REDUCING THE COMMUNICATION AND COMPUTATIONAL COST OF RANDOM FOURIER FEATURES KERNEL LMS IN DIFFUSION NETWORKS

Daniel G. Tiglea, Renato Candido*, Luis A. Azpicueta-Ruiz†, and Magno T. M. Silva**

*Universidade de São Paulo, Brazil
{dtiglea, renatocan, magno}@lps.usp.br

†Universidad Carlos III de Madrid, Spain
azpicueta@tsc.uc3m.es

ABSTRACT

Diffusion kernel algorithms are interesting tools for distributed nonlinear estimation. However, for the sake of feasibility, it is essential in practice to restrict their computational cost and the number of communications. In this paper, we propose a censoring algorithm for adaptive kernel diffusion networks based on random Fourier features that locally adapts the number of nodes censored according to the estimation error. It presents fast convergence during the transient phase and a significant reduction in the number of censored nodes in the steady state, thus reducing the energy consumption and the computational cost mainly by decreasing the amount of communication between nodes. Simulation results show that the proposed technique can significantly decrease the computational cost with less impact on the convergence rate when compared to existing solutions.

***Index Terms*—** Diffusion Networks, Kernel Adaptive Filtering, Random Fourier Features, Censoring

1. INTRODUCTION

Adaptive diffusion networks are an effective tool for distributed estimation and signal processing since they present better scalability, autonomy, and flexibility in comparison with centralized approaches [1–4]. They have been employed in many applications, such as target localization and tracking [5], spectrum sensing [6], and internet of things (IoT) [7]. Furthermore, they are considered as effective online solutions in nonlinear estimation and classification tasks [8, 9].

The aim of adaptive diffusion networks is to estimate a parameter of interest in a distributed way by using a set of connected nodes. Each node collects data to compute a local estimate in the *adaptation step*. Then, it communicates with nearby nodes to obtain a global estimate in the *combination step*. The order in which these steps are performed leads to two different schemes: the Adapt-Then-Combine (ATC) and Combine-Then-Adapt (CTA) strategies [1].

In nonlinear estimation, where the data are generated by nonlinear functions that lie in a *Reproducing Kernel Hilbert Space*, the literature contains distributed strategies for kernel adaptive filters [10]. The distributed kernel least mean squares (dKLMS) algorithm is the most popular, mainly due to its simplicity [11, 12]. However, it presents some limitations. First, its computational burden increases linearly with time due to the growth of the dictionary [10], and, last and most critical, the dictionary must be shared with all nodes, which has a dramatic impact on the network traffic [9, 11]. Alternative solutions have been proposed to make distributed kernel algorithms computationally more efficient. A preselected dictionary, fixed and common to all nodes in the network, limits the computational cost but causes a performance degradation [9, 11]. An interesting solution was presented in [8], where kernel adaptive filters based on random Fourier features (RFF) were proposed for distributed nonlinear estimation over networks. Even in this case, however, the computational cost may be prohibitively high when the number of features is large. Therefore, to make RFF kernel diffusion networks feasible for practical applications, a reduction in the amount of data measured, processed, and transmitted by each node is necessary.

Moreover, adaptive diffusion networks suffer from other important drawback, related to the battery lifetime. In particular, the power consumption due to the communication processes poses a challenge. Different strategies have been proposed in the literature to allow the nodes to save energy and therefore extend the lifetime of the diffusion network as a whole (see, e.g., [13–20] and their references). One possible approach to this problem is the adoption of *censoring* techniques, which cut the transmission from certain nodes to any of their neighbors, hence allowing censored nodes to turn their transmitters off. In the context of distributed RFF Kernel Adaptive Algorithms, the censoring mechanism of [16] stands out, since it was proposed with this type of solution in mind. Nonetheless, it requires the calculation of the norm of the difference between the last transmitted local estimate and the current one, which becomes costly when the number of features is large.

Focusing on linear solutions, a censoring algorithm was proposed in [19] to reduce the computational and energy cost of adaptive diffusion networks. The premise of this approach is to locally adapt the number of nodes censored according to the estimation error. During the transient phase, it maintains the nodes uncensored, which enables fast convergence. However, when the error is low in magnitude, which generally occurs in steady state, it censors a large number of nodes and, therefore, the communication cost associated with the learning task is reduced. Furthermore, this algorithm was designed so that censored nodes also avoid doing most of the computation that they need to do when they are uncensored.

Since both the computational cost and the energy consumption constraints are prominent issues in the case of diffusion kernel adaptive filters, in this paper, we propose to extend the censoring algorithm of [19] to the RFF-KLMS algorithm of [8] to address simultaneously both concerns. Simulations show that the proposed technique can reduce the number of communications as much as the censoring solution of [16], while significantly decreasing the computational burden and having less impact on the convergence rate.

Organization of the paper and Notation. In Section 2, the RFF-KLMS algorithm is revisited, but considering its normalized version. We describe our proposal to reduce the cost of such algorithm in Section 3, followed by some simulation results and conclusions, in Sections 4 and 5, respectively. We use normal fonts for scalars and bold-

This work was supported by CAPES under Grant 88887.512247/2020-00 and Finance Code 001, by the São Paulo Research Foundation (FAPESP) under Grant 2021/02063-6, and by CNPq under Grant 303826/2022-3. The work of Azpicueta-Ruiz has been partially supported by the Spanish Ministry of Science and Innovation through Grant No. PID2021-124280OB-C21 (MCIU/AEI/FEDER, UE).

face letters for vectors. Moreover, $(\cdot)^{\mathrm{T}}$ denotes transposition, $(\cdot)^*$ the complex conjugate, $\mathrm{E}\{\cdot\}$ the mathematical expectation, $\exp(\cdot)$ the exponential function, $|\cdot|$ the cardinality of a set, and $\|\cdot\|$ the Euclidean norm.

2. RFF-KNLMS DIFFUSION NETWORK

In a network with V nodes connected according to a predefined topology, two nodes are considered neighbors if they are directly connected. The neighborhood of node k, including k itself, is denoted by $\mathcal{N}_k$. Each node k has access to an input signal $u_k(n)$ and to a desired signal $d_k(n) = f_{\mathrm{o}}[\mathbf{u}_k(n)] + v_k$, where $\mathbf{u}_k(n) \in \mathbb{R}^M$ is a column regressor vector, $f_{\mathrm{o}}[\cdot]$ is an unknown nonlinear function to be indentified, and $v_k(n)$ is the measurement noise, which is assumed to be independent of the other variables and zero-mean with variance $\sigma^2_{v_k}$.

To estimate $d_k(n)$ at node k, the input vector $\mathbf{u}_k(n)$ can be mapped into a high-dimensional feature space $\mathcal{H}$ as $\boldsymbol{\varphi}[\mathbf{u}_k(n)]$, using a Mercer's kernel $\kappa : \mathbb{R}^M \times \mathbb{R}^M \rightarrow \mathbb{R}$, such that the kernel trick holds [10,21], i.e., $\kappa(\mathbf{u}, \mathbf{u}') = \boldsymbol{\varphi}^{\mathrm{T}}[\mathbf{u}]\boldsymbol{\varphi}[\mathbf{u}']$. One of the most widely utilized kernels in the literature is the Gaussian kernel, which induces an infinite-dimensional Hilbert space and is defined as $\kappa(\mathbf{u}, \mathbf{u}') = \exp\big(-\|\mathbf{u} - \mathbf{u}'\|^2/(2\sigma^2)\big)$, where $\sigma > 0$ is the kernel width [10].

Instead of using the kernel trick, $\mathbf{u}_k(n)$ can be mapped to a finite-dimensional Euclidean space by using an RFF map $\mathbf{z} : \mathbb{R}^M \rightarrow \mathbb{R}^D$, based on Bochner's theorem. This theorem ensures that, if the kernel is shift invariant and positive definite, which is the case of the Gaussian kernel, the Fourier transform $p(\boldsymbol{\omega})$ of the kernel is a probability density function such that [8,22]

$$\kappa(\mathbf{u}, \mathbf{u}') = \int p(\boldsymbol{\omega}) e^{j\boldsymbol{\omega}^{\mathrm{T}}(\mathbf{u}-\mathbf{u}')} d\boldsymbol{\omega}. \tag{1}$$

Since $p(\boldsymbol{\omega})$ and $\kappa(\mathbf{u}, \mathbf{u}')$ are real, the integral (1) converges when the complex exponentials are replaced by cosines. Thus, the real-valued mapping $z_{\boldsymbol{\omega},b}[\mathbf{u}] \triangleq \sqrt{2}\cos(\boldsymbol{\omega}^{\mathrm{T}}\mathbf{u}+b)$ also satisfies (1) if $\boldsymbol{\omega}$ is drawn from $p(\boldsymbol{\omega})$ and b from the uniform distribution on $[0, 2\pi]$ [8]. Therefore, $\kappa(\mathbf{u}, \mathbf{u}')$ can be computed as $\mathrm{E}\{z_{\boldsymbol{\omega},b}[\mathbf{u}] z^*_{\boldsymbol{\omega},b}[\mathbf{u}']\}$. To reduce the variance of this estimate, a sample average of D randomly chosen $z_{\boldsymbol{\omega},b}[\cdot]$ is used, i.e.,

$$\kappa(\mathbf{u}, \mathbf{u}') \approx \frac{1}{D} \sum_{i=1}^{D} z_{\boldsymbol{\omega}_i,b_i}[\mathbf{u}] z_{\boldsymbol{\omega}_i,b_i}[\mathbf{u}'].$$

As a consequence, the vector $\mathbf{u}_k(n)$ can be mapped to the following D-dimensional RFF vector [8]:

$$\mathbf{z}_k(n) \triangleq \sqrt{\frac{2}{D}} \begin{bmatrix} \cos(\boldsymbol{\omega}_1^{\mathrm{T}}\mathbf{u}_k(n) + b_1) \\ \vdots \\ \cos(\boldsymbol{\omega}_D^{\mathrm{T}}\mathbf{u}_k(n) + b_D) \end{bmatrix}. \tag{2}$$

For the Gaussian kernel, $\boldsymbol{\omega}_i$, $i = 1, \cdots, D$ are drawn from the multivariate Gaussian distribution with zero mean and covariance matrix $\mathbf{I}_D/\sigma^2$, where $\mathbf{I}_D$ is the identity matrix with dimensions $D \times D$ [8,22,23].

Since the RFF space has finite dimension, it is possible to estimate $d_k(n)$ at node k by directly using a similar strategy to that of the linear ATC distributed normalized LMS (dNLMS) algorithm [1]. Thus, the ATC RFF-dKNLMS consists in two steps given by [9]

$$\begin{cases} \boldsymbol{\theta}_k(n+1) = \boldsymbol{\psi}_k(n) + \mu_k(n) e_k(n) \mathbf{z}_k(n) & \text{(3a)} \\ \boldsymbol{\psi}_k(n+1) = \sum_{j \in \mathcal{N}_k} c_{jk} \boldsymbol{\theta}_j(n+1), & \text{(3b)} \end{cases}$$

where $\boldsymbol{\theta}_k$ and $\boldsymbol{\psi}_k$ represent the local and combined D-length weight vectors at node k, respectively, the signal $e_k(n) = d_k(n) - \mathbf{z}_k^{\mathrm{T}}(n)\boldsymbol{\psi}_k(n)$ is the estimation error, and $\mu_k(n)$ is a normalized step size, i.e., $\mu_k(n) = \tilde{\mu}_k/[\delta + \|\mathbf{z}_k(n)\|^2]$ with $0 < \tilde{\mu}_k < 2$ and a small regularization factor $\delta > 0$ [1]. Lastly, $\{c_{jk}\}$ are combination weights satisfying $c_{jk} \geqslant 0$, $\sum_{j \in \mathcal{N}_k} c_{jk} = 1$, and $c_{jk} = 0$ if $j \notin \mathcal{N}_k$. Possible choices for $\{c_{jk}\}$ include fixed or adaptive rules [1]. In particular, the Metropolis rule, which is the one used throughout this paper, sets the combination weights as [1]

$$c_{jk} = \begin{cases} 1/\max\{|\mathcal{N}_k|, |\mathcal{N}_j|\}, & \text{if } j \in \mathcal{N}_k \text{ and} j \neq k \\ 0, & \text{if } j \notin \mathcal{N}_k \\ 1 - \sum_{\substack{j \in \mathcal{N}_k \\ j \neq k}} c_{jk}, & \text{if } j = k. \end{cases}$$

It is important to highlight that, since the nodes share their local estimates, the random features must be common to all of them. Finally, for the sake of simplicity, in this paper we consider only the ATC strategy, but the results can be straightforwardly extended to the CTA one as well.

3. REDUCED-COST RFF-DKNLMS

The censoring mechanism proposed in [19] skips the adaptation step at node k when this node is censored. As a consequence, in the proposed scheme, (3a) is replaced by

$$\begin{aligned} \boldsymbol{\theta}_k(n+1) = &[1 - \bar{s}_k(n)]\boldsymbol{\theta}_k(n) + \\ &\bar{s}_k(n)\big[\boldsymbol{\psi}_k(n) + \mu_k(n) e_k(n) \mathbf{z}_k(n)\big], \end{aligned} \tag{4}$$

where $\bar{s}_k(n) \in \{0, 1\}$ is an adaptive variable to control the censoring, i.e., such that $\bar{s}_k(n) = 0$ when node k is censored and $\bar{s}_k(n) = 1$ otherwise. Thus, when $\bar{s}_k(n) = 0$, the local estimate $\boldsymbol{\theta}_k$ remains unchanged. Assuming that the nodes can store the local estimates sent by their neighbors at past iterations, this means that there is no need for node k to broadcast $\boldsymbol{\theta}_k$ again, and thus it can shut its transmitter off. Furthermore, since $\boldsymbol{\theta}_k$ is not updated, $e_k(n)$ and $\mu_k(n)$ do not have to be calculated, and the neighborhood of node k does not need to compute $c_{ik}\boldsymbol{\theta}_k$ in the combination step again. Finally, if the goal of the algorithm is solely to perform nonlinear system identification, $\mathbf{z}_k^{\mathrm{T}}(n)$ and $\mathbf{z}_k^{\mathrm{T}}(n)\boldsymbol{\psi}_k(n)$ do not have to be calculated either when $\bar{s}_k(n) = 0$.

Similarly to what was done in [19,20], rather than directly adapting $\bar{s}_k(n)$, we introduce an auxiliary parameter $\alpha_k(n) \in [-\alpha^+, \alpha^+]$ such that

$$\bar{s}_k(n) = \begin{cases} 1, \text{ if } \alpha_k(n) \geqslant 0, \\ 0, \text{ otherwise.} \end{cases} \tag{5}$$

The following cost function [19] is then introduced as a key component of our algorithm:

$$J_{\alpha_k}(n) = \phi_k(n)\beta\bar{s}_k(n) + \big[1 - \phi_k(n)\big] \textstyle\sum_{j \in \mathcal{N}_k} c_{jk} e_j^2(n), \tag{6}$$

where $\beta > 0$ is a parameter that controls how much the communication between nodes is penalized and

$$\phi_k(n) \triangleq \phi[\alpha_k(n)] = \frac{\mathrm{sgm}[\alpha_k(n)] - \mathrm{sgm}[-\alpha^+]}{\mathrm{sgm}[\alpha^+] - \mathrm{sgm}[-\alpha^+]}, \tag{7}$$

is a scaled and shifted sigmoid function [19,20,24], with $\mathrm{sgm}[x] = [1 + \exp(-x)]^{-1}$ and $\alpha^+ = 4$ being a common choice in the literature [19,20,24].

The reasoning behind (6) is that, when the error magnitude is high (such as in the transient phase), $J_{\alpha_k}(n)$ is minimized by making $\phi_k(n)$ close to one, keeping node k uncensored. A similar line of thought applies when node k is censored ($\bar{s}_k(n)=0$), thus ensuring that the censoring of node k will end eventually. However, when node k is not censored ($\bar{s}_k(n)=1$) and the error magnitude is sufficiently small, the cost function of (6) is minimized by making $\phi_k(n)$ close to zero. Hence, the algorithm begins to censor node k [19].

The adaptive censoring mechanism is then obtained by taking the derivative of $J_{\alpha_k}(n)$ in (6) with respect to $\alpha_k(n)$. Since $e_j(n)$ is not known when node j is censored, we replace $e_j(n)$ by its latest measurement $\varepsilon_j(n)$, given by $\varepsilon_j(n) = [1-\bar{s}_j(n)]\varepsilon_j(n-1) + \bar{s}_j(n)e_j(n)$. We thus get the following stochastic gradient descent rule [19]:

$$\alpha_k(n+1)=\alpha_k(n)+\mu_s\phi'_k(n)\big[\textstyle\sum_{j\in\mathcal{N}_k} c_{jk}\varepsilon_j^2(n)-\beta\bar{s}_k(n)\big], \quad (8)$$

where $\mu_s > 0$ is a step size and

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

Assuming that the gradient noise can be neglected, it can be shown that node k will be censored at some point if $\beta > \sum_{j\in\mathcal{N}_k} c_{jk}\sigma_{v_j}^2$ [19]. To eliminate the need for *a priori* knowledge of the noise variance, in [20] the parameter β was replaced by a local, time-varying parameter

$$\beta_k(n) = \gamma\widehat{\sigma}^2_{\mathcal{N}_k}(n) = \gamma\textstyle\sum_{j\in\mathcal{N}_k} c_{jk}\widehat{\sigma}^2_{v_j}(n) \quad (10)$$

for each node k, where $\gamma > 1$ is a parameter that the filter designer must choose, $\widehat{\sigma}^2_{\mathcal{N}_k} \triangleq \sum_{j\in\mathcal{N}_k} c_{jk}\widehat{\sigma}^2_{v_j}(n)$, and $\widehat{\sigma}^2_{v_j}(n)$ is an estimate of $\sigma^2_{v_j}$. The greater the value of γ, the more nodes are censored. In this paper, we use the algorithm proposed in [25] to calculate $\widehat{\sigma}^2_{v_k}(n)$ at each node k in an online manner, as suggested in [20]. Furthermore, following the same procedure employed in [20], we also adopt a local and time-varying step size given by

$$\mu_{s_k}(n)=\frac{1}{\delta+\widehat{\sigma}^2_{\mathcal{N}_k}(n)}\left\{\frac{\alpha^+}{(\gamma-1)(\phi'_0-\phi'_{\alpha^+})}\left[\left(\frac{\phi'_o}{\phi'_{\alpha^+}}\right)^{\frac{1}{\Delta n}}-1\right]\right\}, \quad (11)$$

where ϕ'_0 and ϕ'_{α^+} denote ϕ' evaluated at $\alpha_k = 0$ and $\alpha_k = \alpha^+$, respectively, and Δn is the maximum number of iterations between the beginning of the steady state in terms of the mean-squared error (MSE) and the beginning of the censorship of the nodes. We remark that the term between curly braces is a constant. Thus, (11) only requires one extra multiplication, sum, and division per iteration.

Eqs. (3b), (4), (5), and (8) through (11) are the main equations of the proposed algorithm, which we name Adaptive Censoring (AC) RFF-dKNLMS. Finally, it should be remarked that AC-RFF-dKNLMS requires that each uncensored node k transmits $e_k^2(n)$ and $\widehat{\sigma}^2_{v_k}(n)$ to its neighbors. Nonetheless, these data can be sent bundled with $\boldsymbol{\theta}_k(n+1)$ to keep the number of transmissions unchanged. Moreover, this slight increase in overhead results negligible comparing with the reduction in transmissions caused by the number of times each node is censored. This fact will become clear in the next section.

Computational cost. We focus our analysis on the number of multiplications, since they are typically more demanding than additions from a computational perspective. The RFF-dKLNMS algorithm requires approximately $D(M+3+|\mathcal{N}_k|)+1$ multiplications per iteration. In contrast, when node k is censored in the AC-RFF-dKNLMS algorithm, it only needs to update $\alpha_k(n+1)$ in (8) and $\boldsymbol{\psi}_k(n+1)$ in (3b) using the estimates of its uncensored neighbors. Hence, in this situation, it must carry out $2+\sum_{j\in\mathcal{N}_k}\bar{s}_j(n)(D+1)$ multiplications. In this case, (10) does not have to be calculated since $\bar{s}_k(n) = 0$. On the other hand, when node k is not censored, Eqs. (8) through (11) (including the noise power estimation algorithm of [25]) demand approximately $2|\mathcal{N}_k|+11$ more multiplications in comparison with the standard RFF-dKLNMS algorithm. Thus, we can see that the computational cost of AC-RFF-dKNLMS is slightly higher when the nodes are not censored (e.g., during the transient phase), but decreases significantly when they are censored (e.g., in steady state). This situation is summarized in Table 1, where we assume that $\phi'_k(n)$ is implemented by a lookup table, and do not take this operation into account in our analysis.

4. SIMULATION RESULTS

In this section, we compare the performance of our AC-RFF-dKNLMS algorithm to that of other state-of-the-art solutions considering two simulation scenarios (Secs. 4.1 and 4.2), and using the network with $V = 20$ nodes depicted in Fig. 1(a). The noise power $\sigma^2_{v_k}$ is shown in Fig. 1(b) for $k = 1, \cdots, V$, and the input signal $u_k(n)$ is a white Gaussian noise with zero mean and unit variance. The results were obtained over an average of 500 realizations.

Besides the proposed AC-RFF-dKNLMS scheme, we also consider in the simulations the RFF-dKNLMS algorithm with two different censoring approaches: (i) the censoring mechanism of [16], and (ii) a random censoring policy, in which V_s nodes are selected randomly at each iteration to remain uncensored. To enable the comparison between AC-RFF-dKNLMS and the censoring strategy of [16], their parameters γ and τ_n were selected to obtain roughly one node uncensored per iteration in steady state in each case. In the random censoring technique, censored nodes are prevented from updating their local estimates, in a similar fashion to AC-RFF-dKNLMS. In addition, and for the sake of comparison, we also show results obtained with three other approaches: the RFF-dKNLMS algorithm with all nodes uncensored, the RFF-dKNLMS scheme with a non-cooperative communication policy (i.e., $c_{jk} = 0$ if $j \neq k$ and $c_{jk} = 1$ if $j = k$), and the linear dNLMS with all nodes uncensored. In all cases, we adopt $\mu_k = 1$ for every node k and $\delta = 10^{-5}$.

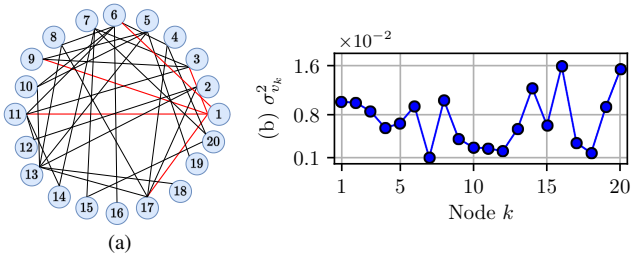

Fig. 1: (a) Network topology, with the connections between node #1 and its neighbors highlighted, and (b) $\sigma^2_{v_k}$ used in the experiments.

4.1. Toy Example

Firstly, we consider a toy example in which $d_k(n) = \mathbf{w}_o\mathbf{z}_k(n) + v_k(n)$, where $\mathbf{w}_o$ is a typically unknown system, oftentimes referred to as the "optimal system" in the adaptive filtering literature [1]. In our simulations, we randomly generated the $D = 50$ coefficients of $\mathbf{w}_o$ following a uniform distribution in $[-1, 1]$. At each iteration,

Table 1: Comparison between RFF-dKNLMS and AC-RFF-dKNLMS: number of multiplications per iteration for each node k.

Algorithm	Multiplications ($\otimes$)
RFF-dKNLMS	$D(M+3+\lvert\mathcal{N}_k\rvert)+1$
AC-RFF-dKNLMS	$\bar{s}_k(n)\times[D(M+3+\lvert\mathcal{N}_k\rvert)+2\lvert\mathcal{N}_k\rvert+12]+[1-\bar{s}_k(n)][2+\sum_{j\in\mathcal{N}_k}\bar{s}_j(n)(D+1)]$

$\mathbf{z}_k(n)$ is generated using the past $M=2$ samples of $u_k(n)$ and RFFs drawn from a multivariate Gaussian distribution with $\sigma^2=10^{-2}$.

We assume that the RFFs utilized to generate $\mathbf{z}_k(n)$ are the same employed by the RFF-dKNLMS algorithm. Although not realistic, this assumption allows us to adopt the network mean-square deviation (NMSD) as a performance indicator, which is given by $\text{NMSD}(n) = \frac{1}{V}\sum_{k=1}^{V}\text{E}\{\|\mathbf{w}_o-\boldsymbol{\psi}_k(n)\|^2\}$. Specifically for the dNLMS algorithm, we adopt $M = D = 50$ to enable the comparison between $\mathbf{w}_o$ and the coefficients of the adaptive filter. In addition, we set $\gamma = 10$ for our AC-RFF-dKNLMS scheme and, to preserve its convergence rate, $\Delta n = 5000$. For the censoring mechanism of [16], named COKE by the authors, we adopted $\tau_n = 0.05$.

In Fig. 2, we show (a) the NMSD curves, (b) the number of uncensored nodes, and (c) the number of multiplications per iteration. We can see that AC-RFF-dKNLMS keeps the nodes uncensored during the transient phase [Fig. 2(b)] and thus converges as fast as RFF-dKNLMS with $V_s = 20$ nodes uncensored, which can be considered as a lower bound regarding the NMSD [Fig. 2(a)]. In contrast, the RFF-dKNLMS with $V_s = 2$ nodes randomly uncensored clearly presents a slower convergence rate. The algorithm of [16] outperforms the random censoring technique while censoring more nodes, but converges slightly slower than AC-RFF-dKNLMS and the RFF-dKNLMS algorithm with $V_s = 20$ nodes uncensored. In terms of the computational cost, the AC-RFF-dKNLMS algorithm demands less multiplications per iteration in steady state than any other solution. On average, AC-RFF-dKNLMS reduces the burden by roughly 95% in comparison with the RFF-dKNLMS algorithm with $V_s = 20$ nodes uncensored and 90% in comparison with the solution of [16] in steady state. This clearly compensates the slight increment in computational burden during the transient phase.

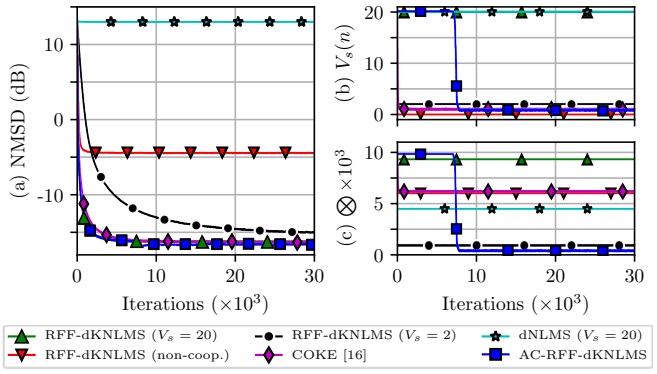

Fig. 2: Simulation results obtained in the toy example scenario. (a) NMSD curves, (b) number of uncensored nodes, and (c) multiplications per iteration.

4.2. Nonlinear Channel Identification

In this subsection, we consider that $d_k(n) = y_k(n)+0.2y_k^2(n)-0.1y_k^3(n)+v_k(n)$, where $y_k(n)$ is the output of a linear channel described by $H(z) = 0.3482+0.8704z^{-1}+0.3482z^{-2}$ with input $u_k(n)$. In this case, we consider $M = 3$ and $D = 500$ features generated with $\sigma^2 = 1$. Moreover, we adopt the network MSE (NMSE) as a performance indicator, which is given by $\text{NMSE}(n) = \frac{1}{V}\sum_{k=1}^{V}\text{E}\{e_k^2(n)\}$, and shown in Fig.3(a) for all the algorithms. We selected $\gamma = 10$ for the AC-RFF-dKNLMS algorithm and set $\Delta n = 10^4$ in order to preserve its convergence rate. For the solution of [16], we considered $\tau_n = 10^{-3}$.

We observe from Fig. 3 that, similarly to what was seen in Fig. 2, AC-RFF-dKNLMS keeps the nodes uncensored during the transient phase and thus preserves the convergence rate of RFF-dKNLMS with $V_s = 20$ nodes uncensored. In terms of the computational cost, the AC-RFF-dKNLMS algorithm carries out less multiplications per iteration in steady state in comparison with any other solution, except for the linear dNLMS algorithm, whose cost was much lower in this scenario due to the adoption of $M = 3$. Once again, we can see that AC-RFF-dKNLMS outperforms the censoring algorithm of [16] in terms of the convergence rate and computational cost.

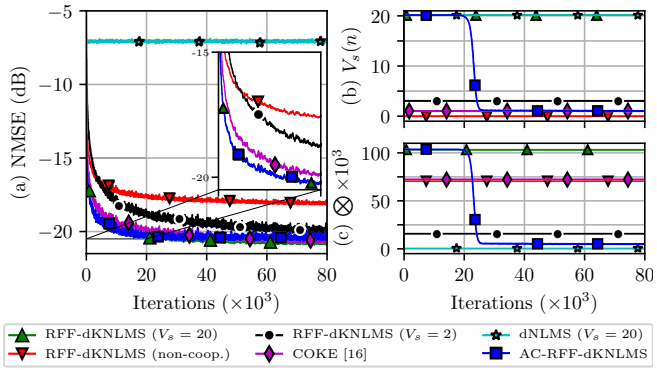

Fig. 3: Simulation results obtained for nonlinear channel identification. (a) NMSE curves, (b) number of uncensored nodes, and (c) multiplications per iteration. For better visualization, we applied a moving-average filter with 64 coefficients to the curves.

5. CONCLUSION

In this paper, we extended the censoring mechanism proposed in [19] and [20] to distributed kernel adaptive filters. The resulting algorithm, named as AC-RFF-dKNLMS, maintains the nodes uncensored when the error is high in magnitude, and censors them otherwise. Since the adaptive censoring mechanism prevents the nodes from doing most of the calculations that they have to carry out when they are not censored, the resulting algorithm also reduces the computational cost drastically, which is of essence when dealing with practical implementations of RFF Kernel algorithms. Simulation results showed that AC-RFF-dKNLMS can outperform other state-of-the-art censoring algorithms in the transient phase, while censoring as many nodes as them in steady state. Furthermore, its computational complexity was shown to be comparatively much lower in steady state, which makes it a very attractive solution. For future work, we intend to test the proposed algorithm in an application using real-world data, compare it with other censoring techniques extended to kernel diffusion networks, and obtain theoretical results for it, such as analytical expressions for the number of nodes censored per iteration.

6. REFERENCES

[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 Transactions on Signal Processing*, vol. 56, pp. 3122 –3136, Jul. 2008.

[3] M. S. E. Abadi and M. S. Shafiee, "Distributed estimation over an adaptive diffusion network based on the family of affine projection algorithms," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 5, no. 2, pp. 234–247, 2019.

[4] S. Chouvardas, K. Slavakis, and S. Theodoridis, "Adaptive robust distributed learning in diffusion sensor networks," *IEEE Transactions on Signal Processing*, vol. 59, no. 10, pp. 4692–4707, Oct 2011.

[5] J. Chen, C. Richard, and A. H. Sayed, "Multitask diffusion adaptation over networks," *IEEE Transactions on Signal Processing*, vol. 62, no. 16, pp. 4129-4144, Aug. 2014.

[6] P. Di Lorenzo, S. Barbarossa, and A. H. Sayed, "Distributed spectrum estimation for small cell networks based on sparse diffusion adaptation," *IEEE Signal Processing Letters*, vol. 20, no. 12, pp. 1261–1265, 2013.

[7] F. Chen, L. Hu, P. Liu, and M. Feng, "A robust diffusion estimation algorithm for asynchronous networks in IoT," *IEEE Internet of Things Journal*, vol. 7, no. 9, pp. 9103–9115, 2020.

[8] P. Bouboulis, S. Chouvardas, and S. Theodoridis, "Online distributed learning over networks in RKH spaces using random fourier features," *IEEE Transactions on Signal Processing*, vol. 66, no. 7, pp. 1920–1932, 2018.

[9] S. Scardapane, J. Chen, and C. Richard, "Combined filtering architectures for complex nonlinear systems," in *Adaptation and learning over networks for nonlinear system modelling*, D. Comminiello and J. Principe, Eds., chapter 10, pp. 223–242. Butterworth-Heinemann/Elsevier, Oxford, 2018.

[10] W. Liu, J. C. Príncipe, and S. Haykin, *Kernel adaptive filtering: a comprehensive introduction*, Wiley, Hoboken, 2010.

[11] W. Gao, J. Chen, C. Richard, and J. Huang, "Diffusion adaptation over networks with kernel least-mean-square," in *2015 IEEE 6th International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*, 2015, pp. 217–220.

[12] B.-S. Shin, H. Paul, and A. Dekorsy, "Distributed kernel least squares for nonlinear regression applied to sensor networks," in *2016 24th European Signal Processing Conference (EUSIPCO)*, 2016, pp. 1588–1592.

[13] R. Arablouei, S. Werner, Y.-F. Huang, and K. Dogançay, "Distributed least mean-square estimation with partial diffusion," *IEEE Transactions on Signal Processing*, vol. 62, no. 2, pp. 472–484, 2014.

[14] S. Chouvardas, K. Slavakis, and S. Theodoridis, "Trading off complexity with communication costs in distributed adaptive learning via krylov subspaces for dimensionality reduction," *IEEE Journal of Selected Topics in Signal Processing*, vol. 7, no. 2, pp. 257–273, 2013.

[15] R. Arroyo-Valles, S. Maleki, and G. Leus, "A censoring strategy for decentralized estimation in energy-constrained adaptive diffusion networks," in *Proc. of 14th IEEE Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, 2013, pp. 155–159.

[16] P. Xu, Z. Tian, Z. Zhang, and Y. Wang, "Coke: Communication-censored kernel learning via random features," in *2019 IEEE Data Science Workshop (DSW)*. IEEE, 2019, pp. 32–36.

[17] I. E. K. Harrane, R. Flamary, and C. Richard, "On reducing the communication cost of the diffusion LMS algorithm," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 5, no. 1, pp. 100–112, 2019.

[18] A. Rastegarnia, "Reduced-communication diffusion RLS for distributed estimation over multi-agent networks," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 67, no. 1, pp. 177–181, 2020.

[19] D. G. Tiglea, R. Candido, and M. T. M. Silva, "A low-cost algorithm for adaptive sampling and censoring in diffusion networks," *IEEE Transactions on Signal Processing*, vol. 69, pp. 58–72, Jan. 2021.

[20] D. G. Tiglea, R. Candido, and M. T. M. Silva, "An adaptive algorithm for sampling over diffusion networks with dynamic parameter tuning and change detection mechanisms," *Digital Signal Processing*, vol. 127, pp. 103587, 2022.

[21] C. Richard, J. C. M. Bermudez, and P. Honeine, "Online prediction of time series data with kernels," *IEEE Transactions on Signal Processing*, vol. 57, no. 3, pp. 1058–1067, Mar. 2009.

[22] P. Bouboulis, S. Pougkakiotis, and S. Theodoridis, "Efficient KLMS and KRLS algorithms: A random fourier feature perspective," in *IEEE Statistical Signal Processing Workshop (SSP)*, June 2016, pp. 1–5.

[23] D. Cuevas and I. Santamaría, "Multi-output kernel adaptive filtering with reduced complexity," in *Proc. IEEE Statistical Signal Processing Workshop (SSP)*, 2021, pp. 306–310.

[24] M. Lázaro-Gredilla, L. A. Azpicueta-Ruiz, A. R. Figueiras-Vidal, and J. Arenas-Garcia, "Adaptively biasing the weights of adaptive filters," *IEEE Transactions on Signal Processing*, vol. 58, no. 7, pp. 3890–3895, Jul. 2010.

[25] T. Strutz, "Estimation of measurement-noise variance for variable-step-size NLMS filters," in *Proc. of European Signal Processing Conference (EUSIPCO)*, 2019.