BARBAROSSA; LORENZO, 2016; LORENZO et al., 2018; SPELTA; MARTINS, 2018; LEWENFUS et al., 2019).

Frequentemente, o custo computacional e/ou o consumo energético das redes adaptativas podem se tornar restrições à sua viabilidade. Por isso, mecanismos de amostragem e censura vêm sendo propostos pela comunidade de processamento de sinais. Idealmente, tais técnicas devem reduzir ao máximo a quantidade de informação coletada e/ou transmitida pela rede sem impactar o seu desempenho. Entretanto, as abordagens apresentadas na literatura afetam perceptivelmente o comportamento das soluções para as quais foram propostas, causando impactos na taxa de convergência e/ou no desempenho em regime permanente (ARROYO-VALLES; MALEKI; LEUS, 2013; GHAREHSHIRAN; KRISHNAMURTHY; YIN, 2013; FERNANDEZ-BES et al., 2015; LORENZO et al., 2018; LORENZO et al., 2017a).

Neste trabalho, propuseram-se mecanismos adaptativos de amostragem e censura para soluções adaptativas com processamento distribuído. Os algoritmos propostos utilizam mais nós quando a magnitude do erro na rede é elevada e menos nós caso contrário. Dessa forma, reduz-se o número de nós amostrados e/ou de transmissões apenas quando o impacto dessa decisão

sobre o desempenho da rede é pequeno. Além disso, foi apresentada uma análise teórica dos mecanismos propostos, permitindo escolhas apropriadas para os seus parâmetros.

As contribuições desta dissertação estão concentradas nos Capítulos 3 e 4, por isso, na Tabela 5, são listados os principais tópicos abordados nesses capítulos e as conclusões são apresentadas a seguir.

Tabela 5: Tópicos abordados na tese

Capítulo	**Tópicos principais**
3	- Algoritmos AS-dNLMS e ASC-dNLMS
	- Comparação com outras técnicas de amostragem e censura
	- Efeitos da amostragem e censura adaptativas em cenários estacionários e não-estacionários
4	- Escolha do parâmetro β de modo a garantir uma redução no número de nós amostrados
	- Efeitos do passo μ_t sobre a taxa de adaptação do mecanismo de amostragem
	- Efeitos do parâmetro β sobre o número de nós amostrados em regime permanente e sobre o custo computacional
	- Condições para garantir que os algoritmos AS-dNLMS e ASC-dNLMS sejam vantajosos em termos de custo computacional em comparação com o dNLMS original

Fonte: Autor.

Os Algoritmos AS-dNLMS e ASC-dNLMS

Os resultados de simulação mostram que, com uma boa escolha dos parâmetros β e μ_t, o algoritmo AS-dNLMS é capaz de manter o desempenho do dNLMS original. Embora o seu custo computacional seja ligeiramente maior no transitório devido à manutenção da amostragem de todos os nós, isso é compensando por uma redução significativa do custo computacional em regime permanente. Além disso, verificou-se que o mecanismo de amostragem proposto pode ser empregado tanto em redes de difusão adaptativas como na filtragem adaptativa baseada em grafos.

Além disso, as simulações mostraram que o algoritmo ASC-dNLMS é capaz de economizar mais energia do que outras técnicas de censura encontradas na literatura enquanto obtém um desempenho semelhante em regime permanente. Assim como o algoritmo AS-dNLMS, ele mantém a taxa de convergência do dNLMS original.

Observou-se ainda que, em cenários com variações rápidas no ambiente, o desempenho dos algoritmos AS-dNLMS e ASC-dNLMS se deteriora perceptivelmente em comparação com o dNLMS ou com as outras técnicas de censura testadas. Em sua forma atual, essa é a principal limitação dos algoritmos propostos. Apesar disso, os resultados obtidos em cenários com variações lentas são encorajadores. Em tais situações, os algoritmos AS-dNLMS e ASC-dNLMS preservam o bom comportamento do dNLMS com todos os nós amostrados e superam as demais técnicas, o que motiva a sua utilização e a realização de trabalhos futuros acerca desses métodos.

Análise Teórica das Técnicas Propostas

Inicialmente, foram obtidas condições necessárias ou suficientes para garantir uma redução no número de nós amostrados. Mostrou-se que tal diminuição depende essencialmente do parâmetro β e que a sua escolha está intimamente relacionada à variância do ruído na rede.

Em relação ao passo de adaptação μ_t, estudou-se como escolhê-lo de modo a regular a velocidade do mecanismo de amostragem e, consequentemente, o momento em que os nós deixam de ser amostrados. Os resultados de simulação validaram as expressões analíticas obtidas e mostraram que, mesmo com uma escolha adequada para β, o uso de um valor inapropriado para μ_t pode afetar a taxa de convergência dos algoritmos propostos.

Analisou-se ainda o efeito do parâmetro β sobre o número esperado de nós amostrados em regime permanente. A expressão analítica obtida indica que essas grandezas são inversamente proporcionais entre si. Trata-se de um resultado relevante, pois tem implicações diretas sobre o custo computacional, o que é de interesse do projetista do filtro. Resultados de simulação obtidos em ambientes estacionários validaram a expressão obtida, embora tenham mostrado que ela superestima o número de nós amostrados quando todos os nós da rede estão sujeitos à mesma variância de ruído. Além disso, mostrou-se que o uso de valores elevados de β pode comprometer a rastreabilidade dos algoritmos AS-dNLMS e ASC-dNLMS. Diante dos resultados de simulação, conjectura-se que a adoção de $\beta \leqslant 5\sigma^2_{\max}$ mantenha o bom desempenho dos algoritmos.

Por fim, estudou-se como garantir que o algoritmo AS-dNLMS seja vantajoso em termos de custo computacional em relação ao dNLMS original. Obteve-se assim uma condição que estabelece um valor mínimo a ser adotado para β em função da ordem do filtro e do número de conexões na rede. Com base nisso, também se estabeleceu uma condição para garantir que

os algoritmos AS-dNLMS e ASC-dNLMS possam ser vantajosos em relação ao dNLMS com todos os nós amostrados para algum valor de β. De maneira geral, quanto maior a ordem M do filtro em comparação com o número total de conexões na rede, mais vantajosos os algoritmos propostos tendem a ser em relação ao dNLMS original. Isso se deve ao fato de que os mecanismos propostos utilizam em cada nó k as informações coletadas por toda a vizinhança $\mathcal{N}_k$ para decidir entre a sua amostragem ou não amostragem. Em contrapartida, o número de operações que deixam de ser realizadas quando os nós não são amostrados é influenciado principalmente pela ordem do filtro. Cabe ressaltar que a condição obtida foi respeitada em todas as simulações realizadas nos Capítulos 3 e 4, o que pode indicar que ela não é excessivamente restritiva.

Trabalhos futuros

Diante das conclusões apresentadas, despontam algumas possibilidades de trabalhos futuros. Dentre as principais, destacam-se:

TF1 **Melhorar a capacidade de rastreamento dos algoritmos AS-dNLMS e ASC-dNLMS**. Como visto na Seção 3.3.3, o desempenho dos algoritmos AS-dNLMS e ASC-dNLMS se deteriora perceptivelmente em comparação com as outras técnicas em cenários do tipo *random-walk* com variações rápidas. Além disso, como foi visto na Seção 4.5, a adoção de $\beta > 5\sigma^2_{\max}$ também pode afetar drasticamente a capacidade dos algoritmos de acompanhar alterações no sistema ótimo. Por isso, seria interessante em trabalhos futuros investigar que modificações poderiam ser feitas nos algoritmos para melhorar a sua rastreabilidade e mitigar o impacto de alterações no ambiente sobre o desempenho.

TF2 **Utilização dos mecanismos de amostragem e censura com outras soluções distribuídas**.

Embora neste trabalho tenha se considerado por simplicidade apenas o algoritmo dNLMS, os mecanismos propostos também podem ser usados em outras soluções distribuídas, como o RLS difuso (CATTIVELLI; LOPES; SAYED, 2008a) ou o APA difuso (LI; CHAMBERS, 2009). Seria interessante verificar se o comportamento dos mecanismos se mantém em relação ao que foi estudado neste trabalho ou se há alguma diferença.

TF3 **Utilização dos algoritmos AS-dNLMS e ASC-dNLMS em conjunto com um estimador da variância do ruído**.

Como visto no Capítulo 4, a escolha do parâmetro β está intimamente relacionada à máxima variância $\sigma^2_{\max}$ do ruído na rede, assim como os seus efeitos sobre os mecanismos propostos. Caso $\sigma^2_{\max}$ seja desconhecida, as regras obtidas para a seleção de β perdem a aplicabilidade. Para contornar isso, uma possibilidade seria utilizar os algoritmos AS-dNLMS e ASC-dNLMS em conjunto com algum estimador para a variância do ruído na rede. Considerando que se tivesse, a cada iteração n, uma estimativa $\widehat{\sigma}^2_k(n)$ de $\sigma^2_{\eta_k}$, o projetista do filtro poderia selecionar uma constante γ_b independente do ruído e fazer $\beta_k(n) = \gamma_b \widehat{\sigma}^2_k(n)$ para $k \in \{1, \cdots, V\}$. Seria interessante testar uma configuração desse tipo e avaliar o seu impacto sobre o comportamento dos algoritmos propostos.

TF4 **Utilização do algoritmo ASC-dNLMS em conjunto com outras técnicas de censura**.
Na Seção 3.3, os algoritmos ASC-dNLMS e ACW-S foram comparados entre si, mas nada impede que essas técnicas sejam utilizadas em conjunto. Analisando-se a Figura 15 da página 36, é interessante notar que o algoritmo ASC-dNLMS emprega todos os nós da rede durante o regime permanente, enquanto que o ACW-S utiliza menos nós durante esse período. Em contrapartida, o ASC-dNLMS utiliza comparativamente menos nós em regime permanente. Seria interessante verificar se, com a utilização simultânea das duas técnicas, é possível alcançar um baixo número de transmissões em ambas as etapas sem prejudicar o desempenho.

TF5 **Utilização de outras funções para compor o mecanismo de amostragem**.
Na Seção 3.1, a função degrau foi utilizada na Equação (3.2) para compor o mecanismo de amostragem devido à sua simplicidade. Entretanto, pode ser interessante testar outros limiares para essa função em trabalhos futuros, ou mesmo considerar a adoção de outras funções em seu lugar, como uma função linear por partes, por exemplo. Tais modificações podem alterar de maneira significativa o comportamento do mecanismo de amostragem, e seria interessante verificar se é possível obter com isso um melhor desempenho em determinadas situações, como no caso em que o sistema ótimo varia segundo um modelo do tipo *random-walk*.

REFERÊNCIAS

AKYILDIZ, I. F.; SU, W.; SANKARASUBRAMANIAM, Y.; CAYIRCI, E. A survey on sensor networks. **IEEE Communications Magazine**, IEEE, v. 40, n. 8, p. 102–114, Ago. 2002.

ANIS, A.; GADDE, A.; ORTEGA, A. Efficient sampling set selection for bandlimited graph signals using graph spectral proxies. **IEEE Transactions on Signal Processing**, v. 64, n. 14, p. 3775–3789, Jul. 2016. ISSN 1053-587X.

ARABLOUEI, R.; WERNER, S.; HUANG, Y.-F.; DOĞANÇAY, K. Distributed least mean-square estimation with partial diffusion. **IEEE Transactions on Signal Processing**, IEEE, v. 62, n. 2, p. 472–484, Jan. 2014.

ARENAS-GARCIA, J.; AZPICUETA-RUIZ, L. A.; SILVA, M. T. M.; NASCIMENTO, V. H.; SAYED, A. H. Combinations of adaptive filters: Performance and convergence properties. **IEEE Signal Processing Magazine**, v. 33, n. 1, p. 120–140, Jan. 2016. ISSN 1053-5888.

ARROYO-VALLES, R.; MALEKI, S.; LEUS, G. A censoring strategy for decentralized estimation in energy-constrained adaptive diffusion networks. In: IEEE Workshop on Signal Processing Advances in Wireless Communications (SPAWC), 14. **Proceedings...** 2013. [S.l.], p. 155–159, 2013.

BALDI, P.; POLLASTRI, G. The principled design of large-scale recursive neural network architectures–dag-rnns and the protein structure prediction problem. **Journal of Machine Learning Research**, v. 4, p. 575–602, Set. 2003.

BARANIUK, R. G. Compressive sensing [lecture notes]. **IEEE Signal Processing Magazine**, v. 24, n. 4, p. 118–121, Jul. 2007.

BARESI, L.; HECKEL, R. Tutorial introduction to graph transformation: A software engineering perspective. **Lecture Notes in Computer Science**, Springer-Verlag GmbH, v. 2505, p. 402–429, 2002.

BERBERIDIS, D. K; KEKATOS, V.; WANG, G.; GIANNAKIS, G. B. Adaptive censoring for large-scale regressions. In: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 40. **Proceedings...** 2015. [S.l.], 2015. p. 5475–5479.

BLONDEL, V. D.; HENDRICKX, J. M.; OLSHEVSKY, A.; TSITSIKLIS, J. N. Convergence in multiagent coordination, consensus, and flocking. In: IEEE Conference on Decision and Control European Control Conference (CDC-ECC'05), 44. **Proceedings...** 2005. [s.n.], p. 2996–3000, 2005

BONDY, J. A.; MURTY, U. S. R. **Graph Theory With Applications**. [S.l.]: Macmillan Press Ltd, 1976.

BUA, A.; GORI, M.; SANTINI, F. Recursive neural networks applied to discourse representation theory. In: International Conference on Artificial Neural Networks, 11. **Proceedings...** 2002. [S.l.], p. 290–295, 2002

BUI, T. D.; RAVI, S.; RAMAVAJJALA, V. Neural graph machines: Learning neural networks using graphs. **arXiv preprint arXiv:1703.04818**, 2017.

CATTIVELLI, F. S.; LOPES, C. G.; SAYED, A. H. Diffusion recursive least-squares for distributed estimation over adaptive networks. **IEEE Transactions on Signal Processing**, IEEE, v. 56, n. 5, p. 1865–1877, Maio 2008.

____. Diffusion recursive least-squares for distributed estimation over adaptive networks. **IEEE Transactions on Signal Processing**, v. 56, n. 5, p. 1865–1877, Maio 2008. ISSN 1053-587X.

CATTIVELLI, F. S.; SAYED, A. H. Diffusion LMS strategies for distributed estimation. **IEEE Transactions on Signal Processing**, IEEE, v. 58, n. 3, p. 1035–1048, Mar. 2009.

CHEN, S.; VARMA, R.; SANDRYHAILA, A.; KOVAČEVIĆ, J. Discrete signal processing on graphs: Sampling theory. **IEEE Transactions on Signal Processing**, v. 63, n. 24, p. 6510–6523, Dez. 2015. ISSN 1053-587X.

CHEN, Y.; GU, Y.; HERO, A. O. Sparse LMS for system identification. In: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 34. **Proceedings...** 2009. [S.l.], p. 3125–3128, 2009.

CHOUVARDAS, S.; SLAVAKIS, K.; THEODORIDIS, S. Trading off complexity with communication costs in distributed adaptive learning via krylov subspaces for dimensionality reduction. **IEEE Journal of Selected Topics in Signal Processing**, IEEE, v. 7, n. 2, p. 257–273, Abr. 2013.

COLLBERG, C.; KOBUROV, S.; NAGRA, J.; PITTS, J.; WAMPLER, K. A system for graph-based visualization of the evolution of software. In: ACM symposium on software visualization, 1. **Proceedings...** 2003. [S.l.], p. 77–86, 2003.

CRANDALL, D.; COSLEY, D.; HUTTENLOCHER, D.; KLEINBERG, J.; SURI, S. Feedback effects between similarity and social influence in online communities. In: International Conference on Knowledge Discovery and Data Mining, 14. **Proceedings...** 2008. [S.l.], p. 160–168, 2008.

DINIZ, P. S. R. **Adaptive Filtering: Algorithms and Practical Implementation**. 3rd. ed. [S.l.]: Springer, 2008.

DONOHO, D. L. Compressed sensing. **IEEE Transactions on Information Theory**, IEEE, v. 52, n. 4, p. 1289–1306, Abr. 2006.

FERNANDEZ-BES, J.; ARENAS-GARCÍA, J.; SILVA, M. T. M.; AZPICUETA-RUIZ, L. A. Adaptive diffusion schemes for heterogeneous networks. **IEEE Transactions on Signal Processing**, v. 65, n.21, p. 5661–5674, Nov. 2017.

FERNANDEZ-BES, J.; ARROYO-VALLES, R.; ARENAS-GARCÍA, J.; CID-SUEIRO, J. Censoring diffusion for harvesting wsns. In: IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP), 4. **Proceedings...** 2015. [S.l.], p. 237–240, 2015.

FRANCESCONI, E.; FRASCONI, P.; GORI, M.; MARINAI, S.; SHENG, J. Q.; SPERDUTI, A. Logo recognition by recursive neural networks. In: SPRINGER International Workshop on Graphics Recognition, 2. **Proceedings...** 1997. [S.l.], p. 104–117, 1997.

GHAREHSHIRAN, O. N.; KRISHNAMURTHY, V.; YIN, G. Distributed energy-aware diffusion least mean squares: Game-theoretic learning. **IEEE Journal of Selected Topics in Signal Processing**, IEEE, v. 7, n. 5, p. 821–836, Out. 2013.

HAYKIN, S. **Adaptive Filter Theory**. 5th. ed. [S.l.]: Pearson, Upper Saddle River, 2014.

HUA, F.; NASSIF, R.; RICHARD, C.; WANG, H.; SAYED, A. H. A preconditioned graph diffusion LMS for adaptive graph signal processing. In: European Signal Processing Conference (EUSIPCO), 26. **Proceedings...** 2018. [S.l.: s.n.], p. 111–115, 2018

INOKUCHI, A.; WASHIO, T.; MOTODA, H. An apriori-based algorithm for mining frequent substructures from graph data. In: SPRINGER European conference on principles of data mining and knowledge discovery, 4. **Proceedings...** 2000. [S.l.], p. 13–23, 2000.

ISUFI, E.; BANELLI, P.; LORENZO, P.; LEUS, G. Observing and tracking bandlimited graph processes. **arXiv preprint arXiv:1712.00404**, 2017.

KRAHMER, E.; ERK, S. V.; VERLEG, A. Graph-based generation of referring expressions. **Computational Linguistics**, MIT Press, v. 29, n. 1, p. 53–72, 2003.

LATOUCHE, P.; ROSSI, F. Graphs in machine learning: an introduction. In: European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN), 23. **Proceedings...** 2015. [S.l.: s.n.], p. 207–218, 2015.

LÁZARO-GREDILLA, M.; AZPICUETA-RUIZ, L. A.; FIGUEIRAS-VIDAL, A. R.; ARENAS-GARCÍA, J. Adaptively biasing the weights of adaptive filters. **IEEE Transactions on Signal Processing**, v. 58, n. 7, p. 3890–3895, Jul. 2010.

LEWENFUS, G.; MARTINS, W. A.; CHATZINOTAS, S.; OTTERSTEN, B. On the use of vertex-frequency analysis for anomaly detection in graph signals. In: Simpósio Brasileiro de Telecomunicações e Processamento de Sinais (SBrT), 37. **Anais...** 2019, p. 1–5, 2019.

LI, L.; CHAMBERS, J. A. Distributed adaptive estimation based on the APA algorithm over diffusion networks with changing topology. In: EEE/SP Workshop on Statistical Signal Processing, 15. **Proceedings...** 2009. [S.l.], p. 757–760, 2009.

LI, R.; WANG, S.; ZHU, F.; HUANG, J. Adaptive graph convolutional neural networks. **arXiv preprint arXiv:1801.03226**, 2018.

LOPES, C. G.; SAYED, A. H. Diffusion adaptive networks with changing topologies. In: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 33. **Proceedings...** 2008. [S.l.], p. 3285–3288, 2008.

____. Diffusion least-mean squares over adaptive networks: Formulation and performance analysis. **IEEE Transactions on Signal Processing**, IEEE, v. 56, n. 7, p. 3122–3136, Jul. 2008.

LORENZO, P. D.; BANELLI, P.; BARBAROSSA, S. Optimal sampling strategies for adaptive learning of graph signals. In: European Signal Processing Conference (EUSIPCO), 25. **Proceedings...** 2017. [S.l.], p. 1684–1688, 2017.

LORENZO, P. D.; BANELLI, P.; BARBAROSSA, S.; SARDELLITTI, S. Distributed adaptive learning of graph signals. **IEEE Transactions on Signal Processing**, IEEE, v. 65, n. 16, p. 4193–4208, Ago. 2017.

LORENZO, P. D.; BANELLI, P.; ISUFI, E.; BARBAROSSA, S.; LEUS, G. Adaptive graph signal processing: Algorithms and optimal sampling strategies. **IEEE Transactions on Signal Processing**, v. 66, n. 13, p. 3584–3598, Jul. 2018. ISSN 1053-587X.

LORENZO, P. D.; BARBAROSSA, S.; BANELLI, P.; SARDELLITI, S. Adaptive least mean squares estimation of graph signals. **IEEE Transactions on Signal and Information Processing over Networks**, IEEE, v. 2, n. 4, p. 555–568, Dez. 2016.

LORENZO, P. D.; BARBAROSSA, S.; SAYED, A. H. Bio-inspired decentralized radio access based on swarming mechanisms over adaptive networks. **IEEE Transactions on Signal Processing**, IEEE, v. 61, n. 12, p. 3183–3197, Jun. 2013.

LUO, Z.; LIU, L.; YIN, J.; LI, Y.; WU, Z. Deep learning of graphs with ngram convolutional neural networks. **IEEE Transactions on Knowledge and Data Engineering**, IEEE, v. 29, n. 10, p. 2125–2139, Out. 2017.

MASON, A. E. W.; BLAKE, E. H. A graphical representation of the state spaces of hierarchical level-of-detail scene descriptions. **IEEE Transactions on Visualization and Computer Graphics**, IEEE, v. 7, n. 1, p. 70–75, Jan.–Mar. 2001.

NASCIMENTO, V. H.; SILVA, M. T. M. Adaptive filters. In: CHELLAPA, R.; THEODORIDIS, S. (Ed.). **Academic Press Library in Signal Processing: Signal Processing Theory and Machine Learning**. Chennai: Academic Press, 2014. v. 1, cap. 12, p. 619–761.

NASSIF, R.; RICHARD, C.; CHEN, J.; SAYED, A. H. A graph diffusion LMS strategy for adaptive graph signal processing. In: Asilomar Conference on Signals, Systems and Computers, 50. **Proceedings...** 2017. [S.l.: s.n.], p. 1973-1976, 2017.

____. Distributed diffusion adaptation over graph signals. In: EEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 43. **Proceedings...** 2018. [S.l.: s.n.], p. 4129–4133, 2018.

NIGRIS, S. D.; BAUTISTA, E.; ABRY, P.; AVRACHENKOV, K.; GONCALVES, P. Fractional graph-based semi-supervised learning. In: European Signal Processing Conference (EUSIPCO), 25. **Proceedings...** 2017. [S.l.: s.n.], p. 356–360, 2017.

PAGANI, G. A.; AIELLO, M. The power grid as a complex network: a survey. **Physica A: Statistical Mechanics and its Applications**, Elsevier, v. 392, n. 11, p. 2688–2700, 2013.

RUSSELL, S. J.; NORVIG, P. **Artificial intelligence: a modern approach**. [S.l.]: Pearson Education Limited, 2016.

SANDRYHAILA, A.; MOURA, J. M. F. Discrete signal processing on graphs. **IEEE Transactions on Signal Processing**, v. 61, n. 7, p. 1644–1656, Abr. 2013. ISSN 1053-587X.

SAYED, A. H. **Adaptive Filters**. [S.l.]: John Wiley & Sons, NJ, 2008.

____. **Adaptation, Learning, and Optimization over Networks**. [S.l.]: Foundations and Trends in Machine Learning, now Publishers Inc., Hanover, MA, 2014. v. 7. 311-801 p.

SHUMAN, D. I.; NARANG, S. K.; FROSSARD, P.; ORTEGA, A.; VANDERGHEYNST, P. The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains. **IEEE Signal Processing Magazine**, IEEE, v. 30, n. 3, p. 83–98, Maio 2013.

SPELTA, M. J.; MARTINS, W. A. Normalized LMS algorithm and data-selective strategies for adaptive graph signal estimation. **Signal Processing**, Elsevier, v. 167, Fev. 2020

TAKAHASHI, N.; YAMADA, I. Link probability control for probabilistic diffusion least-mean squares over resource-constrained networks. In: IEEE International Conference on Acoustics, Speech and Signal Processing, 30. **Proceedings...** 2010. [S.l.], p. 3518–3521, 2010.

TAKAHASHI, N.; YAMADA, I.; SAYED, A. H. Diffusion least-mean squares with adaptive combiners: Formulation and performance analysis. **IEEE Transactions on Signal Processing**, v. 58, p. 4795–4810, Set. 2010.

TSITSVERO, M.; BARBAROSSA, S.; LORENZO, P. D. Signals on graphs: Uncertainty principle and sampling. **IEEE Transactions on Signal Processing**, IEEE, v. 64, n. 18, p. 4845–4860, Set. 2016.

TU, S.-Y.; SAYED, A. H. Optimal combination rules for adaptation and learning over networks. In: IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP), 1.**Proceedings...** 2011. [S.l.], p. 317–320, 2011.

XIAO, L.; BOYD, S. Fast linear iterations for distributed averaging. **Systems & Control Letters**, v. 53, no. 1, p. 65–78, Set. 2004.

XU, S.;DE LAMARE, R. C.; POOR, H. V. Adaptive link selection algorithms for distributed estimation. **EURASIP Journal on Advances in Signal Processing**, Springer, v. 2015, n. 1, p. 86, 2015.

YANG, L.; ZHU, H.; KANG, K.; LUO, X.; QIAN, H.; YANG, Y. Distributed censoring with energy constraint in wireless sensor networks. In: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 43. **Proceedings...** 2018. [S.l.], p. 6428–6432, 2018.

YU, C.-K.; SAYED, A. H. A strategy for adjusting combination weights over adaptive networks. In: IEEE International Conference on Acoustics, Speech, and Signal Process (ICASSP), 38. **Proceedings...** 2013.[s.n.], p. 4579–4583, 2013.

ZHAO, X.; SAYED, A. H. Single-link diffusion strategies over adaptive networks. In: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 37. **Proceedings...** 2012. [S.l.], p. 3749–3752, 2012.

APÊNDICE A – PROCESSAMENTO DE SINAIS EM GRAFOS

Este apêndice se dedica à apresentação de conceitos elementares da teoria de processamento de sinais em grafos, bem como de alguns algoritmos adaptativos voltados a esse tipo de aplicação. Na Seção 2.3.1 é feita uma breve revisão da teoria de grafos. Na Seção A.1, são apresentados os fundamentos da teoria de processamento de sinais em grafos. Em particular, dá-se destaque às ideias utilizadas pelos algoritmos apresentados na Seção A.2. Por sua vez, na Seção A.3 é apresentada uma comparação entre esses algoritmos. Na Seção A.4 são apresentadas as conclusões do apêndice.

A.1 Conceitos fundamentais

Uma vez definido o sinal em um grafo, pode-se também estender a definição de transformada de Fourier a sinais desse tipo. Esse conceito é importante por ser utilizado na obtenção do algoritmo LMS aplicado a grafos de (LORENZO et al., 2016). Por esse motivo, será mostrado a seguir como essa extensão pode ser feita.

Primeiramente, convém definir um operador Laplaciano para grafos. Denotando-se esse operador por $\boldsymbol{\mathcal{L}}$, isso é feito considerando-se (SHUMAN et al., 2013b; LORENZO et al., 2016)

$$\boldsymbol{\mathcal{L}} \triangleq \boldsymbol{\Upsilon} - \mathbf{A}, \tag{A.1}$$

em que $\boldsymbol{\Upsilon}$ é uma matriz diagonal cujo elemento (i,i) é igual ao grau do nó i e $\mathbf{A}$ é a matriz de adjacência do grafo em questão. É possível mostrar que o operador $\boldsymbol{\mathcal{L}}$ assim definido satisfaz a todas as condições necessárias para ser considerado um operador Laplaciano (SHUMAN et al., 2013b).

Uma vez definido $\boldsymbol{\mathcal{L}}$, a extensão do conceito de transformada de Fourier a sinais definidos sobre grafos parte de uma analogia com o caso clássico, como mostrado na Tabela 6. Para

compreender essa analogia, cabe notar que no caso de uma função de tempo contínuo $f(t)$, o operador Laplaciano é usualmente denotado por ∇^2 e consiste em

$$\nabla^2 f(t) = \frac{d^2 f(t)}{dt^2}. \tag{A.2}$$

Esse operador tem como autovetores as exponenciais complexas do tipo $e^{j\omega t}$, em que j denota a unidade imaginária, t denota o tempo e $\omega \in \mathbb{R}$ representa uma frequência, pois

$$\nabla^2[e^{j\omega t}] = -\omega^2 e^{j\omega t}. \tag{A.3}$$

A transformada de Fourier de $f(t)$ é então obtida expandindo-se essa função em termos dos autovetores do Laplaciano dado por (A.2). De maneira análoga, considera-se que a transformada de Fourier estendida a grafos (*graph Fourier transform* – GFT) de um sinal $\mathbf{u}$ definido sobre um grafo é dada pela sua expansão em termos dos autovetores do Laplaciano da Equação (A.1) (SHUMAN et al., 2013b). Para grafos não direcionados, a matriz $\boldsymbol{\mathcal{L}}$ é simétrica e positiva semi-definida. Nesse caso, ela está associada a um conjunto de autovetores ortonormais, cada um associado a um autovalor real e não negativo correspondente (LORENZO et al., 2016) e admite uma decomposição de autovetores e autovalores da forma

$$\boldsymbol{\mathcal{L}} = \boldsymbol{\Omega}\boldsymbol{\Lambda}\boldsymbol{\Omega}^{\mathrm{H}}, \tag{A.4}$$

em que as colunas de $\boldsymbol{\Omega}$ representam os autovetores de $\boldsymbol{\mathcal{L}}$, $\boldsymbol{\Lambda}$ é uma matriz diagonal com os seus respectivos autovalores e $(\cdot)^{\mathrm{H}}$ representa o Hermitiano (LORENZO et al., 2016). Cabe notar que, como o grafo é considerado constante no tempo, $\mathbf{A}$, $\boldsymbol{\Upsilon}$, $\boldsymbol{\mathcal{L}}$, $\boldsymbol{\Lambda}$ e $\boldsymbol{\Omega}$ também o são. Dessa forma, denotando-se a GFT de $\mathbf{u}(n)$ por $\mathbf{s}(n)$, pode-se escrever (SHUMAN et al., 2013b; LORENZO et al., 2016)

$$\mathbf{s}(n) = \boldsymbol{\Omega}^{\mathrm{H}}\mathbf{u}(n). \tag{A.5}$$

O principal interesse por trás da utilização da GFT $\mathbf{s}(n)$ consiste no fato de que, se o sinal $\mathbf{u}(n)$ variar pouco entre nós vizinhos, $\mathbf{s}(n)$ tende a ser **esparsa**, ou seja, a apresentar muitos elementos nulos (LORENZO et al., 2016; SPELTA; MARTINS, 2018). Dessa forma, pode-se reduzir a quantidade de dados utilizados nos cálculos e, consequentemente, o custo computacional. Por exemplo, considerando-se o grafo da Figura 7 e o sinal definido pela Equação (2.23)

Tabela 6: Analogia entre o processamento de sinais em grafos e o caso convencional

Conceito	Processamento de Sinais Convencional	Processamento de sinais em grafos
Operador Laplaciano	$\nabla^2 f(t) = -\dfrac{\partial^2 f(t)}{\partial t^2}$	$\boldsymbol{\mathcal{L}} = \boldsymbol{\Upsilon} - \mathbf{A}$
Autovetores do operador Laplaciano	$f(t) = e^{j\omega t}$, pois $\nabla^2[e^{j\omega t}] = -\omega^2 e^{j\omega t}$.	$\boldsymbol{\Omega}$ obtida por meio da decomposição de $\boldsymbol{\mathcal{L}} = \boldsymbol{\Omega\Lambda\Omega}^{\mathsf{H}}$
Transformada de Fourier	obtida expandindo-se $f(t)$ em termos dos autovetores do operador Laplaciano ∇^2	obtida expandindo-se $\mathbf{u}(n)$ em termos dos autovetores do operador Laplaciano $\boldsymbol{\mathcal{L}}$

Fonte: Autor.

na página 20, a aplicação da Equação (A.1) fornece

$$\boldsymbol{\mathcal{L}} = \begin{bmatrix} 3 & -1 & -1 & -1 & 0 & 0 & 0 \\ -1 & 3 & -1 & -1 & 0 & 0 & 0 \\ -1 & -1 & 3 & -1 & 0 & 0 & 0 \\ -1 & -1 & -1 & 4 & -1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 3 & -1 & -1 \\ 0 & 0 & 0 & 0 & -1 & 2 & -1 \\ 0 & 0 & 0 & 0 & -1 & -1 & 2 \end{bmatrix}. \tag{A.6}$$

Decompondo-se a matriz $\boldsymbol{\mathcal{L}}$ em autovalores e autovetores como na Equação (A.4), obtém-se que a GFT de $\mathbf{u}(n)$ é dada por

$$\mathbf{s}(n) \approx [0{,}7559 \quad 3{,}3362 \quad 0 \quad -0{,}0370 \quad 0 \quad 0 \quad 0{,}5448]^{\mathrm{T}}, \tag{A.7}$$

em que aparecem três elementos nulos.

De maneira análoga, também é possível estender o conceito de anti-transformada de Fourier a sinais definidos sobre grafos. Essa operação pode ser descrita por

$$\mathbf{u}(n) = \boldsymbol{\Omega}\mathbf{s}(n). \tag{A.8}$$

Assim, após realizar todos os cálculos utilizando-se $\mathbf{s}(n)$ em vez de $\mathbf{u}(n)$, é possível recuperá-lo utilizando (A.8) (LORENZO et al., 2016; SPELTA; MARTINS, 2018).

Por fim, cabe mencionar que é possível estender ainda mais conceitos clássicos de processamento de sinais a grafos, como convolução e modulação (SHUMAN et al., 2013b). Entretanto, esses tópicos não serão abordados aqui, pois fogem do escopo do presente trabalho.

A.2 Extensões do algoritmo LMS com processamento centralizado

Nesta seção, são expostas algumas das extensões do algoritmo LMS a sinais definidos sobre grafos. Cabe ressaltar que também foram propostas versões baseadas em grafos de outros algoritmos e que há outras extensões do algoritmo LMS voltadas para esse tipo de situação (HUA et al., 2018; NASSIF et al., 2018). Entretanto, como o objetivo deste apêndice não é realizar uma exposição exaustiva de todos os algoritmos baseados em grafos, tais soluções não serão abordadas.

A.2.1 A versão de Lorenzo et al.

Em (LORENZO et al., 2016), desenvolve-se uma extensão do algoritmo LMS a sinais definidos sobre grafos tendo-se em mente a seguinte aplicação: dado um conjunto de amostras ruidosas coletadas sobre os nós de um grafo ao longo do tempo, como estimar o valor do sinal $\mathbf{u}(n)$ ao longo de toda a rede utilizando apenas um número limitado de nós e de modo a mitigar a influência do ruído?

Para obter um algoritmo capaz de atender a esses requisitos, utiliza-se o fato de que, em muitos casos, a GFT de $\mathbf{u}(n)$ pode ser considerada uma representação **esparsa** do sinal (LORENZO et al., 2016). Assim, a ideia consiste em estimar em tempo real a GFT de $\mathbf{u}(n)$ seguindo-se a definição da Equação (A.5), de modo a permitir a reconstrução do sinal com o menor número de amostras possível a cada iteração.

O sinal amostrado nos nós do grafo é dado pelo vetor (LORENZO et al., 2016)

$$\mathbf{d}(n) = \mathbf{T}(n)\left[\mathbf{u}(n) + \boldsymbol{\eta}(n)\right], \tag{A.9}$$

em que $\boldsymbol{\eta}(n)$ é um vetor de ruído branco i.i.d. não correlacionado com os demais sinais e a matriz $\mathbf{T}(n)$ consiste **em um operador de amostragem**. Trata-se de uma matriz diagonal, cujo elemento $\mathbf{T}_{i,i}(n)$ é igual a 1 se o nó i for amostrado no instante n e igual a 0 caso contrário. Com base nos elementos diagonais de $\mathbf{T}(n)$, define-se o conjunto $\mathcal{T}(n)$ como sendo composto pelos nós amostrados no instante de tempo n. É possível mostrar que uma condição necessária para que o algoritmo consiga estimar o sinal $\mathbf{u}(n)$ a partir dos nós contidos em $\mathcal{T}(n)$ é que a cardinalidade desse conjunto seja maior ou igual ao número de elementos não nulos de $\mathbf{s}(n)$ (LORENZO et al., 2016). Cabe observar que tanto a matriz $\mathbf{T}$ como o conjunto $\mathcal{T}$ podem variar ao longo do

tempo.

Utilizando-se a Equação (A.8), é possível reescrever a Equação (A.9) como (LORENZO et al., 2016)

$$\mathbf{d}(n) = \mathbf{T}(n)\mathbf{\Omega}\mathbf{s}(n) + \mathbf{T}(n)\boldsymbol{\eta}(n). \tag{A.10}$$

A partir da Equação (A.10), é possível formular uma expressão que permite estimar a GFT $\mathbf{s}(n)$ diretamente a partir do sinal medido $\mathbf{d}(n)$. Denotando-se por $\widehat{\mathbf{s}}(n)$ a estimativa de $\mathbf{s}(n)$, a estimativa $\widehat{\mathbf{u}}(n)$ do sinal pode então ser obtida aplicando-se a GFT inversa a $\widehat{\mathbf{s}}(n)$, por meio da expressão $\widehat{\mathbf{u}}(n) = \mathbf{\Omega}\widehat{\mathbf{s}}(n)$ (LORENZO et al., 2016). Essa abordagem é vantajosa, pois permite selecionar a matriz $\mathbf{T}(n)$ em cada iteração com base no número de elementos não nulos de $\widehat{\mathbf{s}}(n)$ segundo alguma estratégia de amostragem pré-determinada (LORENZO et al., 2016).

A partir da Equação (A.10), pode-se escrever o problema como (LORENZO et al., 2016)

$$\min_{\widehat{\mathbf{s}}(n),\ \mathbf{T}(n)\in\mathcal{T}(n)} \mathrm{E}\{\|\mathbf{d}(n) - \mathbf{T}(n)\mathbf{\Omega}\widehat{\mathbf{s}}(n)\|^2\} + \kappa f(\widehat{\mathbf{s}}(n)), \tag{A.11}$$

em que $\mathcal{T}(n)$ é o conjunto que restringe a escolha da matriz $\mathbf{T}(n)$, $f_s(\cdot)$ é uma função introduzida a fim de forçar a esparsidade de $\widehat{\mathbf{s}}(n)$, κ é um parâmetro utilizado para regular o quão esparso se deseja que $\widehat{\mathbf{s}}(n)$ seja.

O algoritmo LMS estendido a grafos é então obtido buscando-se uma solução recursiva para o Problema (A.11). Pode-se mostrar que a solução de (A.11) é dada por (LORENZO et al., 2016)

$$\boxed{\widehat{\mathbf{s}}(n+1) = \mathbf{g}_\gamma\Big(\widehat{\mathbf{s}}(n) + \mu\mathbf{\Omega}^{\mathrm{H}}\mathbf{T}(n)\big[\mathbf{d}(n) - \mathbf{\Omega}\widehat{\mathbf{s}}(n)\big]\Big),} \tag{A.12}$$

em que a função $\mathbf{g}_\gamma(\cdot)$ depende da função $f_s(\cdot)$ escolhida para compor a função custo e μ é um passo de adaptação como o utilizado no LMS clássico.

Uma escolha simples mas eficiente para a função $\mathbf{g}_\gamma(\cdot)$, consiste em aplicar(LORENZO et al., 2016)

$$\mathbf{g}_\gamma(s_i) = \begin{cases} s_i, \text{ se } |s_i| > \gamma \\ 0, \text{ caso contrário} \end{cases}, \tag{A.13}$$

em que γ é um limiar. Em (LORENZO et al., 2016), adota-se $\gamma = \kappa\mu$. Considerando essa função $\mathbf{g}_\gamma(\cdot)$, o algoritmo necessita de $3V^2+V$ multiplicações, $3V^2-2V$ somas e $2V$ comparações a cada iteração para o cálculo da estimativa $\widehat{\mathbf{u}}(n+1)$. Um esquema do funcionamento do algoritmo é

mostrado na Figura 23. Note que novamente blocos constituídos de matrizes têm como saída o produto da matriz pelo vetor de entrada.

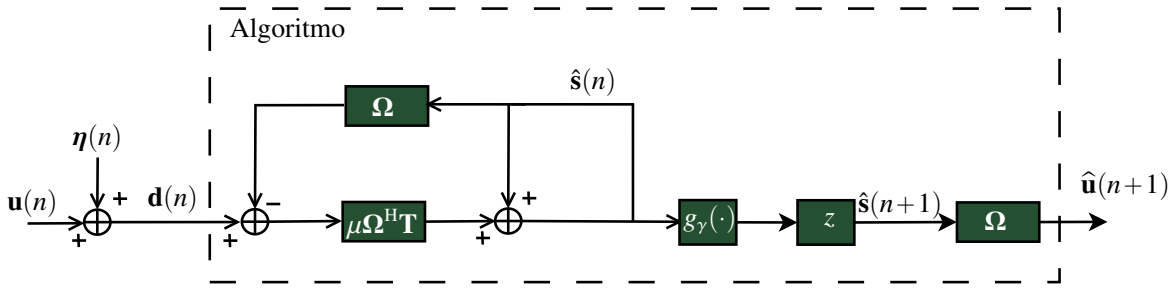


Figura 23: Representação esquemática do funcionamento do algoritmo de (LORENZO et al., 2016).
Fonte: Autor.

Nota-se que há uma série de diferenças entre a Equação (A.12) e a equação de adaptação do algoritmo LMS clássico (HAYKIN, 2014; DINIZ, 2008; NASCIMENTO; SILVA, 2014; SAYED, 2008). Em primeiro lugar, em vez de atualizar um vetor de coeficientes, atualiza-se em (A.12) a estimativa da GFT do sinal $\mathbf{u}(n)$. Além disso, uma segunda diferença consiste na utilização de uma função que força a estimativa da GFT de $\mathbf{u}(n)$ a ser esparsa, o que permite que nem todos os nós sejam utilizados na estimativa do sinal definido sobre o grafo. Por fim, cabe ressaltar que embora se trate de um algoritmo voltado essencialmente à predição de sinais sobre grafos, o termo "predição" adquire neste contexto com um significado ligeiramente diferente do que ocorre em aplicações de filtragem adaptativa "convencional". A predição que o algoritmo procura efetuar não é exatamente temporal, mas sim espacial, uma vez que o objetivo é, a partir de um conjunto limitado de nós amostrados, estimar o sinal em cada nó do grafo.

A.2.2 A versão de Nassif et al.

Em (NASSIF et al., 2017), considera-se que há um vetor desejado $\mathbf{d}(n)$ a ser estimado que está relacionado ao sinal sobre o grafo $\mathbf{u}(n)$ por meio da Equação (2.26). Como o ruído é assumido i.i.d., sua matriz de covariância é diagonal dada por $\mathbf{R}_\eta = \mathrm{E}\{\boldsymbol{\eta}(n)\boldsymbol{\eta}^{\mathrm{T}}(n)\} = \mathrm{diag}\{\sigma^2_{\eta,i}\}_{k=1}^{V}$, em que $\sigma^2_{\eta_i}$ é a variância do ruído no nó i. Definindo-se uma matriz $\widetilde{\mathbf{X}}(n)$ de dimensão $V \times M$ como

$$\widetilde{\mathbf{X}}(n) \triangleq [\mathbf{u}(n), \mathbf{A}\mathbf{u}(n), \cdots, \mathbf{A}^{M-1}\mathbf{u}(n)], \tag{A.14}$$

o vetor de coeficientes ótimos $\mathbf{w}^{\mathrm{o}}$ pode ser encontrado para $M \geqslant L$ resolvendo-se o problema de otimização (NASSIF et al., 2017)

$$\mathbf{w}^{\mathrm{o}} = \arg\min_{\mathbf{w}} \{\mathrm{E}\|\mathbf{d}(n) - \widetilde{\mathbf{X}}(n)\mathbf{w}\|^2\}. \tag{A.15}$$

Embora seja possível calcular analiticamente o valor de $\mathbf{w}^{\mathrm{o}}$ por meio da Equação (A.15), isso depende do conhecimento da matriz de autocorrelação de $\widetilde{\mathbf{X}}(n)$ e da correlação cruzada entre $\widetilde{\mathbf{X}}(n)$ e $\mathbf{d}(n)$, que podem ser desconhecidas ou variar no tempo (NASSIF et al., 2017). Uma alternativa consiste em utilizar o método do gradiente estocástico (HAYKIN, 2014; DINIZ, 2008; NASCIMENTO; SILVA, 2014). Fazendo-se isso, obtém-se (NASSIF et al., 2017)

$$\boxed{\mathbf{w}(n+1) = \mathbf{w}(n) + \mu\widetilde{\mathbf{X}}^{\mathrm{T}}(n)\big[\mathbf{d}(n) - \widetilde{\mathbf{X}}(n)\mathbf{w}(n)\big],} \tag{A.16}$$

em que μ é um **passo de adaptação** análogo ao utilizado no algoritmo LMS clássico.

Em termos de custo computacional, esse algoritmo necessita de $2MV + V + V^2(M-1)$ multiplicações e $V^2(M-1) + MV - M + 2V$ somas a cada iteração para calcular sua estimativa do vetor desejado, dada por $\widetilde{\mathbf{X}}(n)\mathbf{w}(n)$.

Nota-se que há uma analogia entre o vetor $\mathbf{w}(n)$ e a matriz $\widetilde{\mathbf{X}}(n)$ de (A.16) e os vetores de coeficientes e de entrada do algoritmo LMS clássico, respectivamente. Essa analogia se torna ainda mais clara quando se leva em conta o paralelo estabelecido entre a matriz de adjacência $\mathbf{A}$ do grafo e o operador atraso unitário.

A.3 Análise comparativa entre as versões centralizadas

Como foi dito na Seção A.2, em (LORENZO et al., 2016), foi proposta uma versão do algoritmo LMS voltada a aplicações de predição. Em (NASSIF et al., 2017), em contrapartida, foi proposta uma outra versão, voltada principalmente a situações envolvendo identificação de sistemas. O objetivo desta seção consiste em comparar seus comportamentos e as diferentes ideias que embasam seus desenvolvimentos. Para isto, são mostrados resultados de simulação em diferentes situações. Em todas elas, considerou-se um grafo gerado aleatoriamente em cada realização segundo o modelo de Erdös-Renyi (NASSIF et al., 2017). Na Figura 24, é mostrado um exemplo desse grafo com $V = 20$ nós. Além disso, considerou-se que a variância do ruído é diferente em cada nó, como mostrado na Figura 25. São considerados dois tipos de aplicação:

identificação de sistemas e predição, que são abordados respectivamente nas Subseções A.3.1 e A.3.2.

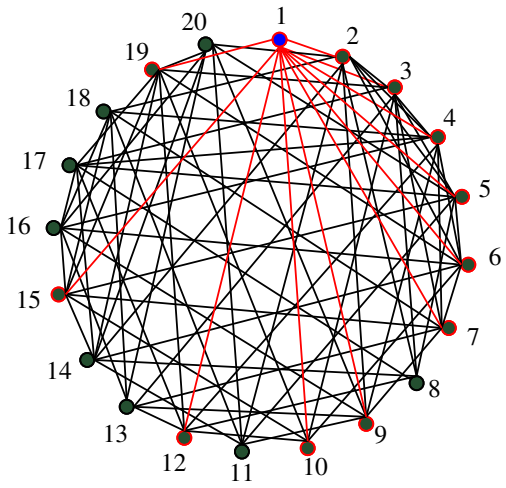


Figura 24: Grafo gerado com $V = 20$ nós segundo o modelo de Erdös-Renyi (NASSIF et al., 2017). Em destaque, o nó 1 e suas conexões.

Fonte: Autor.

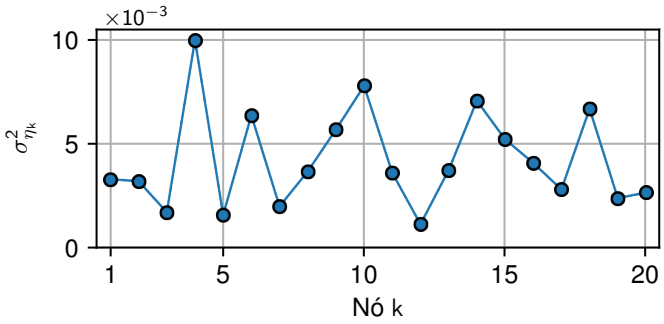


Figura 25: Variância do ruído em cada nó do grafo.
Fonte: Autor.

Nas comparações de desempenho dos algoritmos, utilizou-se o MSE, definido como

$$\mathrm{MSE}(n) \triangleq \frac{1}{V}\mathrm{E}\{\|\mathbf{d}(n) - \widehat{\mathbf{d}}(n)\|^2\}, \tag{A.17}$$

com $\widehat{\mathbf{d}}(n) = \boldsymbol{\Omega}\widehat{\mathbf{s}}(n)$ para o LMS de (LORENZO et al., 2016) ou $\widehat{\mathbf{d}}(n) = \widetilde{\mathbf{X}}(n)\mathbf{w}(n)$ para o LMS de (NASSIF et al., 2017). O MSE foi estimado com uma média de 200 realizações. Além disso, para o LMS de (LORENZO et al., 2016), considerou-se $\mathbf{T}(n) = \mathbf{I}$, em que $\mathbf{I}$ denota a matriz identidade, o que implica a amostragem de todos os nós em todos os instantes de tempo.

A.3.1 Identificação de sistemas

Inicialmente, o vetor desejado foi gerado segundo a Equação (2.26) com $\mathbf{w}^{\mathrm{o}} = [1\ 5]^{\mathrm{T}}$. A fim de testar a capacidade dos algoritmos de estimar sinais variantes no tempo, foram considerados sinais da forma

$$u_i(n) = \chi_i + c_i \sin(\omega_0 n + \varphi_i), \tag{A.18}$$

com $i = 1, 2, \cdots, 20$. Em cada realização os escalares χ_i foram gerados segundo uma distribuição gaussiana de média nula e variância unitária, ao passo que os escalares c_i e θ_i foram geradas seguindo-se distribuições uniformes nos intervalos $[0,\ 1]$ e $[0,\ 2\pi]$, respectivamente. As curvas de MSE são mostradas na Figura 26, considerando-se diferentes valores de ω_0. Os algoritmos foram ajustados para alcançar níveis semelhantes de MSE em regime permanente para $\omega_0 = 0$ nas Figuras 26a e 26b. Para isso, considerou-se $\gamma = 0{,}5\mu$ para o LMS de (LORENZO et al., 2016) e $M = 2$ para o algoritmo de (NASSIF et al., 2017).

É possível observar nas Figuras 26a e 26b que, para $\omega_0 = 0$, ambos os algoritmos atingiram patamares de MSE inferiores a -20 dB em regime permanente, o que indica que o LMS de (LORENZO et al., 2016) é capaz de estimar o vetor desejado gerado segundo (2.26), embora não estime explicitamente o sistema $\mathbf{w}^{\mathrm{o}}$. Verifica-se também que o LMS de (NASSIF et al., 2017) convergiu mais rapidamente do que o de (LORENZO et al., 2016). Entretanto, nessas mesmas figuras se observa que o desempenho em regime permanente do LMS de (LORENZO et al., 2016) piora com o aumento da frequência ω_0, o que não é observado para o LMS de (NASSIF et al., 2017). Possivelmente, este resultado está associado ao fato de que o LMS de (NASSIF et al., 2017) atualiza diretamente as estimativas dos coeficientes do sistema $\mathbf{w}^{\mathrm{o}}$, que foi considerado constante nesta simulação, ao passo que o LMS de (LORENZO et al., 2016) estima o sinal $\mathbf{x}(n)$, que neste caso variou no tempo para $\omega_0 \neq 0$.

Aumentando-se os passos de adaptação dos dois algoritmos para valores em que a divergência não foi observada em nenhuma realização, foram obtidas as curvas de MSE mostradas nas Figuras 26c e 26d. Com $\mu = 0{,}5$ e frequências $\omega_0 = \pi/2500$, $\omega_0 = \pi/10^4$ e $\omega_0 = 0$, o LMS de (LORENZO et al., 2016) alcança um patamar de MSE em regime permanente ligeiramente superior com uma velocidade de convergência maior quando comparado ao LMS de (NASSIF et al., 2017) com $\mu = 0{,}001$. No entanto, ele ainda não consegue acompanhar adequadamente as variações do sinal para $\omega_0 = \pi$. Em contrapartida, o LMS de (NASSIF et al., 2017) apresenta uma velocidade de convergência maior com $\mu = 0{,}001$, mas atinge o mesmo patamar de MSE

em regime permanente quando adaptado com $\mu = 0{,}0005$ independentemente da frequência do sinal.

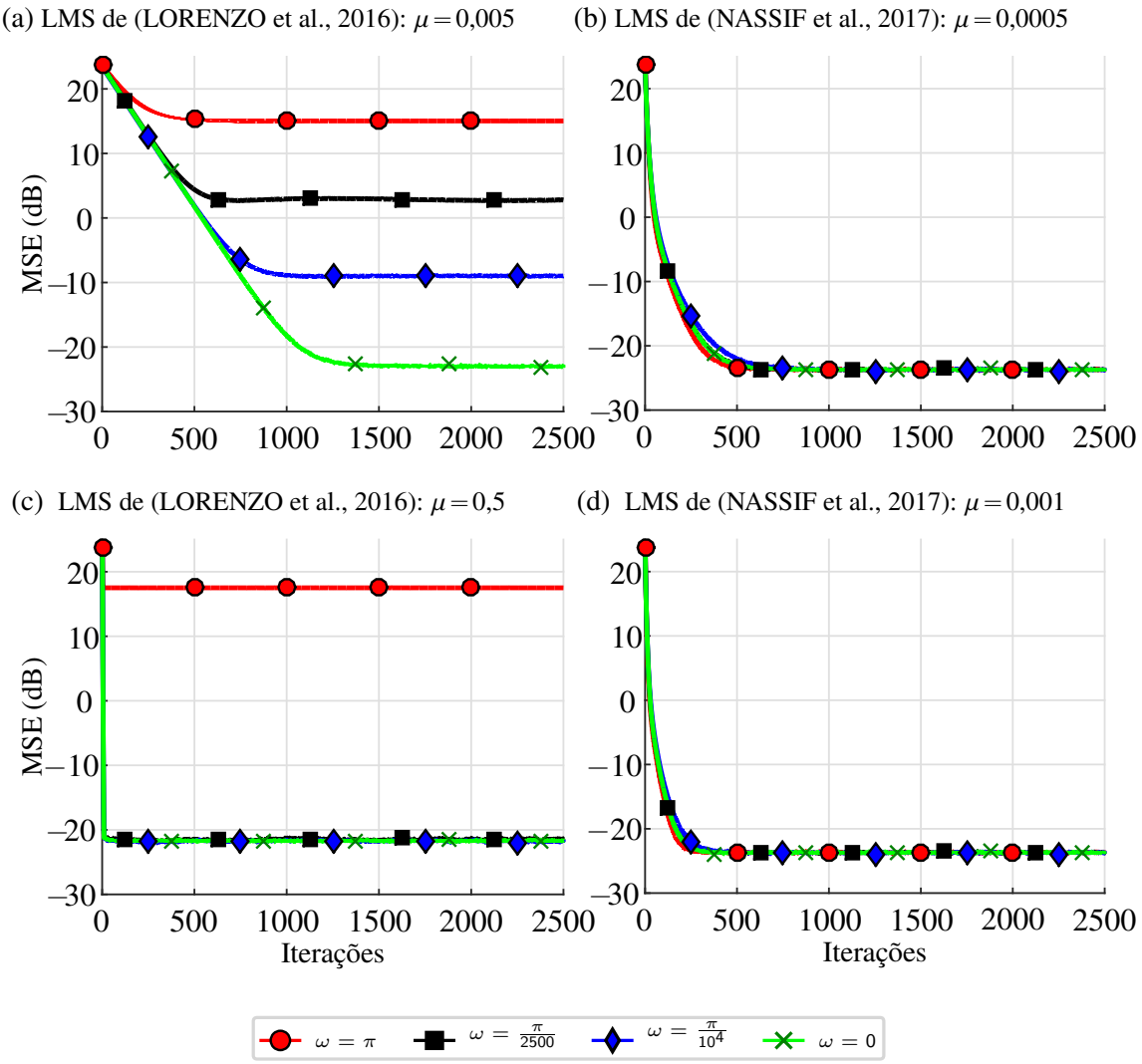


Figura 26: MSE apresentado pelos algoritmos de (LORENZO et al., 2016) e (NASSIF et al., 2017) considerando-se $\mathbf{d}(n)$ dado por (2.26) com $\mathbf{w}^{\mathrm{o}} = [1\ 5]$ e $\mathbf{u}(n)$ variando de forma senoidal no tempo.
Fonte: Autor.

A fim de testar as capacidades dos algoritmos de acompanhar variações no sistema $\mathbf{w}^{\mathrm{o}}$, também foi realizada uma simulação em que o sistema $\mathbf{w}^{\mathrm{o}} = [1\ 5\ 1\ 1]^{\mathrm{T}}$ variou abruptamente para $\mathbf{w}^{\mathrm{o}} = [1\ 5\ 1\ 1\ 0{,}1]^{\mathrm{T}}$ na iteração $n = 1{,}2 \cdot 10^5$. O sinal $\mathbf{u}(n)$ foi considerado constante no tempo, mas gerado de forma aleatória em cada realização seguindo-se uma distribuição gaussiana com média nula e variância unitária.

Considerando-se $M = 4$ e $\mu = 2{,}5 \cdot 10^{-5}$ para o LMS de (NASSIF et al., 2017) e $\mu = 0{,}005$ e $\gamma = 0{,}5\mu$ para o LMS de (LORENZO et al., 2016), foram obtidas as curvas de MSE da Figura 27. Nota-se que, ao contrário do que ocorreu nos gráficos da Figura 26, o algoritmo

de (LORENZO et al., 2016) convergiu mais rapidamente do que o de (NASSIF et al., 2017). Além disso, na iteração em que ocorre a variação do sistema, há um pico nas curvas de MSE dos dois algoritmos. Contudo, o LMS de (LORENZO et al., 2016) é capaz de voltar ao patamar de erro atingido antes da variação, enquanto que o desempenho do LMS de (NASSIF et al., 2017) piora, uma vez que o número de coeficientes usados para identificar o sistema passa a ser insuficiente. Esse comportamento indica que o LMS de (LORENZO et al., 2016) é vantajoso em situações em que o número de coeficientes do sistema é desconhecido ou varia no tempo.

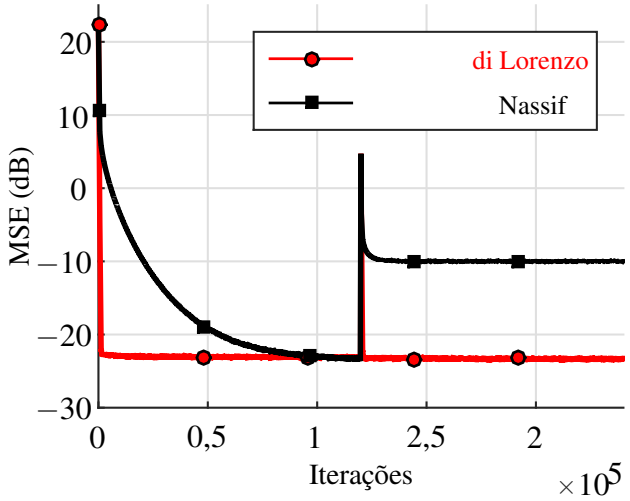


Figura 27: MSE considerando o LMS de (LORENZO et al., 2016) ($\mu = 0{,}005$) e o LMS de (NASSIF et al., 2017) ($M = 4$, $\mu = 2{,}5 \cdot 10^{-5}$), sinal no grafo constante e sistema com variação abrupta em $n = 1{,}2 \cdot 10^5$.

Fonte: Autor.

Em termos de custo computacional, é interessante mencionar que nas simulações da Figura 26, o algoritmo de (LORENZO et al., 2016) efetuou 1220 multiplicações, 1160 somas e 40 comparações por iteração. Em contrapartida, para o LMS de (NASSIF et al., 2017), com $M = 2$, foi necessário calcular 500 multiplicações e 478 somas. Já nas simulações da Figura 27, a quantidade de operações efetuadas por iteração pelo algoritmo de (LORENZO et al., 2016) se manteve, ao passo que, com $M = 4$, o LMS de (NASSIF et al., 2017) efetuou 1380 multiplicações e 1316 somas.

A.3.2 Predição

Nesta seção, são mostradas simulações em que o vetor desejado foi gerado segundo (A.9). Em vez de identificar o sistema ótimo, deseja-se agora estimar diretamente o valor do sinal $\mathbf{u}(n)$ em cada nó do grafo. A exemplo do que foi feito na Subseção A.3.1, foram considerados sinais na forma da Equação (A.18) considerando-se diferentes valores de ω_0. Na Figura 26 são

mostradas as curvas de MSE obtidas com os algoritmos de (LORENZO et al., 2016) e (NASSIF et al., 2017) com diferentes passos de adaptação. Para o LMS de (LORENZO et al., 2016) considerou-se novamemte $\gamma = 0{,}25\mu$. Já para o LMS de (NASSIF et al., 2017) considerou-se $M = 1$.

Analisando-se a Figura 28a, é possível verificar que novamente o desempenho em regime permanente do LMS de (LORENZO et al., 2016) piora com o aumento da frequência ω_0. Além disso, comparando-se com os resultados da Figura 28b, constata-se que ambos os algoritmos atingiram patamares de MSE semelhantes para $\omega_0 = 0$, com o LMS de (NASSIF et al., 2017) convergindo mais rapidamente.

Ainda de modo semelhante ao realizado na Subseção A.3.1, aumentaram-se os passos de adaptação dos dois algoritmos até os valores máximos em que a divergência não foi observada em nenhuma realização. Com isso, foram obtidas as curvas de MSE mostradas nas Figuras 28c e 28d. Nota-se que com $\mu = 0{,}5$ o LMS de (LORENZO et al., 2016) alcança um patamar de MSE em regime permanente ligeiramente superior com uma velocidade de convergência maior quando comparado ao LMS de (NASSIF et al., 2017) com $\mu = 0{,}001$ para as frequências $\omega_0 = \pi/2500$ e $\omega_0 = \pi/10^4$. No entanto, para $\omega_0 = \pi$, o algoritmo de (LORENZO et al., 2016) foi novamente incapaz de acompanhar de maneira satisfatória as variações no sinal.

Por fim, cabe mencionar que nas simulações apresentadas nesta subseção o algoritmo de (LORENZO et al., 2016) efetuou, a cada iteração, 1220 multiplicações, 1160 somas e 40 comparações, ao passo que o LMS de (NASSIF et al., 2017) efetuou 60 multiplicações e 59 somas.

A.4 Conclusões

Neste apêndice, foram apresentados conceitos complementares da teoria de processamento de sinais em grafos, bem como algoritmos adaptativos voltados a esse tipo de aplicação com processamento centralizado. Além disso, foram feitas comparações entre os algoritmos propostos em (LORENZO et al., 2016) e (NASSIF et al., 2017) em cenários envolvendo identificação de sistemas e predição. Para sinais variantes no tempo, os resultados indicam que o algoritmo de (LORENZO et al., 2016) é mais sensível a variações temporais do sinal, o que não ocorre com o algoritmo de (NASSIF et al., 2017). Os resultados também sugerem que adoção de passos de adaptação mais elevados pode reduzir essa sensibilidade para algumas taxas de variação

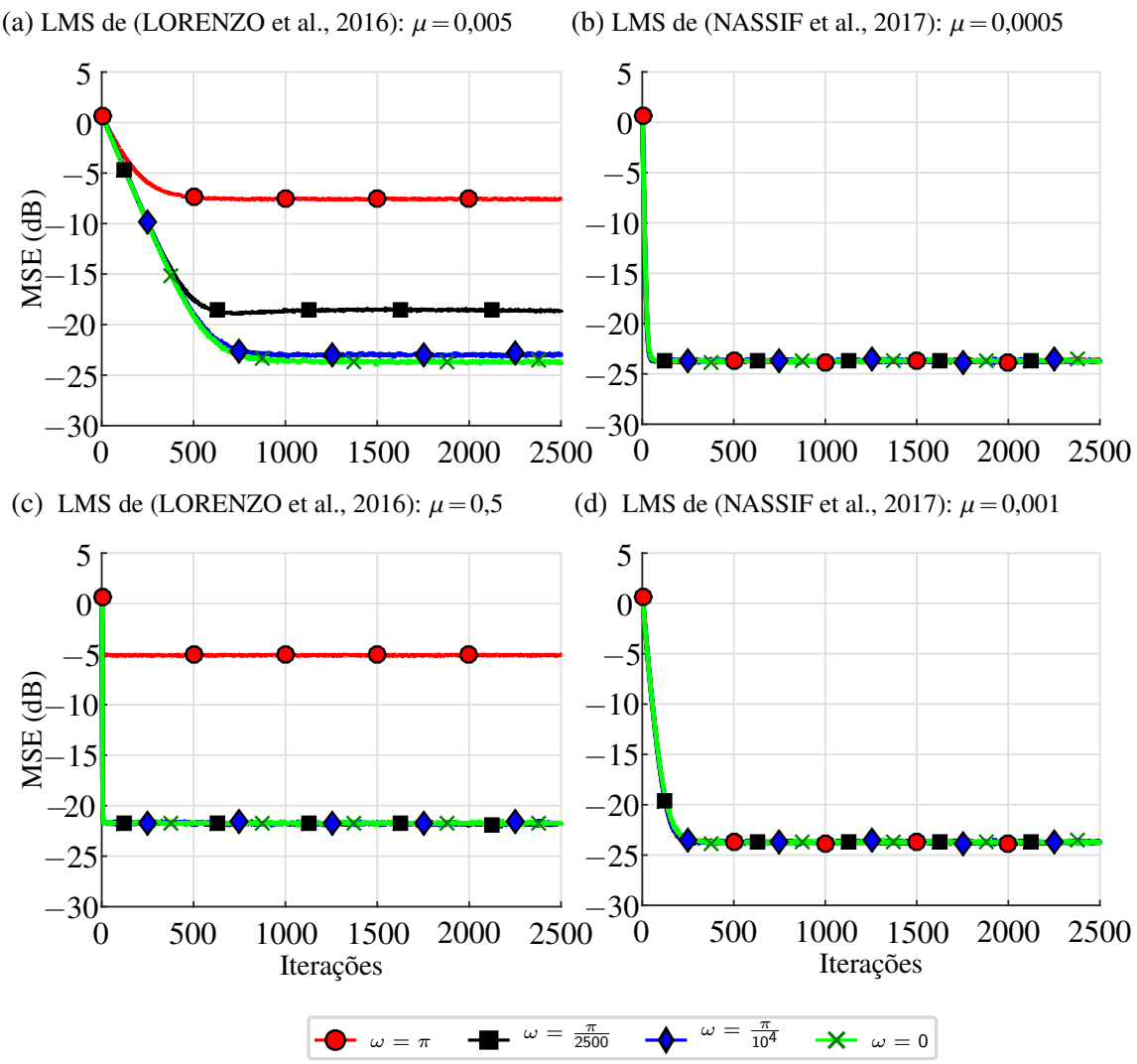


Figura 28: MSE apresentado pelos algoritmos de (LORENZO et al., 2016) e (NASSIF et al., 2017) considerando-se $\mathbf{d}(n)$ dado por (A.9) com $\mathbf{u}(n)$ variando de forma senoidal no tempo.
Fonte: Autor.

do sinal. Verificou-se ainda que quando o vetor desejado é gerado segundo a Equação (2.26), o algoritmo de (NASSIF et al., 2017) pode sofrer mais com alterações no sistema ótimo do que o algoritmo de (LORENZO et al., 2016). Do ponto de vista do custo computacional, verificou-se que quanto maior o número de coeficientes utilizados no LMS de (NASSIF et al., 2017), maior é o custo desse algoritmo em comparação com o LMS de (LORENZO et al., 2016).