

Amostragem Adaptativa para Redes de Difusão com Ajuste de Parâmetros em Tempo Real

Daniel G. Tiglea, Renato Candido e Magno T. M. Silva

Resumo—Recentemente, foi proposto um algoritmo de amostragem para redes de difusão que adapta localmente o número de nós amostrados segundo o erro de estimação, gerando economia energética e computacional. Seu desempenho depende da escolha do parâmetro responsável por penalizar a amostragem, que é função da potência do ruído. Escolhas inadequadas afetam a convergência após mudanças no ambiente. Neste artigo, esse parâmetro é ajustado automaticamente baseado na estimação adaptativa da potência do ruído. Embora aumente ligeiramente o custo computacional, tal modificação melhora a capacidade de rastreamento, reduz a influência de nós ruidosos e ainda é vantajosa na presença de ruído impulsivo.

Palavras-Chave—Redes de difusão adaptativas, amostragem, processamento distribuído de sinais, ruído impulsivo.

Abstract—Recently, we proposed a sampling algorithm for diffusion networks, which adapts locally the number of sampled nodes based on the estimation error. It reduces energy consumption and the computational cost. Its performance depends on the choice of the parameter that penalizes sampling, which is a function of the noise power. Inadequate choices of this parameter affect its tracking capability. In this paper, this parameter is automatically adjusted based on an adaptive estimation of the noise power. Although this modification slightly increases the computational cost, it improves the tracking capability, reduces the influence of noisy nodes and is advantageous in the presence of impulsive noise.

Keywords—Adaptive diffusion networks, sampling, distributed signal processing, impulsive noise.

I. INTRODUÇÃO

Nas últimas décadas, redes de difusão adaptativas se consolidaram como uma ferramenta interessante para o processamento distribuído de sinais. Utilizando nós espacialmente distribuídos e conectados entre si por meio de uma topologia, elas têm se mostrado uma solução eficiente em diversas aplicações, tais como localização e rastreamento de alvos, sensoriamento de espectro em redes móveis, aplicações médicas, entre outras [1]–[12]. Contudo, em certas situações, o custo associado à medição e processamento dos dados em cada nó é proibitivo, tornando necessário algum mecanismo de amostragem [13], [14]. Além disso, nas redes de sensores sem fio, o consumo energético é frequentemente uma limitação crítica. Nesses casos, é comum empregar técnicas de *censura*, que limitam a quantidade de transmissões entre os nós [11], [12].

Recentemente, foi proposto em [15] um algoritmo de amostragem para redes de difusão adaptativas que também pode ser usado como uma técnica de censura. Ele muda o número de nós amostrados com base no erro quadrático na vizinhança de

cada nó. Assim, o número de nós amostrados diminui quando o erro é de baixa magnitude, permitindo uma rápida convergência e uma redução significativa no custo computacional e/ou consumo energético em regime permanente. Entretanto, como a amostragem dos nós não cessa permanentemente, o mecanismo pode detectar mudanças no ambiente. Esse algoritmo se mostrou bem-sucedido em problemas de estimativa de temperatura usando dados de estações meteorológicas do Brasil, bem como em simulações com dados sintéticos [15]. No entanto, a seleção adequada de seus parâmetros depende do valor da maior potência do ruído de medição dentre os nós da rede, que pode não ser conhecida de antemão. Escolhas inadequadas podem levar a uma redução excessiva no número de nós amostrados, comprometendo o desempenho do algoritmo, ou à manutenção da amostragem de todos os nós, acarretando aumento do custo computacional, em vez de redução. Utilizando-se a técnica de [16] para estimar a potência do ruído, neste artigo, é proposta uma versão alternativa do referido algoritmo, que estima a variância do ruído em cada nó e ajusta seus próprios parâmetros de maneira distribuída e em tempo real, dispensando a necessidade de conhecimento *a priori* da potência do ruído. Além disso, esta abordagem melhora a resposta do algoritmo a mudanças abruptas no ambiente e é vantajosa na presença de ruído impulsivo.

Este trabalho está organizado da seguinte forma. Na Seção II, revisita-se o algoritmo AS-dNLMS (*adaptive sampling diffusion normalized least-mean-squares*) de [15]. Na Seção III, obtém-se o algoritmo de amostragem com ajuste automático de parâmetros. Resultados de simulação e as conclusões são apresentados nas Seções IV e V, respectivamente. **Notação.** São utilizadas letras em fontes convencionais para escalares e minúsculas em negrito para vetores. Além disso, $|\cdot|$ denota a cardinalidade, $(\cdot)^T$ transposição, $E\{\cdot\}$ a esperança matemática e $\|\cdot\|$ a norma euclidiana.

II. O ALGORITMO AS-dNLMS

Consideremos uma rede com V nós e uma topologia predefinida. Dois nós são considerados vizinhos se eles podem trocar informações. Nesse caso, a vizinhança do nó k é dada pelo conjunto de seus vizinhos, incluindo o próprio nó k , e é denotada por \mathcal{N}_k . A cada iteração, cada nó k tem acesso a um sinal de entrada $u_k(n)$ e a um sinal de referência $d_k(n) = \mathbf{u}_k^T(n)\mathbf{w}^o + v_k(n)$, em que $\mathbf{u}_k(n) = [u_k(n) \ u_k(n-1) \ \dots \ u_k(n-M+1)]^T$ é um vetor regressor de comprimento M , \mathbf{w}^o é o sistema que se pretende estimar, também chamado na literatura de “sistema ótimo”, e $v_k(n)$ é o ruído de medição no nó k . Esse ruído é considerado independente de qualquer outra variável e estacionário no sentido amplo com média nula e variância $\sigma_{v_k}^2$. O objetivo

da rede é estimar \mathbf{w}° de maneira distribuída resolvendo-se $\min_{\mathbf{w}} \sum_{k=1}^V \mathbb{E}\{[d_k(n) - \mathbf{u}_k^T(n)\mathbf{w}]^2\}$ [1]–[3], [10], [11].

Muitos algoritmos foram propostos para esse problema, sendo o dNLMS um dos mais utilizados [1]–[3]. Com base nele, foi proposto em [15] o algoritmo AS-dNLMS. Seguindo uma configuração do tipo ATC (*adapt-then-combine*)¹, as suas etapas de adaptação e de combinação são respectivamente

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

em que $\bar{s}_k(n) \in \{0,1\}$ é uma variável de amostragem e $\boldsymbol{\psi}_k$ e \mathbf{w}_k representam respectivamente as estimativas local e combinada de \mathbf{w}° no nó k . Além disso,

$$e_k(n) = d_k(n) - \mathbf{u}_k^T(n)\mathbf{w}_k(n), \quad (2)$$

é o erro de estimação e $\mu_k(n) = \tilde{\mu}_k/[\delta + \|\mathbf{u}_k(n)\|^2]$ é um passo de adaptação normalizado, com $0 < \tilde{\mu}_k < 2$ e um pequeno fator de regularização $\delta > 0$ [1]. Por fim, $\{c_{jk}\}$ são pesos de combinação que satisfazem $c_{jk} \geq 0$, $\sum_{j \in \mathcal{N}_k} c_{jk} = 1$ e $c_{jk} = 0$ se $j \notin \mathcal{N}_k$ [2], [3]. Possíveis escolhas para $\{c_{jk}\}$ incluem as regras Uniforme, Metropolis e do Grau Relativo [1], além de esquemas adaptativos como o algoritmo ACW (*adaptive combination weights*) [17]. Esse algoritmo é obtido ao se resolver um problema de otimização com relação aos $\{c_{jk}\}$ e procura atribuir pesos maiores aos nós menos ruidosos. Para evitar divisão por zero, neste trabalho adota-se uma versão regularizada do ACW, cujas equações são dadas por [15], [17]

$$c_{jk}(n) = \frac{[\delta_c + \hat{\sigma}_{jk}^2(n)]^{-1}}{\sum_{\ell \in \mathcal{N}_k} [\delta_c + \hat{\sigma}_{\ell k}^2(n)]^{-1}}, \text{ se } j \in \mathcal{N}_k, \text{ ou } 0, \text{ c.c.}, \quad (3)$$

em que $\delta_c > 0$ é o fator de regularização e

$$\hat{\sigma}_{jk}^2(n) = (1 - \nu_k)\hat{\sigma}_{jk}^2(n-1) + \nu_k\|\boldsymbol{\psi}_j(n+1) - \mathbf{w}_k(n)\|^2, \quad (4)$$

com $\nu_k > 0$ para $k=1, \dots, V$.

A diferença entre os algoritmos dNLMS e AS-dNLMS está na inserção da variável de amostragem $\bar{s}_k(n)$ no termo de correção da Eq. (1a). Quando $\bar{s}_k(n) = 1$, $d_k(n)$ é amostrado e $e_k(n)$ é calculado como em (2). Em contrapartida, se $\bar{s}_k(n) = 0$, $d_k(n)$ não é amostrado, $\mathbf{u}_k^T(n)\mathbf{w}_k(n)$, $e_k(n)$ e $\mu_k(n)$ não são calculados e $\boldsymbol{\psi}_k(n+1) = \mathbf{w}_k(n)$.

Com base na combinação convexa de filtros adaptativos, em vez de adaptar $\bar{s}_k(n)$ diretamente, introduz-se uma variável auxiliar $\alpha_k(n) \in [-\alpha^+, \alpha^+]$ tal que $\bar{s}_k(n) = 0$ para $\phi[\alpha_k(n)] < 0,5$ e $\bar{s}_k(n) = 1$ caso contrário, com $\phi[\cdot]$ dada por [15], [18]

$$\phi[\alpha_k(n)] \triangleq \frac{\text{sgm}[\alpha_k(n)] - \text{sgm}[-\alpha^+]}{\text{sgm}[\alpha^+] - \text{sgm}[-\alpha^+]}, \quad (5)$$

em que $\text{sgm}[x] = (1 + e^{-x})^{-1}$ é a função sigmoide. Na literatura, normalmente se adota $\alpha^+ = 4$ [18]. Para tornar a notação mais compacta, $\phi[\alpha_k(n)]$ será escrito aqui como $\phi_k(n)$. Cabe notar que $\phi[\alpha^+] = 1$, $\phi[0] = 0,5$ e $\phi[-\alpha^+] = 0$. Assim, $\bar{s}_k(n)$ pode ser relacionado a $\alpha_k(n)$ por meio de

$$\bar{s}_k(n) = \begin{cases} 1, & \text{se } \alpha_k(n) \geq 0, \\ 0, & \text{c.c.} \end{cases} \quad (6)$$

Procura-se então minimizar a seguinte função custo [15]

$$J_{\alpha_k}(n) = \phi_k(n)\beta\bar{s}_k(n) + [1 - \phi_k(n)]\sum_{i \in \mathcal{N}_k} c_{ik}(n)e_i^2(n) \quad (7)$$

¹Neste trabalho, apenas a abordagem ATC será considerada, mas os resultados podem ser estendidos para a estratégia CTA (*combine-then-adapt*), em que a ordem das Equações (1a) and (1b) é trocada.

com relação a $\alpha_k(n)$, em que $\beta > 0$ é um parâmetro introduzido para controlar a penalização da amostragem. Quando o erro é elevado em magnitude, $J_{\alpha_k}(n)$ é minimizada fazendo-se $\phi_k(n)$ próximo de um, o que leva à amostragem do nó k . O mesmo vale quando o nó k não é amostrado ($\bar{s}_k = 0$), o que garante que a sua amostragem seja retomada em algum momento. Em contrapartida, quando o nó k é amostrado ($\bar{s}_k = 1$) e o erro é pequeno em magnitude, $J_{\alpha_k}(n)$ é minimizada fazendo-se $\phi_k(n)$ próximo de zero, o que faz com que o algoritmo deixe de amostrar o nó k [15].

O ajuste de $\alpha_k(n)$ é obtido tomando-se a derivada de (7) em relação a $\alpha_k(n)$ e utilizando-se o método do gradiente estocástico. Como não se tem acesso ao erro $e_i(n)$ quando o nó i deixa de ser amostrado, ele é substituído pela sua última medição, denotada por $\varepsilon_i(n)$. Assim, obtém-se [15]

$$\alpha_k(n+1) = \alpha_k(n) + \mu_s \phi'_k(n) \left[\sum_{i \in \mathcal{N}_k} c_{ik}(n) \varepsilon_i^2(n) - \beta \bar{s}_k(n) \right], \quad (8)$$

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

$$\phi'_k(n) = \frac{d\phi[\alpha_k(n)]}{d\alpha_k(n)} = \frac{\text{sgm}[\alpha_k(n)]\{1 - \text{sgm}[\alpha_k(n)]\}}{\text{sgm}[\alpha^+] - \text{sgm}[-\alpha^+]}. \quad (9)$$

As Equações (1) e (8) formam a base do algoritmo AS-dNLMS. Ele preserva a taxa de convergência do dNLMS original, mas apresenta um custo computacional menor em regime, embora haja um pequeno aumento durante o transitório [15]. Cabe notar que o algoritmo requer que cada nó amostrado i transmita $\varepsilon_i^2(n) = e_i^2(n)$ para os seus vizinhos. Entretanto, essa informação pode ser enviada junto com as estimativas locais $\boldsymbol{\psi}_i$, de modo a não aumentar o número de transmissões. Na versão do algoritmo voltada à censura, denominada ASC-dNLMS (*adaptive-sampling-and-censoring diffusion NLMS*), a etapa de adaptação (1a) não é executada quando o nó não é amostrado. Assim, $\boldsymbol{\psi}_k$ e ε_k permanecem inalterados nessas iterações. Considerando que os nós sejam capazes de armazenar os dados enviados pelos vizinhos, isso permite reduzir o número de transmissões. Por fim, cabe observar que se o algoritmo AS-dNLMS for implementado em conjunto com o ACW, o mecanismo de amostragem pode prejudicar a atualização dos pesos de combinação se a amostragem do nó k for interrompida por um longo período de tempo. Nesse caso, $\hat{\sigma}_{kk}^2$ poderia tender a zero em (4). A fim de evitar isso, para $j = k$, substitui-se $\boldsymbol{\psi}_j(n+1)$ em (4) por $\bar{\boldsymbol{\psi}}_k(n+1) \triangleq \bar{s}_k(n)\boldsymbol{\psi}_k(n+1) + [1 - \bar{s}_k(n)]\bar{\boldsymbol{\psi}}_k(n)$.

O parâmetro β desempenha um papel crucial no comportamento do algoritmo AS-dNLMS, pois influencia diretamente o número esperado de nós amostrados. Para que a amostragem do nó k cesse, o valor de α_k deve cair ao longo das iterações até se tornar negativo. Portanto, o termo entre colchetes em (8) deve ser negativo em regime permanente quando $\bar{s}_k(n) = 1$. Para isso, considerando que esse termo seja estatisticamente independente de $\phi'_k(n)$ e que os pesos $\{c_{ik}\}$ sejam constantes e previamente determinados, deve-se ter [15]

$$\beta > \mathbb{E} \left\{ \sum_{i \in \mathcal{N}_k} c_{ik} \varepsilon_i^2(n) \right\}. \quad (10)$$

Considerando $\mathbb{E}\{\varepsilon_i^2(n)\} \approx \sigma_{v_i}^2$ para $i = 1, \dots, V$ em regime permanente e analisando-se o cenário de pior caso, tem-se que

$$\beta > \sigma_{\max}^2 \triangleq \max_i \sigma_{v_i}^2 \quad (11)$$

é uma condição suficiente (mas não necessária) para que na

média haja redução no número de nós amostrados em regime permanente [15]. Entretanto, resultados de simulação mostram que a adoção de valores elevados para β pode prejudicar a capacidade de rastreamento (*tracking*) do algoritmo. Diante disso, a escolha adequada de β se torna difícil se σ_{\max}^2 não for conhecida de antemão.

Além disso, também é preciso selecionar μ_s adequadamente, já que o passo afeta o quão rapidamente os nós deixam de ser amostrados. Aproximando-se $\phi'_k(n)$ por uma reta para $\alpha_k \in [0, \alpha^+]$, pode-se mostrar que, para que a amostragem dos nós cesse em até Δ_n iterações após o algoritmo alcançar o regime permanente, deve-se fazer [15]

$$\mu_s > \frac{\alpha^+}{(\beta - \sigma_{\max}^2)(\phi'_0 - \phi'_{\alpha^+})} \left[\left(\frac{\phi'_0}{\phi'_{\alpha^+}} \right)^{\frac{1}{\Delta_n}} - 1 \right], \quad (12)$$

em que ϕ'_0 e ϕ'_{α^+} são constantes que representam o valor de $\phi'[\cdot]$ em $\alpha_k = 0$ e $\alpha_k = \alpha^+$, respectivamente.

III. AJUSTE AUTOMÁTICO DOS PARÂMETROS

Em vez de atribuir um valor comum de β para todos os nós, pode-se permitir que cada nó k tenha um parâmetro local β_k . Assim, substituindo-se β por β_k em (10) e mantendo-se a hipótese de que $E\{\varepsilon_i^2(n)\} \approx \sigma_{v_i}^2$ em regime permanente, pode-se concluir que

$$\beta_k > \sum_{i \in \mathcal{N}_k} c_{ik} \sigma_{v_i}^2 \triangleq \sigma_{\mathcal{N}_k}^2 \quad (13)$$

é uma condição necessária e suficiente para que a amostragem do nó k cesse em algum momento durante o regime permanente. Considerando que cada nó k possa calcular uma estimativa $\hat{\sigma}_{v_k}^2(n)$ de $\sigma_{v_k}^2$ em tempo real e de forma distribuída e que eles possam transmitir tais estimativas aos seus vizinhos, pode-se escrever

$$\beta_k(n) = \gamma \sum_{i \in \mathcal{N}_k} c_{ik} \hat{\sigma}_{v_i}^2(n) \triangleq \gamma \hat{\sigma}_{\mathcal{N}_k}^2(n), \quad (14)$$

em que $\gamma > 1$ é um parâmetro que o projetista deve escolher.

A literatura contém diversas técnicas para estimar $\sigma_{v_k}^2$ [6], [16], [17]. Dentre elas, a proposta em [16] apresenta uma maior taxa de convergência em comparação com outros métodos e é capaz de lidar com alterações no ambiente, desde que $v_k(n)$ seja estacionário no sentido amplo para $k=1, \dots, V$.

O algoritmo de [16] estima a variância do ruído no nó k a cada iteração com base no valor de $e_k^2(n)$. Quando o nó k é amostrado, três filtros passa-baixas com fatores de esquecimento λ_f , λ_m e λ_s são aplicados, obtendo-se as estimativas $\theta_{f_k}^2$, $\theta_{m_k}^2$ e $\theta_{s_k}^2$ dadas por

$$\theta_{f_k}^2(n) = \lambda_f \theta_{f_k}^2(n-1) + (1 - \lambda_f) e_k^2(n), \quad (15)$$

$$\theta_{m_k}^2(n) = \lambda_m \theta_{m_k}^2(n-1) + (1 - \lambda_m) e_k^2(n) \quad (16)$$

e

$$\theta_{s_k}^2(n) = \lambda_s \theta_{s_k}^2(n-1) + (1 - \lambda_s) e_k^2(n). \quad (17)$$

Em contrapartida, quando o nó k não é amostrado, $\theta_{f_k}^2$, $\theta_{m_k}^2$ e $\theta_{s_k}^2$ permanecem fixos. Em [16], sugere-se usar $\lambda_f = 1 - 1/(5M)$, $\lambda_m = 1 - 1/(15M)$ e $\lambda_s = 1 - 1/(45M)$. Como $\lambda_f > \lambda_m > \lambda_s$, $\theta_{f_k}^2$ converge rapidamente, o que o faz responder mais rápido a alterações no ambiente. Entretanto, essa estimativa é afetada mais intensamente por perturbações

em $e_k^2(n)$. Em contrapartida, $\theta_{s_k}^2$ fornece uma estimativa mais precisa em regime permanente, mas demora mais para convergir e perceber alterações no ambiente. Por fim, $\theta_{m_k}^2$ apresenta um comportamento intermediário [16]. Se nenhuma mudança for detectada no ambiente, ou seja, quando o algoritmo se encontra no seu estado de operação “normal”, uma estimativa intermediária $\theta_{v_k}^2$ é calculada usando [16]

$$\theta_{v_k}^2(n) = \lambda_f \theta_{v_k}^2(n-1) + (1 - \lambda_f) \theta_{\min_k}^2, \quad (18)$$

em que

$$\theta_{\min_k}^2 \triangleq \min\{\theta_{f_k}^2(n), \theta_{m_k}^2(n), \theta_{s_k}^2(n)\}. \quad (19)$$

Independentemente do modo de operação, a estimativa $\hat{\sigma}_{v_k}^2(n)$ é obtida por meio de [16]

$$\hat{\sigma}_{v_k}^2(n) = \min\{\theta_{v_k}^2(n), \theta_{f_k}^2(n)\}. \quad (20)$$

Como alterações no ambiente tendem a ser detectadas primeiro pelas estimativas mais rápidas, o mecanismo de [16] entra no modo de “detecção de mudança” quando

$$\theta_{f_k}^2(n) > \theta_{s_k}^2(n), \quad (21)$$

a não ser que o algoritmo ainda esteja no transitório. Para isso, utiliza-se uma variável auxiliar, que indica se esse modo já foi alcançado anteriormente ou não. Caso seja a primeira vez que o algoritmo entre no modo de detecção, ou se $\theta_{f_k}^2(n) < \theta_{m_k}^2(n)$, $\theta_{v_k}^2$ é calculado por meio de

$$\theta_{v_k}^2(n) = \lambda_m \theta_{v_k}^2(n-1) + (1 - \lambda_m) \theta_{\min_k}^2(n), \quad (22)$$

de forma a mitigar o impacto de flutuações no erro sobre a estimativa da potência do ruído. Em contrapartida, se $\theta_{f_k}^2(n) \geq \theta_{m_k}^2(n)$, considera-se que há uma mudança significativa no ambiente. Para evitar que isso contamine as estimativas, $\theta_{v_k}^2$ é mantido fixo até que se verifique que $\theta_{m_k}^2(n) < \theta_{s_k}^2(n)$, o que leva o algoritmo de volta ao modo de operação normal.

Convém ressaltar que $\hat{\sigma}_{v_k}^2(n)$ pode ser calculado localmente, já que as Equações (15) a (22) utilizam apenas informações disponíveis no próprio nó. Contudo, para calcular (14), cada nó i amostrado deve transmitir a sua estimativa $\hat{\sigma}_{v_i}^2(n)$ para os seus vizinhos. Entretanto, se $\hat{\sigma}_{v_i}^2(n)$ for enviada junto com $\varepsilon_i^2(n)$ e $\psi_i(n)$, isso não aumenta o número de transmissões.

Como agora há um valor de $\beta_k(n)$ diferente para cada nó em vez de um parâmetro global β , o ajuste de μ_s segundo (12) deve ser alterado, levando a um passo $\mu_{s_k}(n)$ para cada nó k . Além disso, como não se presume mais o conhecimento prévio de σ_{\max}^2 , esse termo é substituído por $\hat{\sigma}_{\mathcal{N}_k}^2(n)$ em (12). Por fim, considerando-se (14), pode-se escrever

$$\mu_{s_k}(n) > \frac{1}{\hat{\sigma}_{\mathcal{N}_k}^2(n)} \left\{ \frac{\alpha^+}{(\gamma - 1)(\phi'_0 - \phi'_{\alpha^+})} \left[\left(\frac{\phi'_0}{\phi'_{\alpha^+}} \right)^{\frac{1}{\Delta_n}} - 1 \right] \right\}. \quad (23)$$

Convém notar que o termo entre colchetes na Eq. (23) é constante uma vez que γ , α^+ e Δ_n tenham sido selecionados.

Essa versão do algoritmo é denominada DTAS-dNLMS (*dynamic-tuning AS-dNLMS*). Em comparação com o algoritmo AS-dNLMS, ela requer $1 + 8\bar{s}_k(n) + |\mathcal{N}_k|$ multiplicações a mais, considerando parâmetros β e γ ajustados para obter ao mesmo número esperado de nós amostrados em regime. Além de dispensar o conhecimento prévio de σ_{\max}^2 , essa versão leva a um desempenho melhor quando a variância do ruído em

determinados nós é muito maior do que a verificada no restante da rede, como será mostrado na Seção IV.

É importante mencionar que, embora apenas pesos de combinação fixos tenham sido considerados nesta seção, os algoritmos AS-dNLMS e DTAS-dNLMS podem ser implementados com esquemas adaptativos de seleção dos $\{c_{ik}\}$.

IV. RESULTADOS DE SIMULAÇÃO

Nesta seção, são apresentados resultados de simulação para ilustrar o comportamento do algoritmo DTAS-dNLMS, obtidos como uma média de 100 realizações independentes. Para uma melhor visualização, aplicou-se um filtro de média móvel com 64 coeficientes às curvas apresentadas.

Nas simulações, considera-se uma rede com 20 nós, sendo que para metade deles foi adotado $\tilde{\mu}_k = 0,1$, enquanto para a outra metade adotou-se $\tilde{\mu}_k = 1$, como mostrado na Fig. 1(a). A rede foi gerada de forma aleatória seguindo o modelo de Erdős-Renyi [19], sendo que cada nó tem em média 5 vizinhos. Além disso, cada nó k está sujeito a um ruído gaussiano branco de variância $\sigma_{v_k}^2$, como mostrado na Fig. 1(b). O sinal de entrada $u_k(n)$ também é gaussiano branco, mas com variância unitária para todos os nós $k = 1, \dots, V$. Para o sistema ótimo \mathbf{w}^o , considera-se um vetor aleatório com $M = 50$ coeficientes uniformemente distribuídos no intervalo $[-1, 1]$.

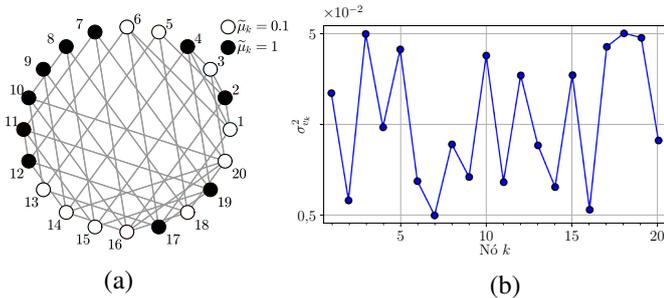


Fig. 1: (a) Rede considerada nas simulações. (b) Variância $\sigma_{v_k}^2$ do ruído para $k = 1, \dots, V$.

Os pesos de combinação são escolhidos utilizando-se o algoritmo ACW com $\nu_k = 0,2$ para $k = 1, \dots, V$ [17]. Além disso, adotam-se $\delta = 10^{-5}$ e $\delta_c = 10^{-8}$ como fatores de regularização. Como métrica de desempenho, adota-se o NMSD (*network mean-square-deviation*), dado por

$$\text{NMSD}(n) = \frac{1}{V} \sum_{k=1}^V \mathbb{E}\{\|\mathbf{w}^o(n) - \mathbf{w}_k(n)\|^2\}. \quad (24)$$

Primeiramente, é feita uma comparação entre os algoritmos DTAS-dNLMS, AS-dNLMS e o algoritmo dNLMS com V_s nós amostrados aleatoriamente a cada iteração, sendo $V_s \in \{5, 20\}$. Cabe observar que $V_s = 20$ corresponde ao caso em que todos os nós são amostrados. Para simular uma mudança abrupta no ambiente, considera-se a partir da metade de cada realização um sistema ótimo \mathbf{w}^o igual ao vetor original multiplicado por um fator de 0,25. Para cada algoritmo, as curvas de NMSD, o número médio de nós amostrados e de multiplicações são mostrados ao longo das iterações nas Figuras 2(a), 2(b) e 2(c), respectivamente. Os valores de β e γ adotados respectivamente para os algoritmos AS-dNLMS e DTAS-dNLMS foram escolhidos de modo a igualar o número

de nós amostrados em regime para ambos os algoritmos e manter o bom desempenho nos dois casos. Por sua vez, adotou-se o mesmo Δ_n para as duas soluções nas Eqs. (12) e (23). Esse valor foi selecionado de modo a garantir que os mecanismos de amostragem entrassem em ação após a convergência em termos de NMSD. Pode-se observar que a amostragem aleatória afeta significativamente a taxa de convergência, ao passo que os algoritmos DTAS-dNLMS e AS-dNLMS preservaram a taxa de convergência do dNLMS original, já que mantiveram a amostragem de todos os nós no transitório. Além disso, eles foram capazes de detectar a mudança abrupta no sistema ideal e aumentaram o número de nós amostrados, o que novamente os levou a uma convergência semelhante à do dNLMS. Por fim, os dois algoritmos apresentam um ligeiro aumento no custo computacional durante o transitório em comparação com o dNLMS com todos os nós amostrados, o que é compensado pela economia em regime permanente. Observa-se ainda que o custo computacional do algoritmo DTAS-dNLMS é sempre ligeiramente superior ao do AS-dNLMS, o que era esperado.

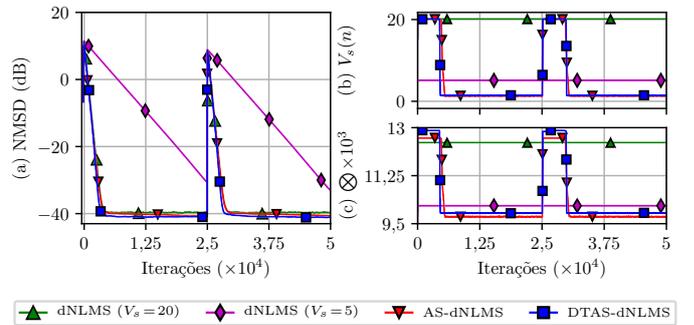


Fig. 2: Comparação entre os algoritmos dNLMS com V_s nós amostrados aleatoriamente ($V_s \in \{5, 20\}$), AS-dNLMS ($\beta = 0,2$, $\Delta_n = 3 \cdot 10^3$) e DTAS-dNLMS ($\gamma = 8,5$, $\Delta_n = 3 \cdot 10^3$): (a) curvas de NMSD e (b) Número de nós amostrados, e (c) Número de multiplicações por iteração.

Nas simulações da Fig. 3, considerou-se o mesmo cenário da Fig. 2, exceto pelo fato de que a máxima variância do ruído na rede, observada no nó 18 na Fig. 1(b), passa a ser dada por $\sigma_{\max}^2 = \sigma_{v_{18}}^2 = 0,5$ em vez de $\sigma_{\max}^2 = 0,05$ como considerado anteriormente. Nesse caso, mantiveram-se os parâmetros do algoritmo DTAS-dNLMS, ao passo que β e μ_s foram reajustados para o AS-dNLMS utilizando as Eqs. (11) e (12). Nesse cenário, observa-se o comprometimento da capacidade de rastreamento do AS-dNLMS. Após a mudança abrupta, o algoritmo apresenta uma taxa de convergência lenta, uma vez que o número de nós amostrados cai rapidamente após o pico provocado pela alteração. Em contrapartida, o DTAS-dNLMS mantém a amostragem por mais tempo após a mudança no ambiente e tem a sua capacidade de rastreamento preservada. Isso pode ser atribuído à adoção de parâmetros β_k diferentes para cada nó, o que diminui o impacto dos nós mais ruidosos na amostragem da rede como um todo, limitando-o às suas vizinhanças. Cabe ressaltar que, embora se possa adotar $\beta \leq \sigma_{\max}^2$ para o AS-dNLMS [15], esse ajuste tende a ser mais difícil, já que passa a requerer um conhecimento mais aprofundado do perfil do ruído em toda a rede.

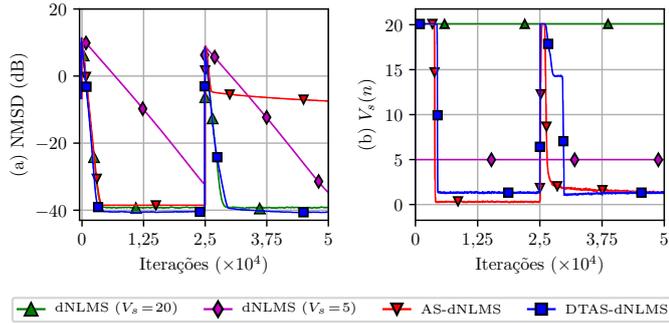


Fig. 3: Comparação entre os algoritmos dNLMS com V_s nós amostrados aleatoriamente ($V_s \in \{5, 20\}$), AS-dNLMS ($\beta = 2$, $\Delta_n = 3 \cdot 10^3$) e DTAS-dLMS ($\gamma = 8,5$, $\Delta_n = 3 \cdot 10^3$): (a) curvas de NMSD e (b) Número de nós amostrados por iteração.

Por fim, nas simulações da Fig. 4, considerou-se um cenário com ruído impulsivo. Nesse caso, além do ruído de medição considerado nas simulações da Fig. 2, um ruído com variância $\sigma_{R_{i,k}}^2 = 10^4 \cdot \sigma_{v_k}^2$ ocorre em cada nó k com probabilidade $p_k = 0,1$. Nessa situação, é possível observar uma melhora no desempenho em regime permanente conforme o número de nós amostrados diminui. Uma possível interpretação para isso consiste no fato de que, embora a etapa de adaptação seja crucial para a convergência, ela tende a introduzir ruído no algoritmo, enquanto que a etapa de combinação tende a filtrá-lo [15], [20]. Além disso, é possível observar que o algoritmo AS-dNLMS não deixa de amostrar os nós nesse cenário, ao contrário do DTAS-dNLMS. Isto se deve à adaptação dos valores de β_k , o que possibilita levar em conta o efeito do ruído impulsivo sobre o erro de estimação. Inicialmente, o DTAS-dNLMS apresenta o mesmo comportamento que os algoritmos AS-dNLMS e dNLMS com 20 nós amostrados, mas passa a convergir a um patamar menor de NMSD depois que passa a amostrar menos nós. Cabe notar que, em regime permanente, o algoritmo amostra em média 0,5 nó por iteração.

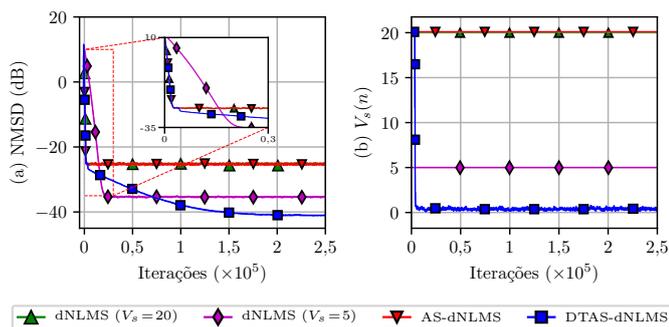


Fig. 4: Comparação entre os algoritmos dNLMS com V_s nós amostrados aleatoriamente ($V_s \in \{5, 20\}$), AS-dNLMS ($\beta = 0,2$, $\Delta_n = 3 \cdot 10^3$) e DTAS-dLMS ($\gamma = 8,5$, $\Delta_n = 3 \cdot 10^3$) em um cenário com ruído impulsivo: (a) curvas de NMSD e (b) Número de nós amostrados por iteração.

V. CONCLUSÕES

Neste trabalho, foi apresentado um algoritmo de amostragem adaptativa para redes de difusão que ajusta seus próprios parâmetros em tempo real. Em contraste com o mecanismo proposto anteriormente, esse algoritmo não requer

o conhecimento da variância do ruído na rede, que pode não estar disponível *a priori*. Além disso, resultados de simulação indicam uma capacidade de rastreamento superior à da versão do algoritmo com parâmetros fixos, já que o ajuste local de cada parâmetro tende a restringir os efeitos de nós ruidosos às suas próprias vizinhanças. Por fim, em cenários com ruído impulsivo, o algoritmo proposto é capaz de amostrar menos nós do que o seu equivalente com parâmetros fixos, levando a um melhor desempenho em regime permanente. Em trabalhos futuros, pretende-se realizar análises teóricas sobre o algoritmo proposto, estudando por exemplo o efeito do parâmetro γ sobre o número esperado de nós amostrados em regime permanente, além de testar o algoritmo em outros cenários de simulação.

REFERÊNCIAS

- [1] A.H. Sayed, *Adaptation, Learning, and Optimization over Networks*, vol. 7, Foundations and Trends in Machine Learning, now Publishers Inc., Hanover, MA, 2014.
- [2] C.G. Lopes and A.H. Sayed, “Diffusion least-mean squares over adaptive networks: Formulation and performance analysis,” *IEEE Trans. Signal Process.*, vol. 56, pp. 3122–3136, Jul. 2008.
- [3] F.S. Cattivelli and A.H. Sayed, “Diffusion LMS strategies for distributed estimation,” *IEEE Trans. Signal Process.*, vol. 58, pp. 1035–1048, Mar. 2009.
- [4] N. Takahashi, I. Yamada, and A.H. Sayed, “Diffusion least-mean squares with adaptive combiners: Formulation and performance analysis,” *IEEE Trans. Signal Process.*, vol. 58, pp. 4795–4810, Sep. 2010.
- [5] C.-K. Yu and A.H. Sayed, “A strategy for adjusting combination weights over adaptive networks,” in *Proc. IEEE ICASSP*, 2013, pp. 4579–4583.
- [6] J. Fernandez-Bes, J. Arenas-García, M.T.M. Silva, and L.A. Azpicueta-Ruiz, “Adaptive diffusion schemes for heterogeneous networks,” *IEEE Trans. Signal Process.*, vol. 65, pp. 5661–5674, Nov. 2017.
- [7] C.G. Lopes and A.H. Sayed, “Diffusion adaptive networks with changing topologies,” in *Proc. IEEE ICASSP*, 2008, pp. 3285–3288.
- [8] X. Zhao and A.H. Sayed, “Single-link diffusion strategies over adaptive networks,” in *Proc. IEEE ICASSP*, 2012, pp. 3749–3752.
- [9] S. Xu, R.C. de Lamare, and H.V. Poor, “Adaptive link selection algorithms for distributed estimation,” *EURASIP J. Adv. in Signal Process.*, vol. 2015, pp. 86, 2015.
- [10] N. Takahashi and I. Yamada, “Link probability control for probabilistic diffusion least-mean squares over resource-constrained networks,” in *Proc. IEEE ICASSP*, 2010, pp. 3518–3521.
- [11] R. Arroyo-Valles, S. Maleki, and G. Leus, “A censoring strategy for decentralized estimation in energy-constrained adaptive diffusion networks,” in *Proc. IEEE SPAWC*, 2013, pp. 155–159.
- [12] J. Fernandez-Bes, R. Arroyo-Valles, J. Arenas-García, and J. Cid-Sueiro, “Censoring diffusion for harvesting WSNs,” in *Proc. IEEE CAMSAP*, 2015, pp. 237–240.
- [13] P. Di Lorenzo, P. Banelli, E. Usufi, S. Barbarossa, and G. Leus, “Adaptive graph signal processing: Algorithms and optimal sampling strategies,” *IEEE Trans. Signal Process.*, vol. 66, pp. 3584–3598, Jul. 2018.
- [14] P. Di Lorenzo, P. Banelli, S. Barbarossa, and S. Sardellitti, “Distributed adaptive learning of graph signals,” *IEEE Trans. Signal Process.*, vol. 65, pp. 4193–4208, Aug. 2017.
- [15] D. G. Tinglea, R. Candido, and M. T. M. Silva, “A low-cost algorithm for adaptive sampling and censoring in diffusion networks,” *IEEE Trans. on Signal Process.*, vol. 69, pp. 58–72, Jan. 2021.
- [16] T. Strutz, “Estimation of measurement-noise variance for variable step-size NLMS filters,” in *Proc. EUSIPCO*, 2019, pp. 1–5.
- [17] S.-Y. Tu and A.H. Sayed, “Optimal combination rules for adaptation and learning over networks,” in *Proc. Int. Workshop on Comput. Adv. in Multi-Sensor Adaptive Process. (CAMSAP)*, 2011, pp. 317–320.
- [18] M. Lázaro-Gredilla, L.A. Azpicueta-Ruiz, A.R. Figueiras-Vidal, and J. Arenas-García, “Adaptively biasing the weights of adaptive filters,” *IEEE Trans. Signal Process.*, vol. 58, pp. 3890–3895, Jul. 2010.
- [19] R. Nassif, C. Richard, J. Chen, and A.H. Sayed, “Distributed diffusion adaptation over graph signals,” in *Proc. IEEE ICASSP*, 2018, pp. 4129–4133.
- [20] J.-W. Lee, J.-T. Kong, W.-J. Song, and S.-E. Kim, “Data-Reserved Periodic Diffusion LMS With Low Communication Cost Over Networks,” *IEEE Access.*, vol. 6, pp. 54636–54650, Sep. 2018.