# Graph Kernel Recursive Least-Squares Algorithms

Vinay Chakravarthi Gogineni*, Valeriya Naumova*, Stefan Werner†, Yih-Fang Huang‡
* SimulaMet-Simula Metropolitan Center for Digital Engineering, Oslo, Norway
† Norwegian University of Science and Technology, Trondheim, Norway
‡ University of Notre Dame, Notre Dame, IN, USA

***Abstract*—This paper presents graph kernel adaptive filters that model nonlinear input-output relationships of streaming graph signals. To this end, we propose centralized and distributed graph kernel recursive least-squares (GKRLS) algorithms utilizing the random Fourier features (RFF) map. Compared with solutions based on the traditional kernel trick, the proposed RFF approach presents two significant advantages. First, it sidesteps the need to maintain a high-dimensional dictionary, whose dimension increases with the number of graph nodes and time, which renders prohibitive computational and storage costs, especially when considering least-squares algorithms involving matrix inverses. Second, the distributed algorithm developed in this paper, referred to here as the graph diffusion kernel recursive least-squares (GDKRLS) algorithm, does not require centralized dictionary training, making it ideal for distributed learning in dynamic environments. To examine the performance of the proposed algorithms, we analyze the mean convergence of the GDKRLS algorithm and conduct numerical experiments. Results confirm the superiority of the proposed RFF-based GKRLS and GDKRLS over their LMS counterparts.**

## I. Introduction

Data collected from large-scale interactive systems such as social, financial, and biological networks exhibit certain intrinsic relations with the network structure. For example, in the case of social network data, the network carries the relationships between users' data entities. The network data and their interdependencies are naturally represented through graphs. Recently, graph signal processing (GSP) was developed in an effort to model and analyze data emerging from graphs [1]–[6]. GSP enables data to be modeled as graph signals defined on a set of nodes of a graph, providing a natural way of accounting for both data (i.e., node attributes) and underlying geometry (i.e., edge attributes). In GSP, the inherent network structure and geometry of the data are exploited by redefining classical signal processing techniques such as sampling [7], convolution [8], and filtering [9] for graph signals.

To deal with graph signals, GSP mainly relies on two fundamental building blocks. The graph shift operator [10] captures the interactions between graph nodes, while the graph Fourier transform (GFT) [11] represents graph signals in the spectral domain of the graph. Equipped with these building blocks, one can construct and analyze graph filters that play a central role in modeling unknown relationships between graph input-output signals. In particular, graph filters are normally constructed as rational functions of the underlying graph shift operator. As with classical signal processing, graph filters may take two distinct forms: finite impulse response (FIR) [9], and infinite impulse response (IIR) [12], [13]. Graph filters of order $L$ are localized on the graph, i.e., only $L$-hop adjacent nodes need to communicate the information for the filter implementation. This attractive feature of graph filters motivated their distributed implementation [14], [15].

Recently, several works have extended the concepts of adaptive algorithms for the estimation of graph signals [16], [17]. In an extension to these works, the concepts of distributed adaptive strategies [18] and GSP were merged to derive graph diffusion least mean square (GDLMS) strategies [19]. Since the graph shift matrix is not orthonormal in general, the graph shift operation does not preserve energy, slowing down the convergence of GDLMS. Graph diffusion preconditioned LMS [20] has been proposed as a solution to this problem. All these works assume a linear relationship between graph input and output signals. However, these linear relationships fail in modeling many real-life systems that have complex input-output relationships. Well-known relationships of this kind include air pressure and temperature [21], and wind speed and turbine output [22]. Adaptive filtering in reproducing kernel Hilbert spaces (RKHS) [23], [24] has been found to be a useful method of modeling these nonlinear relationships. Kernel methods are also proven to be effective in graph setting [25], [26]. Recently, building on the ideas of kernel adaptive filters, graph kernel LMS (GKLMS) and graph diffusion kernel LMS (GDKLMS) algorithms were proposed to capture the nonlinear input-output relationships of graph signals [27], [28]. Like GDLMS, GDKLMS also suffers from slow convergence owing to the large eigenvalue spread of the graph-shifted signals.

In this paper, we develop graph kernel filters based on recursive least-squares (RLS) that are robust to the eigenvalue spread of the correlation matrix of the graph-shifted signal. To this end, we first derive the centralized graph kernel recursive least-squares (GKRLS) algorithm for modeling nonlinear relationships between graph input and output signals. To deal with the growing dimensionality problem that arises in GKLMS, we develop the centralized GKRLS using the concepts of random Fourier features (RFF) [29]. Then, we exploit the graph structure and propose a decentralized solution, namely, graph diffusion kernel RLS (GDKRLS). Our analytical studies show the proposed GDKRLS converges in the mean to an unbiased solution. Finally, we demonstrate the performance of the proposed algorithms through numerical examples.

***Mathematical Notations:*** We use lowercase, bold lowercase and bold uppercase letters to refer to scalars, column vectors and matrices, respectively. An identity matrix of appropriate size is denoted by $\mathbf{I}$. Inverse of the matrix is $(\cdot)^{-1}$, while $(\cdot)^{\mathrm{T}}$ is the transpose. Lastly, $\otimes$, col$(\cdot)$ and blockdiag$(\cdot)$ are the right Kronecker product operator, columnwise stacking operator and block diagonal operator, respectively.

## II. Problem Formulation

Consider a weighted undirected graph $\mathcal{G} = \{\mathcal{N}, \mathcal{E}, \mathbf{W}\}$ with vertex set $\mathcal{N}$ of cardinality $K$ and edge set $\mathcal{E}$ and symmetric weighted adjacency $\mathbf{W} \in \mathbb{R}^{K\times K}$ defining the relations between nodes. The $(k,l)$th entry of the weighted adjacency $w_{kl}$, is the weight associated to an edge/relationship between vertices $k$ and $l$. The corresponding graph Laplacian, $\mathbf{L} = \mathbf{D} - \mathbf{W}$, is a real, symmetric, and positive semidefinite matrix, where $\mathbf{D}$ is a diagonal matrix with $[\mathbf{D}]_{k,k} = |\mathcal{N}_k|$ being the cardinality of the neighborhood of node $k$. At time index $n$, the graph signal is defined as a function $\mathbf{u}_n = [u_{1,n}\, u_{2,n}\, \ldots\, u_{K,n}]^{\mathrm{T}} : \mathcal{N} \to \mathbb{R}$, which maps each node attribute to a scalar value. Moreover, the graph is equipped with the graph shift operator $\mathbf{S} \in \mathbb{R}^{K\times K}$ whose entries $[\mathbf{S}]_{k,l} = s_{kl}$ take nonzero values if and only if $(k,l) \in \mathcal{E}$. Matrices that can model the graph topology such as the graph adjacency, the graph Laplacian and their normalized counterparts, are the usual choices for $\mathbf{S}$ [10]. The graph shift operation at node $k$ is defined as a linear combination of local information from immediate neighboring nodes, i.e., $\sum_{l\in\mathcal{N}_k} s_{kl}u_{l,n}$.

Then, a linear shift-invariant (LSI) FIR graph filter, a polynomial of the graph shift operator, combines graph-shifted signals and provides an output, $\mathbf{y}_n = [y_{1,n}, y_{2,n}, \ldots, y_{K,n}]^{\mathrm{T}}$, given by

$$\mathbf{y}_n = \sum_{i=0}^{L-1} \gamma_i\, \mathbf{S}^i \mathbf{u}_{n-i}, \quad \text{for} \quad n \geq L-1, \tag{1}$$

where $\boldsymbol{\gamma} = [\gamma_0, \gamma_1, \ldots, \gamma_{L-1}]^{\mathrm{T}}$ is the graph filter coefficient vector [19]. The model in (1) efficiently captures the time-varying dynamics of the graph signal. Moreover, it involves only one graph shift operation at each time index $n$. Based on this model, in [19], [20], GDLMS strategies for adaptive filtering of graph signals have been proposed.

However, many real-world problems such as time series prediction, channel equalization, and regression can not be efficiently analyzed with linear models. Therefore, researchers have attempted to address this shortcoming by using a nonlinear model [23], [31]. Inspired by these works, graph kernel adaptive filters have recently been proposed by assuming the following nonlinear relationship between the graph input and output signals [27], [28]:

$$y_{k,n} = f(\mathbf{z}_{k,n}) + \upsilon_{k,n}, \tag{2}$$

where $f : \mathbb{R}^L \to \mathbb{R}$ is a nonlinear continuous function on $\mathbb{R}^L$, $\upsilon_k(n)$ is the observation noise at node $k$, and

$$\mathbf{z}_{k,n} = [u_{k,n}, [\mathbf{S}\mathbf{u}_{n-1}]_k, \ldots, [\mathbf{S}^{L-1}\mathbf{u}_{n-L+1}]_k]^{\mathrm{T}}. \tag{3}$$

Since the graph shift $\mathbf{S}$ does not preserve the energy, GKLMS and GDKLMS algorithms [27], [28] suffer from slow convergence. To solve this problem, in the following, we propose a class of graph kernel filters based on recursive least-squares.

## III. Proposed Algorithm

In estimating the nonlinear function $f(\cdot)$ in (2), kernel methods first map the input regressor $\mathbf{z}_{k,i} \in \mathbb{R}^L$ into a large-dimensional feature space as $\boldsymbol{\phi}(\mathbf{z}_{k,i})$, where the inner products can be evaluated using kernels [24]. A continuous kernel function, which is symmetric, and positive-definite function $\kappa(\cdot,\cdot) : \mathbb{R}^L \times \mathbb{R}^L \to \mathbb{R}$, satisfies Mercer's condition [24], [31]:

$$\kappa(\mathbf{z}_{k,i}, \mathbf{z}_{k,n}) = \boldsymbol{\phi}^{\mathrm{T}}(\mathbf{z}_{k,i})\boldsymbol{\phi}(\mathbf{z}_{k,n}). \tag{4}$$

Without explicitly knowing the mapping $\boldsymbol{\phi}(\cdot)$, inner products in higher dimensional space can be obtained via kernel function evaluation. The kernel is called reproducing kernel if it satisfies the reproducing property [24], namely,

$$\kappa(\mathbf{z}_{k,i}, \mathbf{z}_{k,n}) = \langle \kappa(\cdot, \mathbf{z}_{k,i}), \kappa(\cdot, \mathbf{z}_{k,n}) \rangle_{\mathcal{H}}, \tag{5}$$

where $\mathcal{H}$ is the reproducing kernel Hilbert space (RKHS), the inner product space in which the reproducing kernel is defined and complete [24], and $\langle\cdot,\cdot\rangle_{\mathcal{H}}$ denotes the corresponding inner product. In (5), a representer evaluation at $\mathbf{z}_{k,i}$ is denoted by $\kappa(\cdot, \mathbf{z}_{k,i})$. Our discussion throughout this paper is limited to the Gaussian kernel, which is a well-known Mercer kernel.

### A. RFF-based Centralized GKRLS

In the graph setting, $K$ new input regressors are available at each time index $n$. Thus, given the data pairs $\{\mathbf{z}_{k,i}, d_{k,i}\}_{i=1,k=1}^{n-1,K} \bigcup \{\mathbf{z}_{k,n}\}_{k=1}^{K}$, from representer theorem [32], the estimate of $y_{k,n}$ (i.e., $\hat{y}_{k,n}$) can be expressed as

$$\hat{y}_{k,n} = \sum_{i=1}^{n}\sum_{l=1}^{K} \alpha_{li}\, \kappa(\mathbf{z}_{l,i}, \mathbf{z}_{k,n}) = \mathbf{h}^{\mathrm{T}}\phi(\mathbf{z}_{k,n}), \tag{6}$$

where $\mathbf{h}$ is the vector in $\mathcal{H}$ to be learned. The vector $\mathbf{h}$ can be obtained by minimizing the regularized least-squares problem given below:

$$\min_{\mathbf{h}} \sum_{i=1}^{n}\sum_{l=1}^{K} (y_{l,i} - \mathbf{h}^{\mathrm{T}}\phi(\mathbf{z}_{l,i}))^2 + \lambda\|\mathbf{h}\|^2, \tag{7}$$

where $\lambda$ is the regularization parameter. From (7), one can see that the model dimensionality increases with both time $n$ and size of the graph $K$ (i.e., to obtain system output, we need to perform $Kn$ number of kernel evaluations). In the case of standalone kernel filters, various sparsification methods such as the coherence check criterion and novelty criterion [33], [34] have been used to overcome the problem of growing dimensionality by pruning the redundant input regressors. However, these sparsification methods are not very suitable for network/graph settings because they involve training a dictionary, i.e., every time the underlying model changes, the dictionary must be retrained.

RFF [29] can be used to obtain a computationally efficient and flexible solution for (7). In the $D$-dimensional RFF space,

**Algorithm 1:** RFF-based Centralized GKRLS

**Initialization:** $\mathbf{w}_0 \leftarrow \mathbf{0}$ and $\mathbf{P}_0 = \delta\mathbf{I}$
For every time index $n$, repeat
$\boldsymbol{\psi}_{0,n} \leftarrow \mathbf{w}_{n-1}$, $\mathbf{P}_{0,n} \leftarrow \mathbf{P}_{n-1}$
**for** $k = 1, 2, \ldots, K$ **do**

$$e_{k,n} = y_{k,n} - \mathbf{x}_{k,n}^{\mathrm{T}} \boldsymbol{\psi}_{k-1,n}$$

$$\mathbf{P}_{k,n} = \mathbf{P}_{k-1,n} - \frac{\mathbf{P}_{k-1,n}\mathbf{x}_{k,n}\mathbf{x}_{k,n}^{\mathrm{T}}\mathbf{P}_{k-1,n}}{\lambda + \mathbf{x}_{k,n}^{\mathrm{T}}\mathbf{P}_{k-1,n}\mathbf{x}_{k,n}}$$

$$\boldsymbol{\psi}_{k,n} = \boldsymbol{\psi}_{k-1,n} + \mathbf{P}_{k,n}\, \mathbf{x}_{k,n}\, e_{k,n}$$

**end**
$\mathbf{w}_n \leftarrow \boldsymbol{\psi}_{K,n}$, $\mathbf{P}_n \leftarrow \mathbf{P}_{K,n}$

**Algorithm 2:** RFF-based GDKRLS

**Initialization:** $\mathbf{w}_{k,0} \leftarrow \mathbf{0}$ and $\mathbf{P}_{k,0} = \delta\mathbf{I}$
For every time index $n$, repeat
**Adaptation:** For every node $k$, repeat

$$e_{k,n} = y_{k,n} - \mathbf{x}_{k,n}^{\mathrm{T}}\mathbf{w}_{k,n-1}$$

$$\mathbf{P}_{k,n} = \mathbf{P}_{k,n-1} - \frac{\mathbf{P}_{k,n-1}\mathbf{x}_{k,n}\mathbf{x}_{k,n}^{\mathrm{T}}\mathbf{P}_{k,n-1}}{\lambda + \mathbf{x}_{k,n}^{\mathrm{T}}\mathbf{P}_{k,n-1}\mathbf{x}_{k,n}}$$

$$\boldsymbol{\psi}_{k,n} = \mathbf{w}_{k,n-1} + \mathbf{P}_{k,n}\, \mathbf{x}_{k,n}\, e_{k,n}$$

**Combination:** For every node $k$, repeat

$$\mathbf{w}_{k,n} = \sum_{l \in \mathcal{N}_k} a_{lk}\, \boldsymbol{\psi}_{l,n}$$

a shift-invariant kernel evaluation, i.e., $\kappa(\mathbf{z}_{l,i}, \mathbf{z}_{k,n}) = \kappa(\mathbf{z}_{l,i} - \mathbf{z}_{k,n})$, can be approximated as an inner-product. This approximation modifies the estimation problem (7) into a finite-dimensional linear estimation problem. Moreover, RFF usage avoids the kernel function evaluation. Consider $\mathbf{x}_{k,n}$ to be the mapping of $\mathbf{z}_{k,n}$ into the RFF space $\mathbb{R}^D$. Then, the kernel evaluation can be approximated as $\kappa(\mathbf{z}_{l,i}, \mathbf{z}_{k,n}) \approx \mathbf{x}_{l,i}^{\mathrm{T}}\mathbf{x}_{k,n}$. Thus, the estimate $\hat{y}_{k,n}$ in (6) can be approximated by

$$\hat{y}_{k,n} \approx \Big(\sum_{i=1}^{n}\sum_{l=1}^{K} \alpha_{l,i}\, \mathbf{x}_{l,i}\Big)^{\mathrm{T}} \mathbf{x}_{k,n}(n) = \mathbf{w}^{\mathrm{T}}\mathbf{x}_{k,n}, \tag{8}$$

where $\mathbf{w} \in \mathbb{R}^D$ is the representation of the function $f(\cdot)$ in RFF space. The mapping of $\mathbf{z}_{k,n}$ into the $D$-dimensional RFF space can be accomplished using cosine, exponential and Gaussian feature functions. The same can be read in more detail at [29], [30]. Estimation problem (7) takes the following form in RFF space:

$$\mathbf{w}_n = \min_{\mathbf{w}} \sum_{i=1}^{n}\sum_{l=1}^{K} (y_{l,i} - \mathbf{w}^{\mathrm{T}}\mathbf{x}_{l,i})^2 + \lambda\|\mathbf{w}\|^2. \tag{9}$$

The data from all nodes are collected in global matrices for time $n$: $\mathcal{Y}_n = [\mathbf{y}_n, \mathbf{y}_{n-1}, \ldots, \mathbf{y}_1]^{\mathrm{T}}$ and $\mathcal{X}_n = [\mathbf{X}_n, \mathbf{X}_{n-1}, \ldots \mathbf{X}_1]$ with $\mathbf{y}_i = [y_{1,i}, y_{2,i}, \ldots, y_{K,i}]$ and $\mathbf{X}_i = [\mathbf{x}_{1,i}, \mathbf{x}_{2,i}, \ldots, \mathbf{x}_{K,i}]$, for all $i = 1, 2, \ldots, n$. Then, (9) can alternatively be expressed as

$$\mathbf{w}_n = \min_{\mathbf{w}} \|\mathcal{Y}_n - \mathcal{X}_n^{\mathrm{T}}\mathbf{w}\|^2 + \lambda\|\mathbf{w}\|^2. \tag{10}$$

At time index $n$, the solution to this problem is given by

$$\mathbf{w}_n = \mathbf{P}_n\mathcal{X}_n\mathcal{Y}_n, \tag{11}$$

with $\mathbf{P}_n = (\lambda\mathbf{I} + \mathcal{X}_n\mathcal{X}_n^{\mathrm{T}})^{-1}$. Applying the recursive properties of $\mathbf{P}_n^{-1}$ together with the matrix inversion lemma and following a similar procedure as described in [35], we can update $\mathbf{w}_n$ at each time index $n$ by recursively estimating $\mathbf{P}_n$. Algorithm 1 summarizes the proposed RFF-based centralized GKRLS.

### *B. RFF-based GDKRLS*

It is possible to obtain the solution to the global problem (9) locally at each node in a fully distributed manner, as explained below:

**Adaptation**: By minimizing the local regularized least-squares cost function in RFF space at each node $k$, an intermediate estimate $\boldsymbol{\psi}_{k,n}$ can be computed:

$$\begin{aligned}\boldsymbol{\psi}_{k,n} &= \min_{\boldsymbol{\psi}} \|\mathbf{y}_{k,n} - \mathbf{X}_{k,n}^{\mathrm{T}}\boldsymbol{\psi}\|^2 + \lambda\|\boldsymbol{\psi}\|^2 \\ &= \mathbf{P}_{k,n}\mathbf{X}_{k,n}\mathbf{y}_{k,n},\end{aligned} \tag{12}$$

where $\mathbf{y}_{k,n} = [y_{k,n}, y_{k,n-1}, \ldots, y_{k,1}]^{\mathrm{T}}$, $\mathbf{X}_{k,n} = [\mathbf{x}_{k,n}, \mathbf{x}_{k,n-1}, \ldots, \mathbf{x}_{k,1}]$ and $\mathbf{P}_{k,n} = (\lambda\mathbf{I} + \mathbf{X}_{k,n}\mathbf{X}_{k,n}^{\mathrm{T}})^{-1}$.

**Combination**: Every node diffuses its intermediate estimate vector to its neighbors. The local intermediate estimate vectors are then combined with the neighboring intermediate estimate vectors at each node $k$, resulting in the new local estimate:

$$\mathbf{w}_{k,n} = \sum_{l \in \mathcal{N}_k} a_{lk}\, \boldsymbol{\psi}_{l,n}, \tag{13}$$

where the non-negative combination coefficients $a_{lk}$ satisfy the condition $\sum_{l \in \mathcal{N}_k} a_{lk} = 1$ [18]. Using the recursive property of $\mathbf{P}_{k,n}^{-1}$ together with the matrix inversion lemma, we can obtain the recursion for updating $\boldsymbol{\psi}_{k,n}$ at each time index $n$ and $k$, by recursively estimating $\mathbf{P}_{k,n}$. Algorithm 2 summarizes the proposed RFF-based GDKRLS.

## IV. Mean Convergence Analysis

This section analyzes the mean convergence behavior of the proposed RFF-based GDKLMS. To do so, the following assumptions were made:

- **A**1: The input regressors $\mathbf{x}_{k,n}$ $\forall k, n$ are considered mutually independent with $E[\mathbf{x}_{k,n}\mathbf{x}_{k,n}^{\mathrm{T}}] = \mathbf{R}_{\mathbf{x}_k}$.
- **A**2: The observation noise $\upsilon_{k,n}$ is a zero-mean Gaussian random sequence with variance $\sigma_{\upsilon,k}^2$. Moreover, $\upsilon_{k,n}$ is assumed to be independent of all other data.
- **A**3: The input regressor $\mathbf{x}_{k,n}$ $\forall k$, is an ergodic process. Therefore, for a sufficiently large $n$, we can replace $\mathbf{P}_{k,n}$ and $\mathbf{P}_{k,n}^{-1}$ by their expected values $\mathrm{E}[\mathbf{P}_{\mathrm{k,n}}]$ and $\mathrm{E}[\mathbf{P}_{\mathrm{k,n}}^{-1}]$, respectively. Consequently, we can have $E[\mathbf{P}_{k,n}] \approx E[\mathbf{P}_{k,n}^{-1}]^{-1}$.

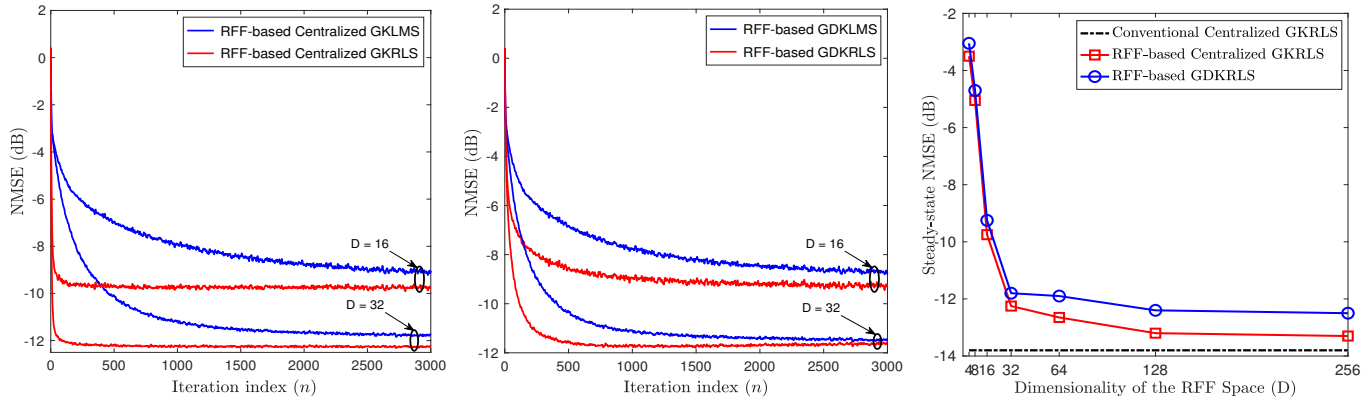


Fig. 1. Performance of the proposed RFF-based GKRLS algorithms: (a) RFF-based centralized GKRLS. (b) RFF-based GDKRLS. Also included are the performance of their LMS counterparts. (c) RFF dimensionality vs. steady-state NMSE of RFF-based centralized GKRLS and RFF-based GDKRLS.

In the $D$-dimensional RFF space, let $\mathbf{w}_{opt}$ be the optimal representation of the nonlinear function $f(\cdot)$ . By defining the intermediate estimate error vector $\tilde{\boldsymbol{\psi}}_{k,n} = \mathbf{w}_{opt} - \boldsymbol{\psi}_{k,n}$ and the estimate error vector $\tilde{\mathbf{w}}_{k,n} = \mathbf{w}_{opt} - \mathbf{w}_{k,n}$, the recursion for the intermediate estimate error vector can be obtained as

$$\tilde{\boldsymbol{\psi}}_{k,n} = \tilde{\mathbf{w}}_{k,n-1} - \mathbf{P}_{k,n}\, \mathbf{x}_{k,n} \left(\mathbf{x}_{k,n}^{\mathrm{T}} \tilde{\mathbf{w}}_{k,n-1} + \upsilon_{k,n}\right), \quad (14a)$$

$$\tilde{\mathbf{w}}_{k,n} = \sum_{l \in \mathcal{N}_k} a_{lk} \tilde{\boldsymbol{\psi}}_{k,n}. \quad (14b)$$

Using the recursive property of $\mathbf{P}_{k,n}^{-1}$, (14a) can be rewritten as

$$\tilde{\boldsymbol{\psi}}_{k,n} = \mathbf{P}_{k,n}\mathbf{P}_{k,n-1}^{-1}\tilde{\mathbf{w}}_{k,n-1} - \mathbf{P}_{k,n}\, \mathbf{x}_{k,n}\, \upsilon_{k,n}. \quad (15)$$

Under the assumption **A**3, $\mathbf{P}_{k,n}\mathbf{P}_{k,n-1}^{-1} \approx \mathbf{I}$ and $\mathbf{P}_{k,n} \approx \frac{1}{n}\mathbf{R}_{\mathbf{x}_k}^{-1}$. Using these approximations, (15) can be further simplified as

$$\tilde{\boldsymbol{\psi}}_{k,n} = \tilde{\mathbf{w}}_{k,n-1} - \frac{1}{n}\, \mathbf{R}_{\mathbf{x}_k}^{-1}\, \mathbf{x}_{k,n}\, \upsilon_{k,n}. \quad (16)$$

Finally, by collecting the local data into global matrices, we have the following global recursion for the estimate error vector:

$$\tilde{\mathbf{w}}_n = \boldsymbol{\mathcal{A}}\tilde{\mathbf{w}}_{n-1} - \frac{1}{n}\boldsymbol{\mathcal{A}}_n\, \mathbf{R}_{\mathbf{x}}^{-1}\hat{\boldsymbol{\mathcal{X}}}_n\boldsymbol{\upsilon}_n, \quad (17)$$

where

$$\begin{aligned} \tilde{\mathbf{w}}_n &= \mathrm{col}\{\tilde{\mathbf{w}}_{1,n}, \tilde{\mathbf{w}}_{2,n}, \ldots, \tilde{\mathbf{w}}_{K,n}\}, \quad (18)\\ \hat{\boldsymbol{\mathcal{X}}}_n &= \mathrm{blockdiag}\{\mathbf{x}_{1,n}, \mathbf{x}_{2,n}, \cdots, \mathbf{x}_{K,n}\},\\ \mathbf{R}_{\mathbf{x}} &= \mathrm{blockdiag}\{\mathbf{R}_{\mathbf{x}_1}, \mathbf{R}_{\mathbf{x}_2}, \cdots, \mathbf{R}_{\mathbf{x}_K}\},\\ \boldsymbol{\upsilon}_n &= [\upsilon_{1,n}, \upsilon_{2,n}, \cdots, \upsilon_{K,n}]^{\mathrm{T}}, \end{aligned}$$

and $\boldsymbol{\mathcal{A}} = \mathbf{A}^{\mathrm{T}} \otimes \mathbf{I}$ with $[\mathbf{A}]_{k,l} = a_{kl}$. Thus, taking the expectation on both sides of (17) together with **A**2, we obtain

$$E[\tilde{\mathbf{w}}_n] = \boldsymbol{\mathcal{A}}\mathrm{E}[\tilde{\mathbf{w}}_{\mathrm{n-1}}]. \quad (19)$$

Since the matrix $\boldsymbol{\mathcal{A}}$ is left stochastic, its spectral radius is less than one. Therefore, one can see that the proposed GDKRLS converges in mean and is also asymptotically unbiased in the RFF space.

## V. Experimental Results

Experimental results are presented in this section to demonstrate the performance of the proposed graph kernel recursive least-squares algorithms. Our experiment is based on a 20-node connected Erdös-Renyi graph with edge probability 0.2. The shift matrix $\mathbf{S}$ was constructed in the following manner: according to the existing adjacency, each edge weight value was generated from a standard uniform distribution $\mathcal{U}(0, 1]$; then, the shift matrix was normalized by its largest eigenvalue. At each node $k$, a time-series realization $t_{k,n}$ and the observation noise $\upsilon_{k,n}$ were drawn from zero-mean Gaussian distributions with covariance matrices $\mathbf{R}_{t_k} = \mathrm{diag}\{\sigma_{t_k}^2\}$ and $\mathbf{R}_{\upsilon} = \mathrm{diag}\{\sigma_{\upsilon_k}^2\}$, respectively, with $\sigma_{t_k}^2 \in \mathcal{U}[1, 1.5]$ and $\sigma_{\upsilon_k}^2 \in \mathcal{U}[0.001, 0.01]$. For each time index $n$, we then have a $K$-dimensional vector, i.e., $\boldsymbol{\tau}_n = [t_{1,n}, t_{2,n}, \ldots, t_{K,n}]^{\mathrm{T}}$. Finally, the graph signal $\mathbf{u}_n$, was generated by projecting $\boldsymbol{\tau}_n$ onto the specified graph, by solving the following optimization problem: $\mathbf{u}_n = \min_{\mathbf{u}} \left\{\|\boldsymbol{\tau}_n - \mathbf{u}\|_2^2 + \beta \mathbf{u}^{\mathrm{T}}\mathbf{L}\mathbf{u}\right\}$ with $\beta = 0.6$. A Metropolis rule [18] was employed for computing the combination coefficients $a_{lk}$ in distributed implementations. The network-level mean square error (NMSE), given by $\mathrm{NMSE} = \frac{1}{K}\sum_{k=1}^{N} e_{k,n}^2$, was considered as the performance metric. The proposed algorithms were simulated to estimate the following nonlinear input-output relationship over a graph:

$$\begin{aligned} f(\mathbf{z}_{k,n}) = &\sqrt{z_{k,1,n}^2 + \sin^2(\pi\ z_{k,4,n})}\\ &+ \left(0.8 - 0.5\exp(-z_{k,2,n}^2)\right) z_{k,3,n}. \end{aligned} \quad (20)$$

The same estimation task was also carried out by RFF-based GKLMS (with $\mu = 0.03$) and GDKLMS (with $\mu = 0.75$) algorithms for comparative evaluation. The results are shown in Figs. 1a and 1b, by plotting the NMSE in dB against the iteration index $n$ obtained by averaging over 1000 independent experiments. From Figs. 1a and 1b, it can be seen that the proposed algorithms exhibit faster convergence rate and better steady-state NMSE compared to their LMS counterparts. Moreover, the proposed algorithms show faster convergence rate and better steady-state NMSE when the RFF space dimension $D$ increases. Next, to study the effect of $D$

(i.e., the dimensionality of the RFF space) on the performance of the proposed RFF-based GKRLS algorithms, we repeated the above simulation exercise for different values of $D$, say $D \in \{4, 8, 16, 32, 64, 128, 256\}$. The corresponding steady-state NMSE (in dB) vs. $D$ is shown in Fig. 1c. In addition, the steady-state NMSE of the conventional GKRLS (i.e., the solution for (7), which requires $nK$ number of kernel function evaluations for estimating the system output $\hat{y}_{k,n}$) was also plotted for comparison purposes. From Fig. 1c, we can see that RFF provides a resonable approximation of the kernel evaluation for $D \geq 32$.

## VI. Conclusions

In this paper, RFF-based GKRLS and GDKRLS algorithms were proposed for modeling the nonlinear relationships between graph input-output signals. Using RFF simplified the GDKRLS implementation and bypassed dictionary learning, which is generally regarded as an essential task in kernel methods. Our analysis showed that the GDKRLS converges in the mean sense to an asymptotically unbiased solution. We performed numerical simulations to investigate the performance of the proposed algorithms. The results showed that the proposed graph kernel adaptive filters based on least-squares could overcome the problem of large eigenvalue spread of the graph-shifted signals.

## VII. Acknowledgment

The Research Council of Norway supported this work.

## References

[1] A. Sandryhaila and J. M. F. Moura, "Discrete Signal Processing on Graphs," *IEEE Trans. Signal Process.*, vol. 61, no. 7, pp. 1644-1656, Apr. 2013

[2] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega and P. Vandergheynst, "The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains," *IEEE Signal Process. Mag.*, vol. 30, no. 3, pp. 83-98, May 2013.

[3] A. Sandryhaila and J. M. F. Moura, "Big data analysis with signal processing on graphs: Representation and processing of massive data sets with irregular structure," *IEEE Signal Process. Mag.*, vol. 31, no. 5, pp. 80-90, Sep. 2014.

[4] A. Ortega, P. Frossard, J. Kovačević, J. M. F. Moura and P. Vandergheynst, "Graph signal processing: overview, challenges, and applications," in *Proc. IEEE*, vol. 106, no. 5, pp. 808-828, May 2018.

[5] M. M. Bronstein, J. Bruna, Y. LeCun, A. Szlam and P. Vandergheynst, "Geometric deep learning: Going beyond Euclidean data," *IEEE Signal Process. Mag.*, vol. 34, no. 4, pp. 18-42, Jul. 2017.

[6] X. Dong, D. Thanou, L. Toni, M. Bronstein and P. Frossard, "Graph signal processing for machine learning: A review and new perspectives," *IEEE Signal Process. Mag.*, vol. 37, no. 6, pp. 117-127, Nov. 2020.

[7] S. Chen, R. Varma, A. Sandryhaila and J. Kovačević, "Discrete signal processing on graphs: Sampling theory," *IEEE Trans. Signal Process.*, vol. 63, no. 24, pp. 6510-6523, Dec. 2015.

[8] F. Gama, E. Isufi, G. Leus and A. Ribeiro, "Graphs, convolutions, and neural networks: From graph filters to graph neural networks," *IEEE Signal Process. Mag.*, vol. 37, no. 6, pp. 128-138, Nov. 2020.

[9] A. Sandryhaila and J. M. F. Moura, "Discrete signal processing on graphs: Graph filters," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2013, pp. 6163-6166.

[10] A. Gavili and X. Zhang, "On the shift operator, graph frequency, and optimal filtering in graph signal processing," *IEEE Trans. Signal Process.*, vol. 65, no. 23, pp. 6303-6318, Dec. 2017.

[11] A. Sandryhaila and J. M. F. Moura, "Discrete signal processing on graphs: Graph Fourier transform," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2013, pp. 6167-6170.

[12] X. Shi, H. Feng, M. Zhai, T. Yang and B. Hu, "Infinite impulse response graph filters in wireless sensor networks," *IEEE Signal Process. Lett.*, vol. 22, no. 8, pp. 1113-1117, Aug. 2015.

[13] E. Isufi, A. Loukas, A. Simonetto and G. Leus, "Autoregressive moving average graph filtering," *IEEE Trans. Signal Process.*, vol. 65, no. 2, pp. 274-288, Jan. 2017.

[14] S. Segarra, A. G. Marques and A. Ribeiro, "Distributed implementation of linear network operators using graph filters," in *Proc. Allerton Conf. Commun., Control, Comput.*, 2015, pp. 1406-1413.

[15] A. Loukas, A. Simonetto, and G. Leus, "Distributed autoregressive moving average graph filters," *IEEE Signal Process. Lett.*, vol.22, pp. 1931-1935, Nov. 2015.

[16] P. Di Lorenzo, P. Banelli, S. Barbarossa, and S. Sardellitti, "Distributed adaptive learning of graph signals," *IEEE Trans. Signal Process.*, vol. 65, no. 16, pp. 4193-4208, Aug. 2017.

[17] P. Di Lorenzo, P. Banelli, E. Isufi, S. Barbarossa and G. Leus, "Adaptive graph signal processing: Algorithms and optimal sampling strategies," *IEEE Trans. Signal Process.*, vol. 66, no. 13, pp. 3584-3598, Jul. 2018.

[18] A. H. Sayed, "Adaptive networks," *Proc. IEEE*, vol. 102, pp. 460-497, Apr. 2014.

[19] R. Nassif, C. Richard, J. Chen, and A. H. Sayed, "Distributed diffusion adaptation over graph signals," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2018, pp. 4129-4133.

[20] F. Hua, R. Nassif, C. Richard, H. Wang and A. H. Sayed, "A preconditioned graph diffusion LMS for adaptive graph signal processing," in *Proc. European Signal Process. Conf.*, 2018 pp. 111-115.

[21] A. Venkitaraman, S. Chatterjee and P. Händel, "Predicting graph signals using kernel regression where the input signal is agnostic to a graph," in *IEEE Trans. Signal Inf. Process. Net.,* vol. 5, no. 4, pp. 698-710, Dec. 2019.

[22] J. F. Walker, N. Jenkins and N. Jenkins *Wind Energy Technology*. Wiley, June 1997.

[23] W. Liu, P. P. Pokharel and J. C. Principe, "The kernel least-mean-square algorithm,," *IEEE Trans. Signal Process.*, vol. 56, no. 2, pp. 543-554, Feb. 2008.

[24] W. Liu, J. C. Príncipe, and S. Haykin, *Kernel Adaptive Filtering*. Wiley, 2010.

[25] A. J. Smola and R. Kondor, "Kernels and regularization on graphs," in *Learning theory and kernel machines. Lecture Notes in Computer Science*, vol. 2777, pp. 144-158, Springer, 2003.

[26] D. Romero, V. N. Ioannidis and G. B. Giannakis, "Kernel-based reconstruction of space-time functions on dynamic graphs," *IEEE J. Sel. Topics Signal Process.*, vol. 11, no. 6, pp. 856-869, Sep. 2017.

[27] V. C. Gogineni, V. R. M. Elias, W. A. Martins and S. Werner, "Graph diffusion kernel LMS using random Fourier features," in *Proc. Asilomar Conf. Signals Syst. Comput.*, Nov. 2020, pp. 1–5.

[28] V. R. M. Elias, V. C. Gogineni, W. A. Martins and S. Werner, "Adaptive graph filters in reproducing kernel Hilbert spaces: Design and performance analysis," *IEEE Trans. Signal, Inf. Process. over Net.*, vol. 7, pp. 62-74, Dec. 2021.

[29] A. Rahimi and B. Recht, "Random features for large-scale kernel machines," in *Proc. Adv. Neural Inf. Process. Syst.,* 2007, pp. 1177–1184.

[30] P. Bouboulis, S. Chouvardas, and S. Theodoridis, "Online distributed learning over networks in RKH spaces using random Fourier features," *IEEE Trans. Signal Process.*, vol. 66, no. 7, pp. 1920–1932, Apr. 2018.

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

[32] B. Schölkopf and A. Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization and Beyond.* Cambridge, MA: MIT Press, 2002 .

[33] P. P. Pokharel, W. Liu, J. C. Príncipe, "Kernel least mean square algorithm with constrained growth," *Signal Process.*, vol. 89, no. 3, pp. 257-265, 2009.

[34] K. Chen, S. Werner, A. Kuh, and Y.-F. Huang, "Nonlinear adaptive filtering with kernel set-membership approach," *IEEE Trans. Signal Process.*, vol. 68, pp. 1515-1528, Feb. 2020.

[35] A. H. Sayed and C. G. Lopes, "Distributed recursive least-squares strategies over adaptive networks," in *Proc. Asilomar Conf. Signals, Syst. Comput.*, 2006, pp. 233-237.