



Delft University of Technology

Adaptive Graph Signal Processing Algorithms and Optimal Sampling Strategies

Di Lorenzo, Paolo; Banelli, Paolo; Isufi, Elvin; Barbarossa, Sergio; Leus, Geert

DOI

[10.1109/TSP.2018.2835384](https://doi.org/10.1109/TSP.2018.2835384)

Publication date

2018

Document Version

Final published version

Published in

IEEE Transactions on Signal Processing

Citation (APA)

Di Lorenzo, P., Banelli, P., Isufi, E., Barbarossa, S., & Leus, G. (2018). Adaptive Graph Signal Processing: Algorithms and Optimal Sampling Strategies. *IEEE Transactions on Signal Processing*, 66(13), 3584-3598. Article 8360065. <https://doi.org/10.1109/TSP.2018.2835384>

Important note

To cite this publication, please use the final published version (if applicable).
Please check the document version above.

Copyright

Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

Takedown policy

Please contact us and provide details if you believe this document breaches copyrights.
We will remove access to the work immediately and investigate your claim.

Green Open Access added to TU Delft Institutional Repository

'You share, we take care!' - Taverne project

<https://www.openaccess.nl/en/you-share-we-take-care>

Otherwise as indicated in the copyright section: the publisher is the copyright holder of this work and the author uses the Dutch legislation to make this work public.

Adaptive Graph Signal Processing: Algorithms and Optimal Sampling Strategies

Paolo Di Lorenzo , *Member, IEEE*, Paolo Banelli , *Member, IEEE*, Elvin Isufi , *Student Member, IEEE*, Sergio Barbarossa , *Fellow, IEEE*, and Geert Leus, *Fellow, IEEE*

Abstract—The goal of this paper is to propose novel strategies for adaptive learning of signals defined over graphs, which are observed over a (randomly) time-varying subset of vertices. We recast two classical adaptive algorithms in the graph signal processing framework, namely the least mean squares (LMS) and the recursive least squares (RLS) adaptive estimation strategies. For both methods, a detailed mean-square analysis illustrates the effect of random sampling on the adaptive reconstruction capability and the steady-state performance. Then, several probabilistic sampling strategies are proposed to design the sampling probability at each node in the graph, with the aim of optimizing the tradeoff between steady-state performance, graph sampling rate, and convergence rate of the adaptive algorithms. Finally, a distributed RLS strategy is derived and shown to be convergent to its centralized counterpart. Numerical simulations carried out over both synthetic and real data illustrate the good performance of the proposed sampling and recovery strategies for (distributed) adaptive learning of signals defined over graphs.

Index Terms—Adaptation and learning, graph signal processing, sampling on graphs, successive convex approximation.

I. INTRODUCTION

IN A large number of applications involving sensor, transportation, communication, social, or biological networks, the observed data can be modeled as signals defined over graphs, or graph signals for short. As a consequence, over the last few years, there was a surge of interest in developing novel analysis methods for graph signals, thus leading to the research field known as graph signal processing (GSP), see, e.g., [1]–[3].

Manuscript received September 11, 2017; revised January 26, 2018; accepted April 29, 2018. Date of publication May 16, 2018; date of current version June 4, 2018. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Yuichi Tanaka. This work was supported by H2020 EUJ Project 5G-MiEdge, Nr. 723171. The work of Paolo Di Lorenzo was supported by the “Fondazione Cassa di Risparmio di Perugia”. (*Corresponding author: Paolo Di Lorenzo.*)

P. Di Lorenzo and S. Barbarossa are with the Department of Information Engineering, Electronics, and Telecommunications, Sapienza University of Rome, Rome 00185, Italy (e-mail: paolo.dilorenzo@uniroma1.it; sergio.barbarossa@uniroma1.it).

P. Banelli is with the Department of Engineering, University of Perugia, Perugia 06125, Italy (e-mail: paolo.banelli@unipg.it).

E. Isufi is with the Department of Engineering, University of Perugia, Perugia 06125, Italy, and also with the Electrical Engineering, Mathematics and Computer Science, Delft University of Technology, CD Delft 2826, The Netherlands (e-mail: isufi-1@tudelft.nl).

G. Leus is with the Electrical Engineering, Mathematics and Computer Science, Delft University of Technology, CD, Delft 2826, The Netherlands (e-mail: g.j.t.leus@tudelft.nl).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TSP.2018.2835384

The goal of GSP is to extend classical processing tools to the analysis of signals defined over an irregular discrete domain, represented by a graph, and one interesting aspect is that such methods typically come to depend on the graph topology, see, e.g., [2], [4]–[7]. Probably, the most important processing tool is represented by graph spectral analysis, which hinges on the definition of the graph Fourier transform (GFT). The GFT was defined in two alternative ways, i.e., the projection of the graph signal onto the eigenvectors of either the Laplacian, see, e.g., [1], [8], [9], or of the adjacency matrix, see, e.g. [2], [10]. Quite recently, a third approach was also proposed, which builds the GFT as the set of orthonormal vectors that minimizes the Lovász extension of the graph cut size [11].

A fundamental task in GSP is to infer the attributes of some vertices from the observation of other vertices. In the GSP literature, this learning task is known as *interpolation from samples*, and emerges whenever cost constraints limit the number of vertices that we can directly observe. An important contribution to sampling theory in GSP is given by [8], later extended in [12] and, very recently, in [10], [13]–[16]. Then, several interpolation methods were proposed to reconstruct signals on graphs from a subset of samples. Machine learning methods typically exploit smoothness of the signal over the graph, see, e.g., [17]–[20], whereas GSP usually considers estimators for graph signals that adhere to the bandlimited model, i.e., signals that belong to the span of some eigenvectors of the graph Laplacian or adjacency matrices, see, e.g., [10], [13], [16]. Sampling theory for graph signals will represent the basic tool to enable proactive sensing of large-scale cyber physical systems, with the aim of ensuring proper control of the system within its physical constraints and guaranteeing a target level of performance, by only checking a limited number of nodes, over a limited number of time instants. In fact, as illustrated in several recent works as, e.g., [10], [13], [21]–[23], when sampling graph signals, what matters is not only how many samples are taken but, most importantly, *where* such samples are collected over the graph, thus catalyzing the interest for designing novel criteria and algorithms to select the set of sampled vertices.

In many applications such as, e.g., transportation networks, brain networks, or communication networks, the observed graph signals are typically time-varying. This requires the development of effective methods capable to learn and track dynamic graph signals from a carefully designed, possibly time-varying, sampling set. Some previous works have considered this specific learning task, see, e.g., [24]–[27]. Specifically, [24] proposed

an LMS estimation strategy enabling adaptive learning and tracking from a limited number of smartly sampled observations. The LMS method in [24] was then extended to the distributed setting in [25]. The work in [26] proposed a kernel-based reconstruction framework to accommodate time-evolving signals over possibly time-evolving topologies, leveraging spatio-temporal dynamics of the observed data. Finally, reference [27] proposes a distributed method for tracking bandlimited graph signals, assuming perfect observations (i.e., there is no measurement noise) and a fixed sampling strategy.

The goal of this paper is to provide fundamental contributions toward the development of an adaptive graph signal processing framework, whose aim is to extend classical adaptive processing methods to the analysis of signals defined over graphs. The main contributions of this work are summarized in the following two points.

- 1) Similarly to the distributed case in [25], we extend the LMS algorithm of [24] to incorporate a *probabilistic sampling* mechanism, where each node of the graph, at every time instant, is sampled with a given probability. Then, we derive a mean-square analysis that illustrates the role played by the sampling probabilities on the reconstruction capabilities and performance of the LMS algorithm. On the basis of the developed analysis, we design probabilistic sampling strategies aimed at minimizing the graph sampling rate (or maximizing the mean-square performance) while imposing accuracy (and/or sampling) constraints.
- 2) We propose an RLS reconstruction strategy that collects data over the graph by the same probabilistic sampling method. Then, after giving necessary and sufficient conditions for adaptive graph signal reconstruction, we formulate an optimization problem to design the sampling probability at each node in the graph, with the aim of reducing the sampling rate while guaranteeing signal reconstruction and a prescribed steady-state performance. Finally, we derive a distributed RLS strategy for graph signals exploiting the alternating direction method of multipliers (ADMM) [28].

The proposed methods can be applied to any graph structure describing the observed signal and, under a bandlimited assumption, enable adaptive reconstruction and tracking from a limited number of observations taken over a (possibly time-varying) subset of vertices. An interesting feature of our strategies is that this subset is allowed to vary over time, provided that the *expected* sampling set satisfies specific conditions enabling graph signal recovery.

The rest of the paper is organized as follows. In Section II, we summarize some GSP tools that will be used throughout the paper. Section III introduces the proposed LMS algorithm for adaptive learning of graph signals, illustrating the conditions enabling signal reconstruction from a (time-varying) subset of samples, deriving a detailed mean-square analysis, and developing useful sampling strategies. Similarly, Section IV considers RLS on graphs, along with its mean-square properties, sampling strategies, and distributed implementation. Then, in Section V we report several numerical simulations, aimed at assessing the validity of the theoretical analysis and the performance

of the proposed algorithms. Finally, Section VI draws some conclusions.

II. BACKGROUND ON GRAPH SIGNAL PROCESSING

Let us consider a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ consisting of a set of N nodes $\mathcal{V} = \{1, 2, \dots, N\}$, along with a set of weighted edges $\mathcal{E} = \{a_{ij}\}_{i,j \in \mathcal{V}}$, such that $a_{ij} > 0$, if there is a link from node j to node i , or $a_{ij} = 0$, otherwise. The adjacency matrix \mathbf{A} of a graph is the collection of all the weights, i.e., $\mathbf{A} = \{a_{ij}\}_{i,j = 1, \dots, N}$. The Laplacian matrix is defined as $\mathbf{L} = \text{diag}(\mathbf{1}^T \mathbf{A}) - \mathbf{A}$, where $\mathbf{1}$ denotes the vector composed of all ones, and $\text{diag}(\mathbf{x})$ is a matrix having \mathbf{x} as main diagonal, and zeros elsewhere. If the graph is *undirected*, the Laplacian matrix is symmetric and positive semi-definite, and can be decomposed as $\mathbf{L} = \mathbf{U} \mathbf{\Lambda} \mathbf{U}^H$, where \mathbf{U} collects an orthonormal set of eigenvectors of \mathbf{L} in its columns, whereas $\mathbf{\Lambda}$ is a diagonal matrix containing the real eigenvalues of \mathbf{L} .

A signal \mathbf{x} over a graph \mathcal{G} is defined as a mapping from the vertex set to the set of complex numbers, i.e., $\mathbf{x} : \mathcal{V} \rightarrow \mathbb{C}$. The GFT \mathbf{s} of a signal \mathbf{x} is defined as the projection onto the set of vectors $\mathbf{U} = \{\mathbf{u}_i\}_{i=1, \dots, N}$, i.e.,

$$\mathbf{s} = \mathbf{U}^H \mathbf{x}, \quad (1)$$

where $\{\mathbf{u}_i\}_{i=1, \dots, N}$ form an orthonormal basis and are commonly chosen as the eigenvectors of either the graph Laplacian [1], or of the adjacency matrix [2], [10] (always assuming these matrices are normal). In this paper, we basically follow the approach from [1], but the theory can be directly extended to other cases by simply substituting in (1) the corresponding GFT basis. We denote the support of \mathbf{s} in (1) as

$$\mathcal{F} = \{i \in \{1, \dots, N\} : s_i \neq 0\},$$

and the *bandwidth* of the graph signal \mathbf{x} is defined as the cardinality of \mathcal{F} , i.e., $|\mathcal{F}|$. The space of all signals whose GFT is supported on the set \mathcal{F} is known as the *Paley-Wiener space* for the set \mathcal{F} [8].

Finally, given a subset of vertices $\mathcal{S} \subseteq \mathcal{V}$, we define a vertex-limiting operator as the diagonal matrix

$$\mathbf{D}_{\mathcal{S}} = \text{diag}\{\mathbf{1}_{\mathcal{S}}\}, \quad (2)$$

where $\mathbf{1}_{\mathcal{S}}$ is the set indicator vector, whose i -th entry is equal to one, if $i \in \mathcal{S}$, or zero otherwise. Similarly, given a subset of frequency indices $\mathcal{F} \subseteq \mathcal{V}$, we introduce the filtering operator

$$\mathbf{B}_{\mathcal{F}} = \mathbf{U}_{\mathcal{F}} \mathbf{U}_{\mathcal{F}}^H, \quad (3)$$

where $\mathbf{U}_{\mathcal{F}} \in \mathbb{C}^{N \times |\mathcal{F}|}$ collects the subset of columns of matrix \mathbf{U} in (1) associated to the frequency indices \mathcal{F} . It is immediate to check that both matrices $\mathbf{D}_{\mathcal{S}}$ and $\mathbf{B}_{\mathcal{F}}$ are self-adjoint and idempotent, and so they represent orthogonal projectors onto the set of vertices \mathcal{S} and the Paley-Wiener space \mathcal{F} , respectively.

III. ADAPTIVE LEAST MEAN SQUARES ESTIMATION OF GRAPH SIGNALS

Let us consider a signal $\mathbf{x}^o = \{x_i^o\}_{i=1}^N \in \mathbb{C}^N$ defined over the graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. To enable sampling of \mathbf{x}^o without loss of information, the following is assumed:

Assumption 1 (Bandlimited): The signal \mathbf{x}^o is \mathcal{F} -bandlimited over the graph \mathcal{G} , i.e., its spectral content is different from zero only on the set of indices \mathcal{F} . ■

If the signal support \mathcal{F} is fixed and known beforehand, from (1), the graph signal \mathbf{x}^o can be modeled in compact form as:

$$\mathbf{x}^o = \mathbf{U}_{\mathcal{F}} \mathbf{s}^o, \quad (4)$$

where $\mathbf{s}^o \in \mathbb{C}^{|\mathcal{F}|}$ is the vector of GFT coefficients of the frequency support of the graph signal \mathbf{x}^o . At each time n , noisy samples of the signal are taken over a time-varying subset of vertices, according to the following model:

$$\mathbf{y}[n] = \mathbf{D}_{\mathcal{S}[n]} (\mathbf{x}^o + \mathbf{v}[n]) = \mathbf{D}_{\mathcal{S}[n]} \mathbf{U}_{\mathcal{F}} \mathbf{s}^o + \mathbf{D}_{\mathcal{S}[n]} \mathbf{v}[n] \quad (5)$$

where $\mathbf{D}_{\mathcal{S}[n]} = \text{diag}\{d_1[n], \dots, d_N[n]\} \in \mathbb{R}^{N \times N}$ [cf. (2)], with $d_i[n]$ denoting a sampling random binary coefficient, which is equal to 1 if $i \in \mathcal{S}[n]$, and 0 otherwise (i.e., $\mathcal{S}[n]$ represents the *instantaneous*, random sampling set at time n); and $\mathbf{v}[n] \in \mathbb{C}^N$ is zero-mean, spatially and temporally independent observation noise, with covariance matrix $\mathbf{C}_v = \text{diag}\{\sigma_1^2, \dots, \sigma_N^2\}$. The estimation task consists in recovering the vector \mathbf{x}^o (or, equivalently, its GFT \mathbf{s}^o) from the noisy, streaming, and partial observations $\mathbf{y}[n]$ in (5). Following an LMS approach [29], from (5), the optimal estimate for \mathbf{s}^o can be found as the vector that solves the optimization problem¹:

$$\min_{\mathbf{s}} \mathbb{E} \|\mathbf{D}_{\mathcal{S}[n]} (\mathbf{y}[n] - \mathbf{U}_{\mathcal{F}} \mathbf{s})\|^2 \quad (6)$$

where $\mathbb{E}(\cdot)$ denotes the expectation operator, and in (6) we have exploited the fact that $\mathbf{D}_{\mathcal{S}[n]}$ is an idempotent matrix for any fixed n [cf. (2)]. A typical LMS-type solution proceeds to optimize (6) by means of a stochastic steepest-descent procedure, relying only on instantaneous information. Thus, letting $\widehat{\mathbf{s}}[n]$ be the current estimate of vector \mathbf{s}^o , the LMS algorithm for graph signals evolves as:

$$\widehat{\mathbf{s}}[n+1] = \widehat{\mathbf{s}}[n] + \mu \mathbf{U}_{\mathcal{F}}^H \mathbf{D}_{\mathcal{S}[n]} (\mathbf{y}[n] - \mathbf{U}_{\mathcal{F}} \widehat{\mathbf{s}}[n]), \quad (7)$$

where $\mu > 0$ is a (sufficiently small) step-size, and we have exploited the fact that $\mathbf{D}_{\mathcal{S}[n]}$ is an idempotent operator. Finally, exploiting (4) and (3), recursion (7) can be equivalently recast with respect to the current estimates of graph signal \mathbf{x}^o , say $\widehat{\mathbf{x}}[n]$, as illustrated in Algorithm 1. At every iteration n , Algorithm 1 evolves implementing an alternating orthogonal projection onto the instantaneous sampling set $\mathcal{S}[n]$ (through $\mathbf{D}_{\mathcal{S}[n]}$) and the Paley-Wiener space \mathcal{F} (through $\mathbf{B}_{\mathcal{F}}$), with an overall complexity given by $O(|\mathcal{F}||\mathcal{S}[n]|)$. The LMS strategy in Algorithm 1 extends the method previously proposed in [24] by incorporating the random sampling mechanism defined by the *time-varying* sampling operator $\mathbf{D}_{\mathcal{S}[n]}$. Of course, the mean-square properties of the LMS recursion crucially depend on the properties of the random sampling operator $\mathbf{D}_{\mathcal{S}[n]}$. Thus, in the following, we will show how the design of the sampling strategy affects the reconstruction capability, the learning rate, and the steady-state performance of Algorithm 1. Before moving forward, we introduce an independence assumption on the random sampling process.

¹Note that, since $\mathbf{U}_{\mathcal{F}}$ has orthonormal columns, estimating \mathbf{s}^o or \mathbf{x}^o is equivalent from a mean square error perspective.

Algorithm 1: LMS On Graphs.

Start with random $\widehat{\mathbf{x}}[0]$. Given a sufficiently small step-size $\mu > 0$, for each time $n \geq 0$, repeat:

$$\widehat{\mathbf{x}}[n+1] = \widehat{\mathbf{x}}[n] + \mu \mathbf{B}_{\mathcal{F}} \mathbf{D}_{\mathcal{S}[n]} (\mathbf{y}[n] - \widehat{\mathbf{x}}[n])$$

Assumption 2 (Independent sampling): The random variables extracted from the sampling process $\{d_i[l]\}$ are temporally and spatially independent, for all i , and $l \leq n$. ■

Assumption 2 can be easily realized in practice if we collect data using a set of sensors, each one working in an independent fashion (i.e., spatial independence) and without taking into account past or future actions (i.e., temporal independence).

A. Adaptive Reconstruction From Probabilistic Sampling

Assuming the random sampling and observation processes $\{d_i[n]\}_{i=1}^N$ and $\{\mathbf{y}[n]\}$ to be stationary over time,² the solution of problem (6) is given by the vector \mathbf{s}^o that satisfies:

$$\mathbf{U}_{\mathcal{F}}^H \text{diag}(\mathbf{p}) \mathbf{U}_{\mathcal{F}} \mathbf{s}^o = \mathbf{U}_{\mathcal{F}}^H \mathbb{E}\{\mathbf{y}[n]\}, \quad (8)$$

where $\mathbf{p} = (p_1, \dots, p_N)^T \in \mathbb{R}^N$ represents the sampling probability vector, with $p_i = \mathbb{E}\{d_i[n]\}$, $i = 1, \dots, N$, denoting the probability that node i is sampled at time n . The system of equations (8) admits a unique solution \mathbf{s}^o if the square matrix $\mathbf{U}_{\mathcal{F}}^H \text{diag}(\mathbf{p}) \mathbf{U}_{\mathcal{F}}$ is full rank or, equivalently, if

$$\lambda_{\min} \left(\mathbf{U}_{\mathcal{F}}^H \text{diag}(\mathbf{p}) \mathbf{U}_{\mathcal{F}} \right) > 0, \quad (9)$$

where $\lambda_{\min}(\mathbf{Y})$ is the minimum eigenvalue of matrix \mathbf{Y} . Also, let us denote the *expected sampling set* by

$$\overline{\mathcal{S}} = \{i = 1, \dots, N \mid p_i > 0\},$$

i.e., the set of nodes of the graph that are sampled with a probability strictly greater than zero. To provide a necessary and sufficient condition for signal reconstruction, we proceed similarly to [13], [25]. Since $p_i > 0$ for all $i \in \overline{\mathcal{S}}$,

$$\text{rank} \left(\mathbf{U}_{\mathcal{F}}^H \text{diag}(\mathbf{p}) \mathbf{U}_{\mathcal{F}} \right) = \text{rank} \left(\mathbf{U}_{\mathcal{F}}^H \mathbf{D}_{\overline{\mathcal{S}}} \mathbf{U}_{\mathcal{F}} \right), \quad (10)$$

i.e., condition (9) holds true if matrix $\mathbf{U}_{\mathcal{F}}^H \mathbf{D}_{\overline{\mathcal{S}}} \mathbf{U}_{\mathcal{F}}$ has full rank, where $\mathbf{D}_{\overline{\mathcal{S}}}$ is a vertex-limiting operator that projects onto the expected sampling set $\overline{\mathcal{S}}$ [cf. (2)]. Let us now introduce $\mathbf{D}_{\overline{\mathcal{S}}^c} = \mathbf{I} - \mathbf{D}_{\overline{\mathcal{S}}}$, which projects onto the complement of the expected sampling set, i.e. $\overline{\mathcal{S}}^c = \mathcal{V} \setminus \overline{\mathcal{S}} = \{i = 1, \dots, N \mid p_i = 0\}$. Then, signal reconstruction is possible if $\mathbf{U}_{\mathcal{F}}^H \mathbf{D}_{\overline{\mathcal{S}}} \mathbf{U}_{\mathcal{F}} = \mathbf{I} - \mathbf{U}_{\mathcal{F}}^H \mathbf{D}_{\overline{\mathcal{S}}^c} \mathbf{U}_{\mathcal{F}}$ is invertible, i.e., if $\|\mathbf{U}_{\mathcal{F}}^H \mathbf{D}_{\overline{\mathcal{S}}^c} \mathbf{U}_{\mathcal{F}}\|_2 < 1$ or, equivalently, if

$$\|\mathbf{D}_{\overline{\mathcal{S}}^c} \mathbf{U}_{\mathcal{F}}\|_2 < 1. \quad (11)$$

As illustrated in [13], [24], condition (11) implies that there are no \mathcal{F} -bandlimited signals that are perfectly localized over the set $\overline{\mathcal{S}}^c$, thus providing a *necessary* and *sufficient* condition

²The stationarity assumption is necessary to study analytically the asymptotic behavior of adaptive filters, and is typically used in the literature [29].

for adaptive graph signal reconstruction. In particular, differently from previous works on deterministic and time-invariant sampling of graph signals, see, e.g., [8], [10], [12]–[15], [24], condition (11) depends on the *expected* sampling set. As a consequence, the proposed LMS algorithm with probabilistic sampling does not need to collect all the data necessary to reconstruct the graph signal at each iteration, but can learn acquiring the needed information over time. The only important thing required by condition (11) is that a sufficiently large number of nodes is sampled in *expectation* (i.e., the expected sampling set \bar{S} contains a sufficiently large number of nodes). Moreover, as we will see in the following sections, a great advantage of our probabilistic methods with respect to deterministic ones comes from a simplification of the algorithms that we use to select the optimal sampling strategy. Indeed, deterministic sampling typically leads to complex integer optimization problems, which must be either relaxed or solved approximatively using greedy approaches, when a brute force method is not feasible due to complexity reasons. On the other side, using our probabilistic formulation, we can obtain sampling strategies that are based on convex problem formulations [see, e.g., (18) and (35)], whose globally optimal solution can be found using efficient numerical methods. To this aim, in the sequel, we will carry out a detailed mean-square analysis for Algorithm 1.

B. Mean-Square Performance

We now illustrate how the sampling probability vector \mathbf{p} affects the mean-square behavior of Algorithm 1. Let $\tilde{\mathbf{x}}[n] = \hat{\mathbf{x}}[n] - \mathbf{x}^o$ and $\tilde{\mathbf{s}}[n] = \hat{\mathbf{s}}[n] - \mathbf{s}^o$ be the error vectors on the graph signal and its GFT, respectively, at time n . Thus, using (5) in (7), we obtain:

$$\tilde{\mathbf{s}}[n+1] = (\mathbf{I} - \mu \mathbf{U}_{\mathcal{F}}^H \mathbf{D}_{S[n]} \mathbf{U}_{\mathcal{F}}) \tilde{\mathbf{s}}[n] + \mu \mathbf{U}_{\mathcal{F}}^H \mathbf{D}_{S[n]} \mathbf{v}[n]. \quad (12)$$

Starting from (12), it is possible to derive a complete mean-square analysis of Algorithm 1, which relies also on the following assumption.

Assumption 3 (Small step-size): The step-size μ is chosen sufficiently small so that terms that depend on higher-order powers of μ can be ignored. ■

The main results are summarized in the following Theorem.

Theorem 1: Given model (5), under Assumptions 2 and 3, and for any initial condition, Algorithm 1 is stable in the mean-square error sense if the sampling probability vector \mathbf{p} and the step-size μ are chosen to satisfy (11) and

$$0 < \mu < \frac{2\lambda_{\min}(\mathbf{U}_{\mathcal{F}}^H \text{diag}(\mathbf{p})\mathbf{U}_{\mathcal{F}})}{\lambda_{\max}^2(\mathbf{U}_{\mathcal{F}}^H \text{diag}(\mathbf{p})\mathbf{U}_{\mathcal{F}})}. \quad (13)$$

Furthermore, the mean-square deviation (MSD) writes as³

$$\begin{aligned} \text{MSD} &= \lim_{n \rightarrow \infty} \mathbb{E} \|\tilde{\mathbf{x}}[n]\|^2 = \lim_{n \rightarrow \infty} \mathbb{E} \|\tilde{\mathbf{s}}[n]\|^2 \\ &= \frac{\mu}{2} \text{Tr} \left[\left(\mathbf{U}_{\mathcal{F}}^H \text{diag}(\mathbf{p})\mathbf{U}_{\mathcal{F}} \right)^{-1} \mathbf{U}_{\mathcal{F}}^H \text{diag}(\mathbf{p})\mathbf{C}_v \mathbf{U}_{\mathcal{F}} \right] + O(\mu^2) \end{aligned} \quad (14)$$

³Note that, if (11) holds and considering spatially white noise samples with equal variance σ_v^2 , expression (14) reduces to $\text{MSD} = \frac{\mu}{2} |\mathcal{F}| \sigma_v^2 + O(\mu^2)$.

and the convergence rate α is well approximated by

$$\alpha \simeq 1 - 2\mu\lambda_{\min}(\mathbf{U}_{\mathcal{F}}^H \text{diag}(\mathbf{p})\mathbf{U}_{\mathcal{F}}) \quad (15)$$

$$\text{when } \mu \ll \frac{2\lambda_{\min}(\mathbf{U}_{\mathcal{F}}^H \text{diag}(\mathbf{p})\mathbf{U}_{\mathcal{F}})}{\lambda_{\max}^2(\mathbf{U}_{\mathcal{F}}^H \text{diag}(\mathbf{p})\mathbf{U}_{\mathcal{F}})}.$$

Proof: See Appendix A. ■

C. Optimal Sampling Strategies

The mean-square analysis in Sec. III-B illustrates how the convergence rate and the mean-square performance of Algorithm 1 depend on the sampling probability vector \mathbf{p} [cf. (14) and (15)]. Then, following a sparse sensing approach [30], [31], the goal of this section is to develop optimal sampling strategies aimed at designing the probability vector \mathbf{p} that optimizes the tradeoff between steady-state performance, graph sampling rate, and convergence rate of Algorithm 1. In the sequel, under Assumption 3, we neglect the term $O(\mu^2)$ in (14), and consider (15) as the convergence rate.

1) *Minimum Graph Sampling Rate Subject to Learning Constraints:* The first sampling strategy aims at designing the probability vector \mathbf{p} that minimizes the total sampling rate over the graph, i.e., $\mathbf{1}^T \mathbf{p}$, while guaranteeing a target performance in terms of MSD in (14) and of convergence rate in (15). The optimization problem can be cast as:

$$\begin{aligned} \min_{\mathbf{p}} \quad & \mathbf{1}^T \mathbf{p} \\ \text{subject to} \quad & \lambda_{\min}(\mathbf{U}_{\mathcal{F}}^H \text{diag}(\mathbf{p})\mathbf{U}_{\mathcal{F}}) \geq \frac{1 - \bar{\alpha}}{2\mu} \\ & \text{Tr} \left[\left(\mathbf{U}_{\mathcal{F}}^H \text{diag}(\mathbf{p})\mathbf{U}_{\mathcal{F}} \right)^{-1} \mathbf{U}_{\mathcal{F}}^H \text{diag}(\mathbf{p})\mathbf{C}_v \mathbf{U}_{\mathcal{F}} \right] \leq \frac{2\gamma}{\mu} \\ & \mathbf{0} \leq \mathbf{p} \leq \mathbf{p}^{\max} \end{aligned} \quad (16)$$

The first constraint imposes that the convergence rate of the algorithm is larger than a desired value, i.e., α in (15) is smaller than a target value, say, e.g., $\bar{\alpha} \in (0, 1)$. Note that the first constraint on the convergence rate also guarantees adaptive signal reconstruction [cf. (9)]. The second constraint guarantees a target mean-square performance, i.e., the MSD in (14) must be less than or equal to a prescribed value, say, e.g., $\gamma > 0$. Finally, the last constraint limits the probability vector to lie in the box $p_i \in [0, p_i^{\max}]$, for all i , with $0 \leq p_i^{\max} \leq 1$ denoting an upper bound on the sampling probability at each node that might depend on external factors such as, e.g., limited energy, processing, and/or communication resources, node or communication failures, etc.

Unfortunately, problem (16) is non-convex, due to the presence of the non-convex constraint on the MSD. To handle the non-convexity of (16), we follow two different approaches. In the first place, under Assumption 3 [i.e., neglecting the terms $O(\mu^2)$], we exploit an upper bound of the MSD function in (14),

given by [32]:

$$\text{MSD}(\mathbf{p}) \leq \overline{\text{MSD}}(\mathbf{p}) \triangleq \frac{\mu}{2} \frac{\text{Tr}(\mathbf{U}_{\mathcal{F}}^H \text{diag}(\mathbf{p}) \mathbf{C}_v \mathbf{U}_{\mathcal{F}})}{\lambda_{\min}(\mathbf{U}_{\mathcal{F}}^H \text{diag}(\mathbf{p}) \mathbf{U}_{\mathcal{F}})}, \quad (17)$$

for all $\mathbf{p} \in \mathbb{R}^N$. Of course, replacing the MSD function (14) with the bound (17), the second constraint of problem (16) is always satisfied. Thus, exploiting the upper bound (17), we formulate a surrogate optimization problem for the selection of the probability vector \mathbf{p} , which can be cast as:

$$\begin{aligned} & \min_{\mathbf{p}} \mathbf{1}^T \mathbf{p} \\ & \text{subject to} \\ & \lambda_{\min}(\mathbf{U}_{\mathcal{F}}^H \text{diag}(\mathbf{p}) \mathbf{U}_{\mathcal{F}}) \geq \frac{1 - \bar{\alpha}}{2\mu} \\ & \frac{\text{Tr}(\mathbf{U}_{\mathcal{F}}^H \text{diag}(\mathbf{p}) \mathbf{C}_v \mathbf{U}_{\mathcal{F}})}{\lambda_{\min}(\mathbf{U}_{\mathcal{F}}^H \text{diag}(\mathbf{p}) \mathbf{U}_{\mathcal{F}})} \leq \frac{2\gamma}{\mu} \\ & \mathbf{0} \leq \mathbf{p} \leq \mathbf{p}^{\max} \end{aligned} \quad (18)$$

Problem (18) is now a convex optimization problem. Indeed, the second constraint of problem (18) involves the ratio of a convex function over a concave function. Since both functions at numerator and denominator of (17) are differentiable and positive for all \mathbf{p} satisfying the first and third constraint of problem (18), the function is pseudo-convex [33], and all its sub-level sets are convex sets. This argument, coupled with the convexity of the objective function and of the sets defined by the first and third constraints, proves the convexity of the problem in (18), whose global solution can be found using efficient numerical tools [34].

The second approach exploits successive convex approximation (SCA) methods [35], whose aim is to find local optimal solutions of (16). The issue in (16) is the non-convexity of the feasible set. Thus, following the approach from [35], we define the (non-convex) set \mathcal{T} and the (convex) set \mathcal{K} as:

$$\mathcal{T} \triangleq \left\{ \begin{array}{l} \lambda_{\min}(\mathbf{U}_{\mathcal{F}}^H \text{diag}(\mathbf{p}) \mathbf{U}_{\mathcal{F}}) \geq \frac{1 - \bar{\alpha}}{2\mu} \\ \frac{\mu}{2} \text{Tr} \left[\left(\mathbf{U}_{\mathcal{F}}^H \text{diag}(\mathbf{p}) \mathbf{U}_{\mathcal{F}} \right)^{-1} \mathbf{U}_{\mathcal{F}}^H \text{diag}(\mathbf{p}) \mathbf{C}_v \mathbf{U}_{\mathcal{F}} \right] \leq \gamma \\ \mathbf{0} \leq \mathbf{p} \leq \mathbf{p}^{\max} \end{array} \right.$$

and

$$\mathcal{K} \triangleq \left\{ \begin{array}{l} \lambda_{\min}(\mathbf{U}_{\mathcal{F}}^H \text{diag}(\mathbf{p}) \mathbf{U}_{\mathcal{F}}) \geq \frac{1 - \bar{\alpha}}{2\mu} \\ \mathbf{0} \leq \mathbf{p} \leq \mathbf{p}^{\max} \end{array} \right.$$

Then, we replace the (second) non-convex constraint in \mathcal{T} with a convex approximation $\widetilde{\text{MSD}}(\mathbf{p}; \mathbf{z}) : \mathcal{K} \times \mathcal{T} \rightarrow \mathbb{R}$ that satisfies the following conditions:

- C1) $\widetilde{\text{MSD}}(\bullet; \mathbf{z})$ is convex on \mathcal{K} for all $\mathbf{z} \in \mathcal{T}$;
- C2) $\widetilde{\text{MSD}}(\mathbf{z}; \mathbf{z}) = \text{MSD}(\mathbf{z})$ for all $\mathbf{z} \in \mathcal{T}$;
- C3) $\text{MSD}(\mathbf{p}) \leq \widetilde{\text{MSD}}(\mathbf{p}; \mathbf{z})$ for all $\mathbf{p} \in \mathcal{K}$ and $\mathbf{z} \in \mathcal{T}$;
- C4) $\widetilde{\text{MSD}}(\bullet; \bullet)$ is continuous on $\mathcal{K} \times \mathcal{T}$;
- C5) $\nabla_{\mathbf{p}} \text{MSD}(\mathbf{z}) = \nabla_{\mathbf{p}} \widetilde{\text{MSD}}(\mathbf{z}; \mathbf{z})$ for all $\mathbf{z} \in \mathcal{T}$;

Algorithm 2: SCA Method for Problem (16).

Set $k = 1$. Start with $\mathbf{p}[1] \in \mathcal{T}$. Then, for $k \geq 1$, repeat until convergence the following steps:

- S.1) $\hat{\mathbf{p}}[k] = \arg \min_{\mathbf{p} \in \tilde{\mathcal{T}}(\mathbf{p}[k])} \mathbf{1}^T \mathbf{p} + \frac{\tau}{2} \|\mathbf{p} - \mathbf{p}[k]\|^2$
 - S.2) $\mathbf{p}[k+1] = \mathbf{p}[k] + \varepsilon[k] (\hat{\mathbf{p}}[k] - \mathbf{p}[k])$
-

C6) $\nabla_{\mathbf{p}} \widetilde{\text{MSD}}(\bullet; \bullet)$ is continuous on $\mathcal{K} \times \mathcal{T}$;

where $\nabla_{\mathbf{p}} \widetilde{\text{MSD}}(\mathbf{z}; \mathbf{z})$ denotes the partial gradient of $\widetilde{\text{MSD}}$ with respect to the first argument evaluated at \mathbf{z} (the second argument is kept fixed at \mathbf{z}). Among the possible choices for $\widetilde{\text{MSD}}$ (see [35] for details), since the MSD function in (14) has Lipschitz continuous gradient on \mathcal{K} (with Lipschitz constant L), a possible choice for the approximation is given by:

$$\begin{aligned} \widetilde{\text{MSD}}(\mathbf{p}; \mathbf{z}) &= \text{MSD}(\mathbf{z}) + \nabla_{\mathbf{p}} \text{MSD}(\mathbf{z})^T (\mathbf{p} - \mathbf{z}) \\ &\quad + \frac{L}{2} \|\mathbf{p} - \mathbf{z}\|^2, \end{aligned} \quad (19)$$

with $\mathbf{p} \in \mathcal{K}$, $\mathbf{z} \in \mathcal{T}$, and satisfying all conditions C1-C6. Thus, given the current estimate of \mathbf{p} at time k , say $\mathbf{p}[k]$, we define a surrogate optimization set $\tilde{\mathcal{T}}(\mathbf{p}[k])$ given by:

$$\tilde{\mathcal{T}}(\mathbf{p}[k]) \triangleq \left\{ \begin{array}{l} \lambda_{\min}(\mathbf{U}_{\mathcal{F}}^H \text{diag}(\mathbf{p}) \mathbf{U}_{\mathcal{F}}) \geq \frac{1 - \bar{\alpha}}{2\mu} \\ \widetilde{\text{MSD}}(\mathbf{p}; \mathbf{p}[k]) \leq \gamma \\ \mathbf{0} \leq \mathbf{p} \leq \mathbf{p}^{\max} \end{array} \right.$$

with $\widetilde{\text{MSD}}(\mathbf{p}; \mathbf{p}[k])$ given by (19). Then, the SCA algorithm for problem (16) proceeds as described in Algorithm 2. The updating scheme reads: at every iteration k , given the current estimate $\mathbf{p}[k]$, the first step of Algorithm 2 solves a surrogate optimization problem involving the objective function $\mathbf{1}^T \mathbf{p}$, augmented with a proximal regularization term (with $\tau > 0$), and the surrogate set $\tilde{\mathcal{T}}(\mathbf{p}[k])$. Then, the second step of Algorithm 2 generates the new point $\mathbf{p}[k+1]$ as a convex combination of the current estimate $\mathbf{p}[k]$ and the solutions $\hat{\mathbf{p}}[k]$ of the surrogate problem, using a step-size sequence $\varepsilon[k]$. Several choices for the step-size rule are possible to guarantee convergence of Algorithm 2. In particular, we might use a diminishing step-size rule satisfying the following properties:

$$\varepsilon[k] > 0, \quad \sum_{k=0}^{\infty} \varepsilon[k] = \infty, \quad \sum_{k=0}^{\infty} \varepsilon^2[k] < \infty. \quad (20)$$

Using a step-size rule satisfying (20), and assuming $\tilde{\mathcal{T}}(\mathbf{p}[k])$ as a surrogate feasible set at each iteration k , the sequence generated by Algorithm 2 converges to a local optimal solution of problem (16), see [35, Theorem 2] for details.

2) *Minimum MSD With Sampling and Learning Constraints:* The second sampling strategy aims at designing the probability vector \mathbf{p} that minimizes the MSD in (14), while imposing that the convergence rate is larger than a desired value, and the total sampling rate $\mathbf{1}^T \mathbf{p}$ is limited by a budget constraint. The

Algorithm 3: Dinkelbach Method for Problem (22).

Set $k = 1$. Start with $\mathbf{p}[1] \in \mathcal{C}$ and $\omega[1] = f(\mathbf{p}[1])/g(\mathbf{p}[1])$. Then, for $k \geq 1$, repeat the following steps:

$$\text{S.1) } \mathbf{p}[k+1] = \arg \min_{\mathbf{p} \in \mathcal{C}} h(\mathbf{p}, \omega[k])$$

S.2) If $h(\mathbf{p}[k+1], \omega[k]) = 0$, STOP and $\mathbf{p}^* = \mathbf{p}[k+1]$; otherwise, $\omega[k+1] = f(\mathbf{p}[k+1])/g(\mathbf{p}[k+1])$, $k = k+1$, and go to S.1

optimization problem can then be cast as:

$$\begin{aligned} \min_{\mathbf{p}} \quad & \text{Tr} \left[\left(\mathbf{U}_{\mathcal{F}}^H \text{diag}(\mathbf{p}) \mathbf{U}_{\mathcal{F}} \right)^{-1} \mathbf{U}_{\mathcal{F}}^H \text{diag}(\mathbf{p}) \mathbf{C}_v \mathbf{U}_{\mathcal{F}} \right] \\ \text{s.t. } \quad & \mathbf{p} \in \mathcal{C} \triangleq \begin{cases} \lambda_{\min} \left(\mathbf{U}_{\mathcal{F}}^H \text{diag}(\mathbf{p}) \mathbf{U}_{\mathcal{F}} \right) \geq \frac{1 - \bar{\alpha}}{2\mu} \\ \mathbf{1}^T \mathbf{p} \leq \mathcal{P} \\ \mathbf{0} \leq \mathbf{p} \leq \mathbf{p}^{\max} \end{cases} \end{aligned} \quad (21)$$

where $\mathcal{P} \in [0, \mathbf{1}^T \mathbf{p}^{\max}]$ is the budget on the total sampling rate. Problem (21) has a convex feasible set \mathcal{C} , but it is non-convex because of the MSD objective function. Again, to handle the non-convexity of (21), we can follow two different paths. Exploiting the bound in (17), it is possible to formulate an approximated optimization problem for the design of the probability vector \mathbf{p} , which can be cast as:

$$\min_{\mathbf{p} \in \mathcal{C}} \frac{\text{Tr} \left(\mathbf{U}_{\mathcal{F}}^H \text{diag}(\mathbf{p}) \mathbf{C}_v \mathbf{U}_{\mathcal{F}} \right)}{\lambda_{\min} \left(\mathbf{U}_{\mathcal{F}}^H \text{diag}(\mathbf{p}) \mathbf{U}_{\mathcal{F}} \right)}. \quad (22)$$

Problem (22) is a convex/concave fractional program [36], i.e., a problem that involves the minimization of the ratio of a convex function over a concave function, both defined over the convex set \mathcal{C} . As previously mentioned, since both functions at numerator and denominator of (17) are differentiable and positive for all $\mathbf{p} \in \mathcal{C}$, the objective function of (22) is pseudo-convex in \mathcal{C} [33]. As a consequence, any local minimum of problem (22) is also a global minimum, and any descent method can be used to find such optimal solutions [36].

To solve problem (22), in this paper we consider a method based on the Dinkelbach algorithm [37], which converts the fractional problem (22) into the iterative solution of a sequence of parametric problems as:

$$\min_{\mathbf{p} \in \mathcal{C}} h(\mathbf{p}, \omega) = f(\mathbf{p}) - \omega g(\mathbf{p}) \quad (23)$$

with ω denoting the free parameter to be selected by the algorithm, and

$$f(\mathbf{p}) = \text{Tr} \left(\mathbf{U}_{\mathcal{F}}^H \text{diag}(\mathbf{p}) \mathbf{C}_v \mathbf{U}_{\mathcal{F}} \right), \quad (24)$$

$$g(\mathbf{p}) = \lambda_{\min} \left(\mathbf{U}_{\mathcal{F}}^H \text{diag}(\mathbf{p}) \mathbf{U}_{\mathcal{F}} \right). \quad (25)$$

Letting $\omega(\mathbf{p}) = f(\mathbf{p})/g(\mathbf{p})$, and noting that $h(\mathbf{p}^*, \omega(\mathbf{p}^*)) = 0$ at the optimal value \mathbf{p}^* , the Dinkelbach method proceeds as described in Algorithm 3, and is guaranteed to converge to

Algorithm 4: SCA Method for Problem (21).

Set $k = 1$. Start with $\mathbf{p}[1] \in \mathcal{C}$. Then, for $k \geq 1$, repeat until convergence the following steps:

$$\text{S.1) } \hat{\mathbf{p}}[k] = \arg \min_{\mathbf{p} \in \mathcal{C}} \widetilde{\text{MSD}}(\mathbf{p}; \mathbf{p}[k])$$

$$\text{S.2) } \mathbf{p}[k+1] = \mathbf{p}[k] + \varepsilon[k] \left(\hat{\mathbf{p}}[k] - \mathbf{p}[k] \right)$$

global optimal solutions of the approximated problem (22), see, e.g., [36], [37].

The second approach aims at finding local optimal solutions of (21) using an SCA method with provable convergence guarantees. Following the approach proposed in [35], and letting $\mathbf{p}[k]$ be the guess of the probability vector at iteration k , the SCA algorithm proceeds as described in Algorithm 4. More formally, the updating scheme reads: at every iteration k , given the current estimate $\mathbf{p}[k]$, the first step of Algorithm 4 solves a surrogate optimization problem involving the function $\widetilde{\text{MSD}}(\mathbf{p}; \mathbf{p}[k])$, which represents a strongly convex approximant of MSD in (14) at the point $\mathbf{p}[k]$ that satisfies the following conditions:

- F1) $\widetilde{\text{MSD}}(\bullet; \mathbf{z})$ is uniformly strongly convex on $\mathcal{C} \forall \mathbf{z} \in \mathcal{C}$;
- F2) $\nabla_{\mathbf{p}} \widetilde{\text{MSD}}(\mathbf{z}; \mathbf{z}) = \nabla_{\mathbf{p}} \text{MSD}(\mathbf{z})$ for all $\mathbf{z} \in \mathcal{C}$;
- F3) $\nabla_{\mathbf{p}} \widetilde{\text{MSD}}(\bullet; \bullet)$ is continuous on \mathcal{C} .

Among all the possible choices satisfying the requirements F1-F3 (see [35] for details), in this paper we consider the following surrogate function:

$$\begin{aligned} \widetilde{\text{MSD}}(\mathbf{p}; \mathbf{p}[k]) &= \frac{\tau}{2} \|\mathbf{p} - \mathbf{p}[k]\|^2 \\ &+ \frac{\mu}{2} \text{Tr} \left[\left(\mathbf{U}_{\mathcal{F}}^H \text{diag}(\mathbf{p}[k]) \mathbf{U}_{\mathcal{F}} \right)^{-1} \mathbf{U}_{\mathcal{F}}^H \text{diag}(\mathbf{p}) \mathbf{C}_v \mathbf{U}_{\mathcal{F}} \right] \\ &+ \frac{\mu}{2} \text{Tr} \left[\left(\mathbf{U}_{\mathcal{F}}^H \text{diag}(\mathbf{p}) \mathbf{U}_{\mathcal{F}} \right)^{-1} \mathbf{U}_{\mathcal{F}}^H \text{diag}(\mathbf{p}[k]) \mathbf{C}_v \mathbf{U}_{\mathcal{F}} \right] \end{aligned} \quad (26)$$

with $\tau > 0$, which satisfies F1-F3 and preserves much of the convexity hidden in the original function (14). Then, the second step of Algorithm 4 generates the new point $\mathbf{p}[k+1]$ as a convex combination of the current estimate $\mathbf{p}[k]$ and the solutions $\hat{\mathbf{p}}[k]$ of the surrogate problem, exploiting the step-size sequence $\varepsilon[k]$. Under conditions (20) on the step-size sequence $\varepsilon[k]$, and using (26), the sequence generated by Algorithm 4 converges to a local optimal solution of (21), see [35, Theorem 2] for details.

IV. RECURSIVE LEAST SQUARES ESTIMATION OF GRAPH SIGNALS

As is well known, LMS strategies have low complexity, but typically suffer of slow convergence rate. To improve the learning rate of the adaptive estimation task, we can employ an RLS method, which in turn has a larger computational burden, see, e.g. [29]. In particular, the optimal (centralized) RLS estimate for \mathbf{s}^o at time n , say, $\hat{\mathbf{s}}_c[n]$, can be found as the vector that solves

Algorithm 5: RLS on Graphs.

Start with random $\boldsymbol{\psi}[0]$, and $\boldsymbol{\Psi}[0] = \mathbf{\Pi}$. For $n > 0$, repeat:

$$\begin{aligned}\boldsymbol{\Psi}[n] &= \beta \boldsymbol{\Psi}[n-1] + \mathbf{U}_{\mathcal{F}}^H \mathbf{D}_{\mathcal{S}[n]} \mathbf{C}_v^{-1} \mathbf{U}_{\mathcal{F}} \\ \boldsymbol{\psi}[n] &= \beta \boldsymbol{\psi}[n-1] + \mathbf{U}_{\mathcal{F}}^H \mathbf{D}_{\mathcal{S}[n]} \mathbf{C}_v^{-1} \mathbf{y}[n] \\ \hat{\mathbf{x}}_c[n] &= \mathbf{U}_{\mathcal{F}} \boldsymbol{\Psi}^{-1}[n] \boldsymbol{\psi}[n]\end{aligned}$$

the following optimization problem:

$$\min_{\mathbf{s}} \sum_{l=1}^n \beta^{n-l} \|\mathbf{D}_{\mathcal{S}[l]}(\mathbf{y}[l] - \mathbf{U}_{\mathcal{F}} \mathbf{s})\|_{\mathbf{C}_v^{-1}}^2 + \beta^n \|\mathbf{s}\|_{\mathbf{\Pi}}^2 \quad (27)$$

where $0 \ll \beta \leq 1$ is the exponential forgetting factor, $\mathbf{\Pi} \succeq \mathbf{0}$ is a regularization matrix, and we have exploited the fact that $\mathbf{D}_{\mathcal{S}[n]}$ is an idempotent matrix for all n . Typically, $\mathbf{\Pi} = \delta \mathbf{I}$, where $\delta > 0$ is small [29]. Solving (27) and using (4), the optimal estimate for \mathbf{x}^o at time n is given by:

$$\hat{\mathbf{x}}_c[n] = \mathbf{U}_{\mathcal{F}} \hat{\mathbf{s}}_c[n] = \mathbf{U}_{\mathcal{F}} \boldsymbol{\Psi}^{-1}[n] \boldsymbol{\psi}[n] \quad (28)$$

where

$$\boldsymbol{\Psi}[n] = \sum_{l=1}^n \beta^{n-l} \mathbf{U}_{\mathcal{F}}^H \mathbf{D}_{\mathcal{S}[l]} \mathbf{C}_v^{-1} \mathbf{U}_{\mathcal{F}} + \beta^n \mathbf{\Pi}, \quad (29)$$

$$\boldsymbol{\psi}[n] = \sum_{l=1}^n \beta^{n-l} \mathbf{U}_{\mathcal{F}}^H \mathbf{D}_{\mathcal{S}[l]} \mathbf{C}_v^{-1} \mathbf{y}[l]. \quad (30)$$

The regularization term in (29) avoids invertibility issues in (28), especially at early values of n . Given the structure of the recursion of $\boldsymbol{\Psi}[n]$ and $\boldsymbol{\psi}[n]$ in (29)–(30), we obtain

$$\boldsymbol{\Psi}[n] = \beta \boldsymbol{\Psi}[n-1] + \mathbf{U}_{\mathcal{F}}^H \mathbf{D}_{\mathcal{S}[n]} \mathbf{C}_v^{-1} \mathbf{U}_{\mathcal{F}}, \quad (31)$$

$$\boldsymbol{\psi}[n] = \beta \boldsymbol{\psi}[n-1] + \mathbf{U}_{\mathcal{F}}^H \mathbf{D}_{\mathcal{S}[n]} \mathbf{C}_v^{-1} \mathbf{y}[n], \quad (32)$$

with $\boldsymbol{\Psi}[0] = \mathbf{\Pi}$, which recursively update both $\boldsymbol{\Psi}[n]$ and $\boldsymbol{\psi}[n]$ given their previous values. Thus, the RLS algorithm for graph signals evolves as illustrated in Algorithm 5, which has computational complexity of the order of $O(|\mathcal{F}|^3)$, due to the presence of the inverse operation $\boldsymbol{\Psi}^{-1}[n]$. Since typically $|\mathcal{F}| \ll N$, the cost $O(|\mathcal{F}|^3)$ is often affordable.⁴ The properties of the RLS algorithm strongly depend on the random sampling operator $\mathbf{D}_{\mathcal{S}[n]}$. Thus, we will show how the design of the random sampling strategy affects the reconstruction capability and the steady-state performance of Algorithm 5. Since the study of the mean-square performance of RLS adaptive filters is challenging [29], in the sequel we will exploit the following assumption in order to make the analysis tractable.⁵

Assumption 4 (Ergodicity): $\exists n_0$ such that, for all $n > n_0$, the time average $\boldsymbol{\Psi}[n]$ in (29) can be replaced by its expected value, i.e., $\boldsymbol{\Psi}[n] = \mathbb{E} \boldsymbol{\Psi}[n]$, for $n > n_0$. ■

⁴A sequential version of the RLS on graphs can be readily derived using the matrix inversion lemma, and leading to a complexity equal to $O(|\mathcal{F}|^2)$.

⁵This is a common assumption in the analysis of RLS-type algorithms, see, e.g., [29], and yields good results in practice.

A. Adaptive Reconstruction From Probabilistic Sampling

Under Assumption 4, the steady state behavior of matrix $\boldsymbol{\Psi}[n]$ in (29) can be approximated as:

$$\begin{aligned}\bar{\boldsymbol{\Psi}} &= \lim_{n \rightarrow \infty} \boldsymbol{\Psi}[n] = \lim_{n \rightarrow \infty} \mathbb{E} \boldsymbol{\Psi}[n] \\ &= \frac{1}{1-\beta} \mathbf{U}_{\mathcal{F}}^H \text{diag}(\mathbf{p}) \mathbf{C}_v^{-1} \mathbf{U}_{\mathcal{F}}\end{aligned} \quad (33)$$

where $\mathbf{p} = (p_1, \dots, p_N)^T \in [0, 1]^N$ represents the sampling probability vector. Thus, from (28) and (33), we deduce that asymptotic reconstruction of \mathbf{x}^o necessarily requests the positive (semi-) definite matrix $\mathbf{U}_{\mathcal{F}}^H \text{diag}(\mathbf{p}) \mathbf{C}_v^{-1} \mathbf{U}_{\mathcal{F}}$ to be invertible (or full rank), which is similar to condition (9) in Sec. III-A for LMS reconstruction. Thus, proceeding as in Sec. III-A, under the assumption that the observation noise is spatially uncorrelated, i.e., matrix \mathbf{C}_v is diagonal, it is easy to show that condition (11) is necessary and sufficient to guarantee adaptive graph signal reconstruction using Algorithm 5.

B. Mean-Square Performance

In this section, we illustrate the effect of the probability vector \mathbf{p} on the mean-square behavior of Algorithm 5.

Theorem 2: Assume model (5), condition (9), Assumptions 2 and 4 hold. Then, Algorithm 5 is mean-square stable, with mean-square deviation given by

$$\begin{aligned}\text{MSD} &= \lim_{n \rightarrow \infty} \mathbb{E} \|\hat{\mathbf{x}}_c[n] - \mathbf{x}^o\|^2 \\ &= \frac{1-\beta}{1+\beta} \text{Tr} \left[\left(\mathbf{U}_{\mathcal{F}}^H \text{diag}(\mathbf{p}) \mathbf{C}_v^{-1} \mathbf{U}_{\mathcal{F}} \right)^{-1} \right].\end{aligned} \quad (34)$$

Proof: See Appendix B. ■

C. Optimal Sampling Strategies

Exploiting the results obtained in Sec. IV-B, the goal of this section is to develop optimal sampling strategies aimed at selecting the probability vector \mathbf{p} to optimize the performance of the RLS on Graphs. In particular, the method leads to optimal sampling strategies aimed at selecting a probability vector \mathbf{p} that minimizes the total sampling rate, while guaranteeing a target value of MSD. To this aim, we formulate the following optimization problem:

$$\begin{aligned}\min_{\mathbf{p}} \quad & \mathbf{1}^T \mathbf{p} \\ \text{s.t.} \quad & \text{Tr} \left[\left(\mathbf{U}_{\mathcal{F}}^H \text{diag}(\mathbf{p}) \mathbf{C}_v^{-1} \mathbf{U}_{\mathcal{F}} \right)^{-1} \right] \leq \gamma \frac{1+\beta}{1-\beta} \\ & \mathbf{0} \leq \mathbf{p} \leq \mathbf{p}^{\max}\end{aligned} \quad (35)$$

The linear objective function in (35) represents the total graph sampling rate that has to be minimized. From (34), the first constraint in (35) imposes that the MSD must be less than or equal to a constant $\gamma > 0$ [this constraint implicitly guarantees the reconstruction condition in (9)]. As before, the last constraint limits the vector to lie in the box $p_i \in [0, p_i^{\max}]$, for all i . It is easy to check the convexity of problem (35), whose global solution can be found using efficient algorithms [34]. Obviously, one can

consider also the related problem where we aim at minimizing the MSD in (34) while imposing a maximum budget on the total graph sampling rate, which also translates into a convex optimization program.

D. Distributed Adaptive Implementation

In many practical systems, data are collected in a distributed network, and sharing local information with a central processor might be either unfeasible or not efficient, owing to the large volume of data, time-varying network topology, and/or privacy issues [38], [39]. Motivated by these observations, in this section we extend the RLS strategy in Algorithm 5 to a distributed setting, where the nodes of the graph are connected through a sparse communication network described by the graph $\mathcal{G}_c = (\mathcal{V}, \mathcal{E}_c)$. The topology of the communication graph \mathcal{G}_c does not necessarily coincide with that of the graph used to process the data, i.e., \mathcal{G} . To ensure the diffusion of information over the network, we assume the following.

Assumption 5 (Topology): The communication graph \mathcal{G}_c is symmetric and connected. ■

To derive distributed solution methods for problem (27), let us introduce local copies $\{\mathbf{s}_i\}_{i=1}^N$ of the global variable \mathbf{s} , and recast problem (27) in the following equivalent form:

$$\begin{aligned} \min_{\{\mathbf{s}_i\}_{i=1}^N} & \sum_{i=1}^N \sum_{l=1}^n \beta^{n-l} \frac{d_i[l]}{\sigma_i^2} (y_i[l] - \mathbf{u}_{\mathcal{F},i}^H \mathbf{s}_i)^2 + \frac{\beta^n}{N} \sum_{i=1}^N \|\mathbf{s}_i\|_{\mathbf{\Pi}}^2 \\ \text{subject to} & \quad \mathbf{s}_i = \mathbf{s}_j \quad \text{for all } i \in \mathcal{V}, j \in \mathcal{N}_i, \end{aligned} \quad (36)$$

where \mathcal{N}_i denotes the set of local neighbors of agent i on the communication graph \mathcal{G}_c , and $\mathbf{u}_{\mathcal{F},i}^H$ is the i -th row of matrix $\mathbf{U}_{\mathcal{F}}$. Now, letting $\mathbf{s} = \{\mathbf{s}_i\}_{i=1}^N$ and $\boldsymbol{\lambda} = \{\boldsymbol{\lambda}_{ij}\}_{i \in \mathcal{V}, j \in \mathcal{N}_i}$, the augmented Lagrangian for problem (36) writes as:

$$\begin{aligned} \mathcal{L}_a(\mathbf{s}, \boldsymbol{\lambda}) &= \sum_{i=1}^N \sum_{l=1}^n \beta^{n-l} \frac{d_i[l]}{\sigma_i^2} (y_i[l] - \mathbf{u}_{\mathcal{F},i}^H \mathbf{s}_i)^2 \\ &+ \frac{\beta^n}{N} \sum_{i=1}^N \|\mathbf{s}_i\|_{\mathbf{\Pi}}^2 + \sum_{i=1}^N \sum_{j \in \mathcal{N}_i} \boldsymbol{\lambda}_{ij}^T (\mathbf{s}_i - \mathbf{s}_j) \\ &+ \frac{\varrho}{4} \sum_{i=1}^N \sum_{j \in \mathcal{N}_i} \|\mathbf{s}_i - \mathbf{s}_j\|^2 \end{aligned} \quad (37)$$

where $\varrho > 0$ is a positive coefficient. Since the augmented Lagrangian function in (37) is strongly convex for all n , we can employ ADMM to solve problem (36), see, e.g., [28]. The first step of ADMM aims at minimizing the augmented Lagrangian in (37) with respect to the primal variable \mathbf{s} . In particular, we apply a parallel method where each node minimizes (37) with respect to its local variable, while keeping fixed the variables of its neighbors at the previous iteration. The local variables at each node can then be updated as the solution of the following

Algorithm 6: DRLS on Graphs.

Start with $\{\boldsymbol{\psi}_i[0]\}_{i=1}^N$, $\{\mathbf{s}_i[0,0]\}_{i=1}^N$, $\{\boldsymbol{\lambda}_{ij}[0,0]\}_{i \in \mathcal{V}, j \in \mathcal{N}_i}^N$ chosen at random, and set $\{\boldsymbol{\Psi}_i[0]\}_{i=1}^N = \mathbf{\Pi}/N$, and $\varrho \in [0, \varrho_{\max}]$.

for $n > 0$ **do**

All $i \in \mathcal{V}$: update $\boldsymbol{\Psi}_i[n]$ and $\boldsymbol{\psi}_i[n]$ using (40) and (41);

for $k = 0, \dots, K-1$ **do**

All $i \in \mathcal{V}$: transmit $\boldsymbol{\lambda}_{ij}[n, k]$ to each $j \in \mathcal{N}_i$;

All $i \in \mathcal{V}$: update $\widehat{\mathbf{s}}_i[n, k+1]$ using (39);

All $i \in \mathcal{V}$: transmit $\widehat{\mathbf{s}}_i[n, k+1]$ to each $j \in \mathcal{N}_i$;

All $i \in \mathcal{V}$: update $\{\boldsymbol{\lambda}_{ij}[n, k+1]\}_{j \in \mathcal{N}_i}$ using (42);

end

end

optimization problems:

$$\begin{aligned} \widehat{\mathbf{s}}_i[n, k+1] &= \underset{\mathbf{s}_i}{\operatorname{argmin}} \sum_{l=1}^n \beta^{n-l} \frac{d_i[l]}{\sigma_i^2} (y_i[l] - \mathbf{u}_{\mathcal{F},i}^H \mathbf{s}_i)^2 \\ &+ \frac{\beta^n}{N} \|\mathbf{s}_i\|_{\mathbf{\Pi}}^2 + \sum_{j \in \mathcal{N}_i} (\boldsymbol{\lambda}_{ij}[n, k] - \boldsymbol{\lambda}_{ji}[n, k])^T \mathbf{s}_i \\ &+ \frac{\varrho}{2} \sum_{j \in \mathcal{N}_i} \|\mathbf{s}_i - \widehat{\mathbf{s}}_j[n, k]\|^2, \quad i = 1, \dots, N. \end{aligned} \quad (38)$$

Each local subproblem in (38) corresponds to an unconstrained minimization that admits closed-form solution given by:

$$\begin{aligned} \widehat{\mathbf{s}}_i[n, k+1] &= \left(\boldsymbol{\Psi}_i[n] + \varrho |\mathcal{N}_i| \mathbf{I} \right)^{-1} \left[\boldsymbol{\psi}_i[n] + \varrho \sum_{j \in \mathcal{N}_i} \widehat{\mathbf{s}}_j[n, k] \right. \\ &\quad \left. - \frac{1}{2} \sum_{j \in \mathcal{N}_i} (\boldsymbol{\lambda}_{ij}[n, k] - \boldsymbol{\lambda}_{ji}[n, k]) \right], \end{aligned} \quad (39)$$

for $i = 1, \dots, N$, where, setting $\boldsymbol{\Psi}_i[0] = \mathbf{\Pi}/N$, we have

$$\boldsymbol{\Psi}_i[n] = \beta \boldsymbol{\Psi}_i[n-1] + d_i[n] \mathbf{u}_{\mathcal{F},i} \mathbf{u}_{\mathcal{F},i}^H / \sigma_i^2, \quad (40)$$

$$\boldsymbol{\psi}_i[n] = \beta \boldsymbol{\psi}_i[n-1] + d_i[n] y_i[n] \mathbf{u}_{\mathcal{F},i} / \sigma_i^2. \quad (41)$$

Finally, the second step of the ADMM algorithm updates the Lagrange multipliers as:

$$\boldsymbol{\lambda}_{ij}[n, k+1] = \boldsymbol{\lambda}_{ij}[n, k] + \frac{\varrho}{2} \left(\widehat{\mathbf{s}}_j[n, k+1] - \widehat{\mathbf{s}}_i[n, k+1] \right), \quad (42)$$

for $i \in \mathcal{V}$, $j \in \mathcal{N}_i$. Recursions (39) and (42) constitute the ADMM-based distributed RLS algorithm (DRLS), whereby all sensors $i \in \mathcal{V}$ keep track of their local estimate $\widehat{\mathbf{s}}_i$ and their multipliers $\{\boldsymbol{\lambda}_{ij}\}_{j \in \mathcal{N}_i}$, which can be arbitrarily initialized. Then, all the steps of the DRLS strategy for distributed adaptive reconstruction of graph signals are summarized in Algorithm 6. The proposed DRLS method has a double time-scale. The inner loop on index k has the goal of forcing consensus among variables $\widehat{\mathbf{s}}_i[n, k]$. However, when the network is deployed to track a time-varying graph signal, one cannot afford large delays in-between consecutive sensing instants. In this case, we can run a single consensus iteration per acquired observation, i.e., $K = 1$ in Algorithm 6, thus making the method suitable for operation in

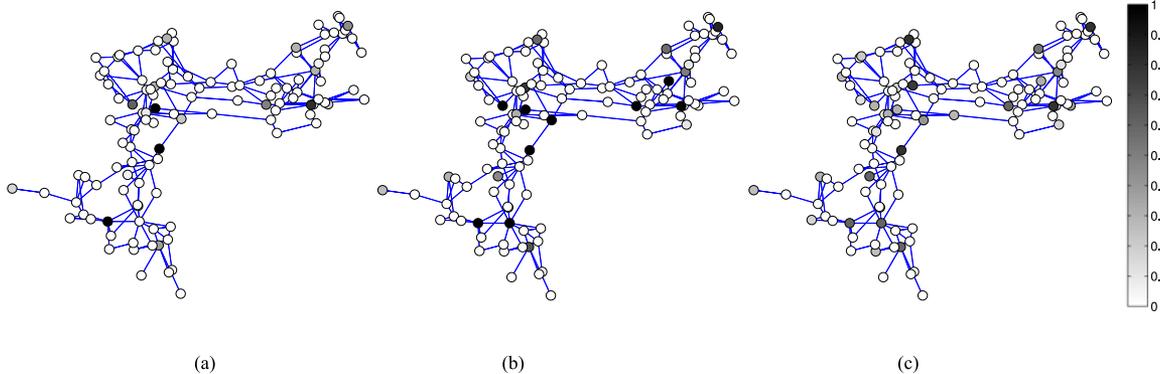


Fig. 1. IEEE 118 bus system: Graph Topology, and optimal probability vectors obtained from (22), for different values of $\bar{\alpha}$.

nonstationary environments. In the sequel, we will validate the proposed distributed approach via numerical simulations.

V. NUMERICAL RESULTS

In this section, we illustrate the performance of the proposed strategies applied to both synthetic and real data.

A. LMS on Graphs

Let us consider an application to the IEEE 118 Bus Test Case, which represents a portion of the American Electric Power System (in the Mid-western US) as of December 1962 [40]. The graph is composed of 118 nodes (i.e., buses), and its topology (i.e., transmission lines connecting buses) is illustrated in Fig. 1. The dynamics of the power generators give rise to smooth graph signals (e.g., powers, currents, voltages), whose spectral content is assumed to be limited to the first ten eigenvectors of the Laplacian matrix of the graph in Fig. 1, i.e., $|\mathcal{F}| = 10$. The observation noise in (5) is zero-mean, Gaussian, with a diagonal covariance matrix where each element is illustrated in Fig. 2 (bottom). The other parameters are: $\mu = 0.1$, and $\mathcal{P} = 118$. Then, in Fig. 2 (A, B, and C), we plot the optimal probability vector obtained using Algorithm 3 for different values of $\bar{\alpha}$ (0.99 for case A, 0.98 for both cases B and C) and upper bound vectors \mathbf{p}^{\max} (1 for cases A and B and as illustrated in the figure for case C). In all cases, we have checked that the constraint on the convergence rate in (21) is attained strictly. From Fig. 2, as expected, we notice how the method enlarges the expected sampling set if we either require (see cases A and B) a faster convergence rate (i.e., a smaller $\bar{\alpha}$), or if stricter bounds are imposed on the maximum probability to sample “important” nodes (see case C). Also, from Fig. 2, it is clear that the method finds a very sparse probability vector and avoids to assign large sampling probabilities to nodes having large noise variances. We remark that with the proposed formulation, sparse sampling patterns are obtained thanks to the optimization of the sampling probabilities, without resorting to any relaxation of complex integer optimization problems. The corresponding positions of the samples collected over the IEEE 118 Bus graph are illustrated in Fig. 1, where the color (in gray scale) of the vertices denotes the value of the sampling probability.

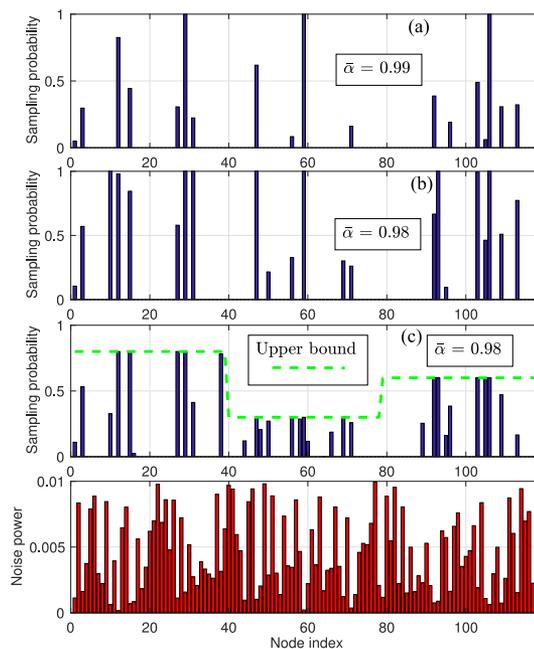


Fig. 2. Optimal probabilities and noise variance, obtained from (22) with Algorithm 3 for different values of $\bar{\alpha}$ and \mathbf{p}^{\max} .

To compare the results obtained using Algorithms 3 and 4, in Fig. 3, we report the temporal evolution of MSD in (14) evaluated for each instantaneous estimate $\mathbf{p}[k]$, obtained using the aforementioned strategies. Both algorithms are initialized such that $p_i[1] = 0.5$ for all i , and we set $\mathbf{p}^{\max} = \mathbf{1}$. The SCA algorithm exploits the surrogate function in (26) with $\tau = 10^{-6}$, and the step size sequence in (27) is chosen to satisfy (20), and obeying to the diminishing rule $\varepsilon[k] = \varepsilon[k-1](1 - \eta\varepsilon[k-1])$, with $\varepsilon[0] = 1$, and $\eta = 0.001$. The surrogate problem in S.1 of Algorithm 4 is solved using the CVX software [34]. As we can see from Fig. 3, both algorithms illustrate a very good convergence behavior, reaching their final state in a few iterations. In particular, we notice that the Algorithm 3 is slightly faster than Algorithm 4. But most importantly, we notice that both methods converge to very similar final solutions. This means that local optimal solutions of the original non-convex problem (21) are very similar to global optimal solutions of the approximated convex/concave fractional problem in (22). This result

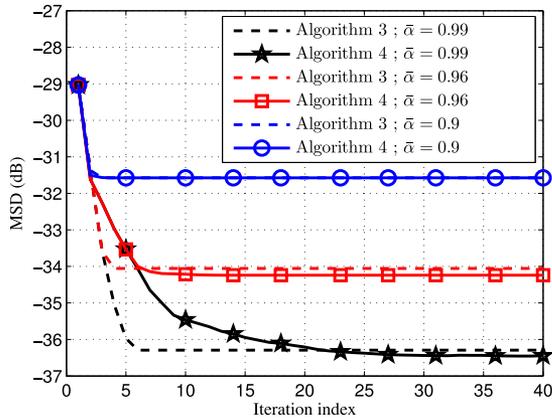


Fig. 3. Behavior of $\text{MSD}(\mathbf{p}[k])$ versus iteration index, obtained using Algorithms 3 and 4.

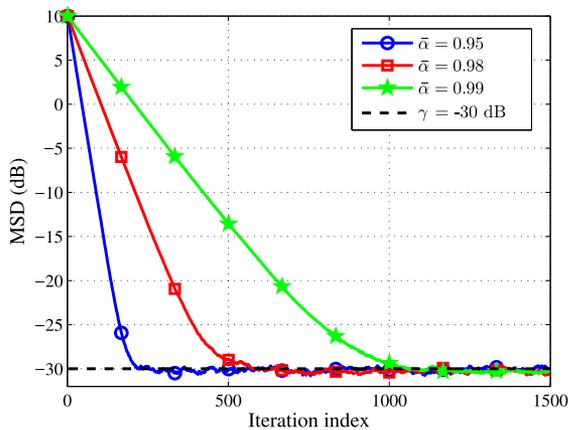


Fig. 4. Learning curve of Algorithm 1, obtained solving (18) for different values of $\bar{\alpha}$.

justifies the approximation made in (17) to formulate the sampling design problem (22) and to derive Algorithm 3, which also guarantees convergence to global optimal solutions.

To validate the theoretical results derived in Sec. III-B, in Fig. 4 we report the learning curve (in terms of MSD) of the LMS on Graphs (cf. Algorithm 1), obtained solving (18) for different values of $\bar{\alpha}$. In particular, we set a target performance on the MSD given by $\gamma = -30$ dB. The other parameters are: $\mu = 0.1$, $\mathbf{p}^{\max} = \mathbf{1}$, and $\mathcal{P} = 118$. The curves are averaged over 100 independent simulations. As we can see from Fig. 4, thanks to the effect of the sampling strategy (18), the LMS algorithm can increase its convergence rate (reducing the value of $\bar{\alpha}$), while always guaranteeing the performance requirement on the MSD. This curve also confirms the theoretical analysis derived in Sec. III-B, and further justifies the approximation of the MSD function made in (17).

Finally, we compare the sampling strategy in (18) with some established sampling methods for graph signals, namely, the leverage score sampling from [22], the Max-Det greedy strategy from [13], and the (uniformly) random sampling. For each strategy, we keep adding nodes to the sampling set according to the corresponding criterion until the constraints on the convergence rate and the MSD in (18) are satisfied. Then, in Fig. 5, we

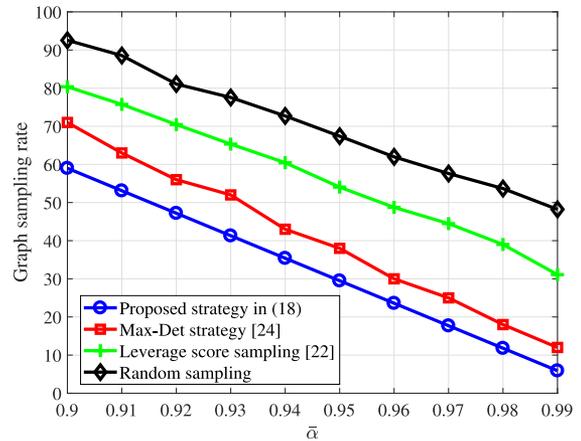


Fig. 5. Graph sampling rate versus convergence rate $\bar{\alpha}$, for different sampling strategies.

report the behavior of the total graph sampling rate $\mathbf{1}^T \mathbf{p}$ versus the parameter $\bar{\alpha}$ in (18), obtained using the four aforementioned strategies. The other parameters are: $\mu = 0.1$, $\mathbf{p}^{\max} = \mathbf{1}$, and $\gamma = -25$ dB. The results for the random sampling strategies are averaged over 200 independent simulations. As expected, from Fig. 5, we notice how the graph sampling rate increases for lower values of $\bar{\alpha}$, i.e., increasing the convergence rate of the algorithm, for all strategies. Furthermore, we can notice the large gain in terms of reduced sampling rate $\mathbf{1}^T \mathbf{p}$ obtained by the proposed strategy with respect to other methods available in the literature.

B. RLS on Graphs

Let us consider the same setting as before. An example of optimal probabilistic sampling, obtained solving problem (35), with $\beta = 0.95$, is illustrated in Fig. 6, for three different values of γ . As we can notice from Fig. 6, the method finds a very sparse probability vector in order to guarantee the performance requirement on the MSD. In all cases, we have checked that the constraint on the MSD is attained strictly. As expected, from Fig. 6, we notice how the proposed method enlarges the expected sampling set if we require a stricter requirement on the MSD, i.e., a lower value of γ , and avoids to select nodes having large noise variance (at least at low values of γ).

To validate the theoretical results derived in Sec. IV-B, in Fig. 7 we report the learning curve (in terms of MSD) of the RLS on graphs (cf. Algorithm 5), with probability vector \mathbf{p} obtained by solving Problem (35) for three different values of γ (the same as in the previous example). The curves are averaged over 200 independent simulations. The theoretical expression of MSD in (34) is also reported for comparison purposes, for all values of γ . As we can see from Fig. 7, simulations match well the theoretical results.

Finally, in Fig. 8, we report the temporal evolution of the MSD obtained using four different algorithms: the proposed LMS on Graphs in Algorithm 1; the RLS on Graphs in Algorithm 5; the Diffusion LMS on Graphs from [25]; and the DRLS on Graphs strategy in Algorithm 6, considering different numbers

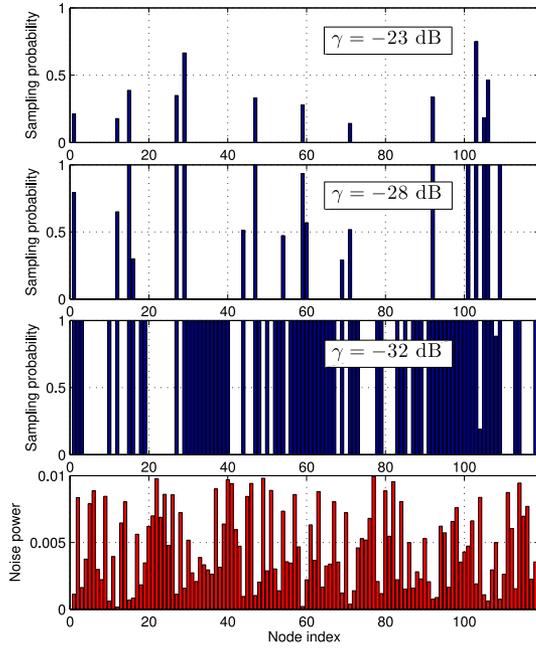


Fig. 6. Optimal sampling probability vector, and noise variance, versus node index, for different values of γ .

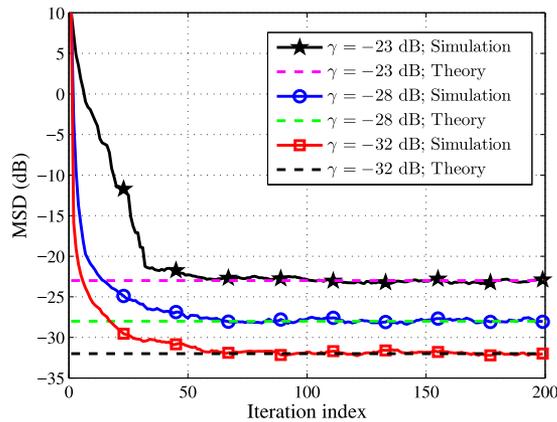


Fig. 7. MSD versus iteration index n , obtained using Algorithm 5 and the Sampling Strategy in (35).

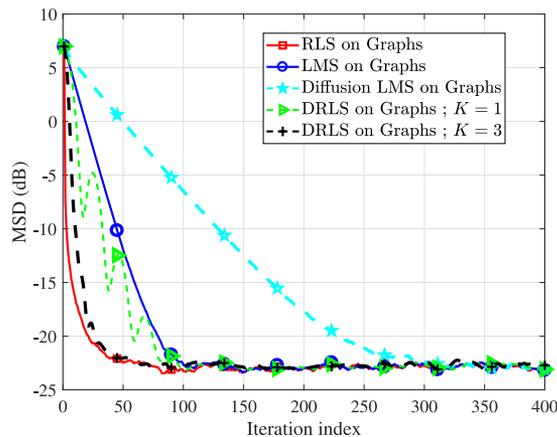


Fig. 8. MSD versus iteration index n , for different algorithms.

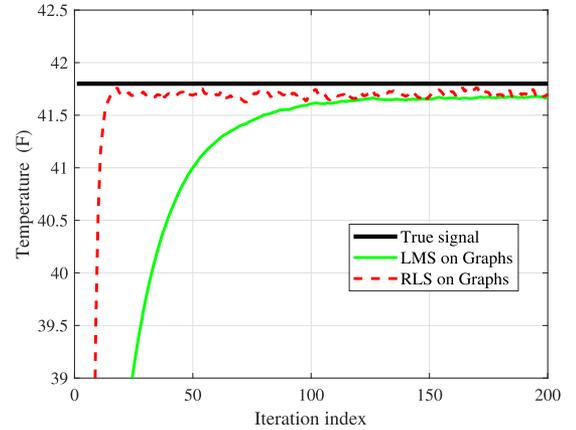


Fig. 9. True temperature and estimates at a randomly picked unobserved station, for different algorithms.

K of inner consensus iterations, i.e., $K = 1$ and $K = 3$. We consider a graph composed of $N = 20$ nodes, whose topology is obtained from a random geometric graph model, and having algebraic connectivity equal to 0.82. The graph signal is such that $|\mathcal{F}| = 5$. The sampling strategy of the RLS (and of the DRLS) was designed solving problem (35), where $\gamma = -23$ dB, $\beta = 0.95$, and $\mathbf{p}^{\max} = \mathbf{1}$. The sampling strategy of the LMS (and of the Diffusion LMS) was obtained solving problem (22), with \mathcal{P} given by the optimal sampling rate obtained by the RLS strategy in (35), $\mathbf{p}^{\max} = \mathbf{1}$, $\mu = 0.1$, and $\bar{\alpha}$ chosen to match the MSD steady-state performance of the RLS method. The communication graph is chosen equal to the processing graph. As we can see from Fig. 8, the RLS strategy is faster than the LMS on Graphs, at the cost of a higher complexity. Also, the proposed DRLS significantly outperforms the Diffusion LMS on graphs proposed in [25]. Finally, increasing the number K of inner consensus iterations, the behavior of the DRLS algorithm approaches the performance of the centralized RLS, at the cost of a larger number of exchanged parameters over the network.

C. Application to Real Data

The first data set collects temperature measurements at $N = 109$ stations across the United States in 2010 [41]. A similarity graph was built using a 7 nearest neighbors approach [26], which relies on geographical distances. The graph signal at each vertex i represents the temperature observed at the i -th station. In first place, we test the performance of the proposed sampling and recovery strategies in a stationary condition. To this aim, we consider a fixed graph signal (the first one recorded in the dataset) and, in Fig. 9, we illustrate the true value of the temperature measured at an unobserved station chosen at random, along with its estimates carried out by the LMS on Graphs in Algorithm 1 (with $\mu = 1.5$ and $|\mathcal{F}| = 40$), and the RLS on Graphs (with $\beta = 0.5$ and $|\mathcal{F}| = 40$) in Algorithm 5. For a fair comparison among the algorithms, the sampling strategy was chosen equal for both of them. In particular, we consider the probabilistic sampling strategy in (35), where $\gamma = 0.1$, and the noise covariance matrix was chosen diagonal, and such that each element was randomly selected in the interval $[0, 0.01]$.

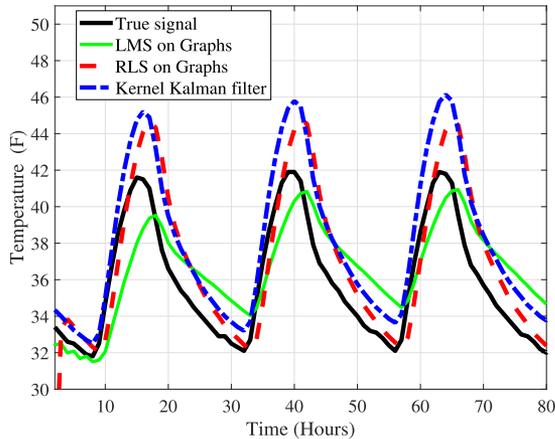


Fig. 10. True temperature and estimates across time at a randomly picked unobserved station.

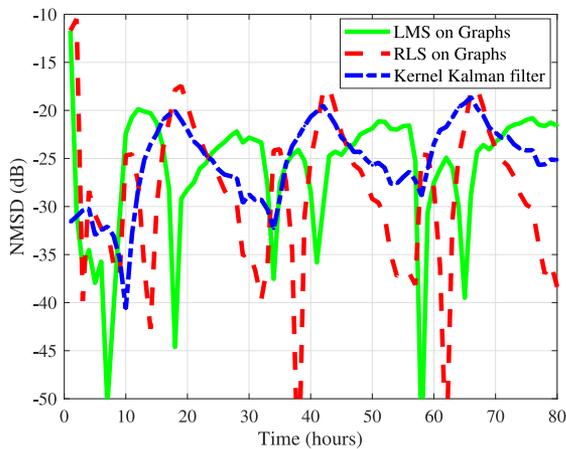


Fig. 11. NMSD versus time, for different algorithms.

As a result, the method selects a sparse probability vector such that the expected sampling set is composed of 42 nodes. As we can notice from Fig. 9, both algorithms are capable to produce a sufficiently accurate estimate of the temperature at the unobserved station, with a small bias of the order of fractions of degree, which is due to the non perfect bandlimitedness of the real graph signal considered in this experiment.

In second place, we aim to test the tracking capabilities of our proposed strategies. To this aim, in Fig. 10, we illustrate the true behavior of the temperature measured at an unobserved station chosen at random, over the first 80 hours of 2010, along with its online estimates carried out using three different algorithms: the LMS on Graphs in Algorithm 1 (with $\mu = 1.5$ and $|\mathcal{F}| = 40$), the RLS on Graphs (with $\beta = 0.5$ and $|\mathcal{F}| = 40$) in Algorithm 5, and the Kernel Kalman Filter (KKF) from [26] (using the same settings as in [26]). For a fair comparison among the recovery strategies, the sampling set is chosen equal for all algorithms and fixed over time, according to the Max-Det strategy in [13], selecting 50 samples. As we can notice from Fig. 10, the proposed LMS and RLS on graphs show good tracking performance that are comparable with KKF, while having a lower computational complexity. This can be noticed also from Fig. 11, which illustrates the temporal behavior of the normalized MSD (NMSD),

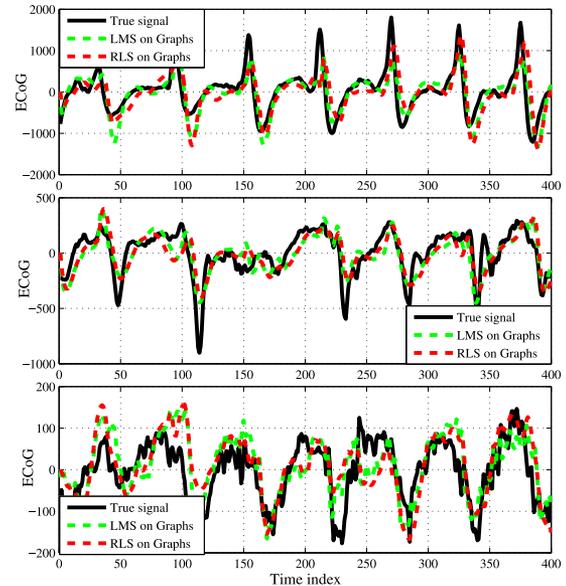


Fig. 12. True ECoG and estimates across time at three randomly selected unobserved regions of the brain.

i.e., $\text{NMSD}_i[n] = (\hat{x}_i[n] - x_i^o[n])^2 / (x_i^o[n])^2$, obtained by the three aforementioned algorithms at the same unobserved node previously considered in Fig. 10.

The second dataset presents test results on Electroencephalography (ECoG), captured through experiments conducted in an epilepsy study [42]. Data were collected over a period of five days, where the electrodes recorded 76 ECoG time series, consisting of voltage levels measured in different regions of the brain (see [42] for further details). The GFT matrix $\mathbf{U}_{\mathcal{F}}$ in (4) is learnt from the first 200 samples of ictal data, using the method proposed in [43], and imposing a bandwidth equal to $|\mathcal{F}| = 30$. In Fig. 12, we illustrate the true behavior of the ECoG present at three unobserved electrodes chosen at random, over the first 400 samples of ictal data, along with estimates carried out using two different algorithms: the LMS on Graphs in Algorithm 1 (with $\mu = 1.5$), and the RLS on Graphs (with $\beta = 0.7$) in Algorithm 5. As before, the sampling set is chosen equal for all algorithms and fixed over time, according to the Max-Det strategy in [13], selecting 32 samples over the graph. As we can notice from Fig. 12, both methods are capable to efficiently infer and track the unknown dynamics of ECoG data at unobserved regions of the brain.

VI. CONCLUSION

In this paper, we have introduced LMS and RLS strategies for adaptive learning of graph signals based on a probabilistic sampling mechanism over the graph. A mean-square analysis sheds light on how the random sampling strategy affects the performance of the proposed methods, and paved the way to the formulation of several criteria aimed at designing the sampling probabilities as an optimal trade-off between graph sampling rate, mean-square performance, and learning rate of the algorithms. Finally, a distributed RLS strategy is derived and is shown to be convergent to its centralized counterpart. Numerical

simulations carried out on both synthetic and real data illustrate the good performance of the proposed methods for (possibly distributed) adaptive learning of graph signals.

As a final comment, we would like to remark that the proposed methods can be applied to the adaptive processing of signals residing on an arbitrary subspace not necessarily related to a graph. In this sense, the methods have a broader applicability that is not strictly limited to GSP. Nevertheless, the sampling operation, which lies at the core of this paper, is the aspect that finds a more direct explanation if we think at the useful signal as residing over a graph, i.e., the graph that makes that signal be represented as bandlimited.

APPENDIX A PROOF OF THEOREM 1

Following energy conservation arguments [44], we consider a general weighted squared error sequence $\|\tilde{\mathbf{s}}[n]\|_{\Phi}^2 = \tilde{\mathbf{s}}[n]^H \Phi \tilde{\mathbf{s}}[n]$, where $\Phi \in \mathbb{C}^{|\mathcal{F}| \times |\mathcal{F}|}$ is an Hermitian nonnegative-definite matrix that we are free to choose. Of course, since $\|\tilde{\mathbf{s}}[n]\|^2 = \|\tilde{\mathbf{x}}[n]\|^2$ for all n [cf. (4)], it is equivalent to consider the mean-square behavior of $\tilde{\mathbf{s}}[n]$ or $\tilde{\mathbf{x}}[n]$. Then, from (12), exploiting Assumption 2 and the fact that $\mathbb{E}\{\mathbf{D}_{S[n]} \mathbf{v}[n] \mathbf{v}[n]^H \mathbf{D}_{S[n]}\} = \mathbf{P} \mathbf{C}_v$, we can establish:

$$\mathbb{E}\|\tilde{\mathbf{s}}[n+1]\|_{\Phi}^2 = \mathbb{E}\|\tilde{\mathbf{s}}[n]\|_{\Phi}^2 + \mu^2 \text{Tr}(\Phi \mathbf{U}_{\mathcal{F}}^H \mathbf{P} \mathbf{C}_v \mathbf{U}_{\mathcal{F}}) \quad (43)$$

where $\text{Tr}(\cdot)$ denotes the trace operator, matrix $\mathbf{P} = \text{diag}(\mathbf{p}) = \mathbb{E}\{\mathbf{D}_{S[n]}\}$, and

$$\begin{aligned} \Phi' &= \mathbb{E}(\mathbf{I} - \mu \mathbf{U}_{\mathcal{F}}^H \mathbf{D}_{S[n]} \mathbf{U}_{\mathcal{F}}) \Phi (\mathbf{I} - \mu \mathbf{U}_{\mathcal{F}}^H \mathbf{D}_{S[n]} \mathbf{U}_{\mathcal{F}}) \quad (44) \\ &= \Phi - \mu \Phi \mathbf{U}_{\mathcal{F}}^H \mathbf{P} \mathbf{U}_{\mathcal{F}} - \mu \mathbf{U}_{\mathcal{F}}^H \mathbf{P} \mathbf{U}_{\mathcal{F}} \Phi \\ &\quad + \mu^2 \mathbb{E}\{\mathbf{U}_{\mathcal{F}}^H \mathbf{D}_{S[n]} \mathbf{U}_{\mathcal{F}} \Phi \mathbf{U}_{\mathcal{F}}^H \mathbf{D}_{S[n]} \mathbf{U}_{\mathcal{F}}\}. \quad (45) \end{aligned}$$

The last term in (45) can be computed in closed form. Letting $\mathbf{U}_{\mathcal{F}} \mathbf{D}_{S[n]} \mathbf{U}_{\mathcal{F}}^H = \sum_{i=1}^N d_i[n] \mathbf{u}_{\mathcal{F},i} \mathbf{u}_{\mathcal{F},i}^H$, where $\mathbf{u}_{\mathcal{F},i}^H$ is the i -th row of $\mathbf{U}_{\mathcal{F}}$, the last term in (45) is given by:

$$\mathbb{E}\{\mathbf{U}_{\mathcal{F}}^H \mathbf{D}_{S[n]} \mathbf{U}_{\mathcal{F}} \Phi \mathbf{U}_{\mathcal{F}}^H \mathbf{D}_{S[n]} \mathbf{U}_{\mathcal{F}}\} = \sum_{i=1}^N \sum_{j=1}^N m_{i,j}^{(2)} \mathbf{C}_{ij} \quad (46)$$

where $\mathbf{C}_{ij} = \mathbf{u}_{\mathcal{F},i} \mathbf{u}_{\mathcal{F},i}^H \Phi \mathbf{u}_{\mathcal{F},j} \mathbf{u}_{\mathcal{F},j}^H$, and

$$m_{i,j}^{(2)} = \mathbb{E}\{d_i[n] d_j[n]\} = \begin{cases} p_i, & \text{if } i = j; \\ p_i p_j, & \text{if } i \neq j. \end{cases} \quad (47)$$

In the sequel, to study mean-square stability of Algorithm 1, we consider the following approximation:

$$\Phi' = \mathbb{E}(\mathbf{I} - \mu \mathbf{U}_{\mathcal{F}}^H \mathbf{D}_{S[n]} \mathbf{U}_{\mathcal{F}}) \Phi (\mathbf{I} - \mu \mathbf{U}_{\mathcal{F}}^H \mathbf{D}_{S[n]} \mathbf{U}_{\mathcal{F}}) \quad (48)$$

$$\simeq (\mathbf{I} - \mu \mathbf{U}_{\mathcal{F}}^H \mathbf{P} \mathbf{U}_{\mathcal{F}}) \Phi (\mathbf{I} - \mu \mathbf{U}_{\mathcal{F}}^H \mathbf{P} \mathbf{U}_{\mathcal{F}}), \quad (49)$$

which is accurate under Assumption 3, i.e., for small step-sizes.⁶ In particular, it is immediate to see that (49) can be obtained from (45)–(46), by substituting the terms p_i in (47) with p_i^2 , for the case $i = j$. Such approximation appears in (49) only in the term

⁶This kind of approximations are typical when studying long-term dynamics of adaptive filters, see, e.g., [45, Ch.4], and lead to good results in practice.

$O(\mu^2)$ and, under Assumption 3, it is assumed to produce a negligible deviation from (44).

Now, we proceed by showing the stability conditions for recursion (43). Letting $\mathbf{H} = \mathbf{U}_{\mathcal{F}}^H \mathbf{P} \mathbf{U}_{\mathcal{F}}$, $\mathbf{Q} = (\mathbf{I} - \mu \mathbf{H})^2$, and $\Phi = \mathbf{I}$, recursion (43) can be bounded as:

$$\mathbb{E}\|\tilde{\mathbf{s}}[n]\|^2 \leq \mathbb{E}\|\tilde{\mathbf{s}}[0]\|_{\mathbf{Q}^n}^2 + \mu^2 c \sum_{l=0}^n \|\mathbf{Q}\|^l \quad (50)$$

where $c = \text{Tr}(\mathbf{U}_{\mathcal{F}}^H \mathbf{P} \mathbf{C}_v \mathbf{U}_{\mathcal{F}})$. We also have

$$\begin{aligned} \|\mathbf{Q}\| &= \|\mathbf{I} - \mu \mathbf{U}_{\mathcal{F}}^H \mathbf{P} \mathbf{U}_{\mathcal{F}}\|^2 = (\rho(\mathbf{I} - \mu \mathbf{U}_{\mathcal{F}}^H \mathbf{P} \mathbf{U}_{\mathcal{F}}))^2 \\ &\leq \max\{(1 - \mu\delta)^2, (1 - \mu\nu)^2\} \\ &\stackrel{(a)}{\leq} 1 - 2\mu\nu + \mu^2\delta^2 = 1 - 2\mu\nu \left(1 - \frac{\mu}{2\nu}\delta^2\right) \quad (51) \end{aligned}$$

where $\rho(\mathbf{X})$ denotes the spectral radius of matrix \mathbf{X} , $\delta = \lambda_{\max}(\mathbf{U}_{\mathcal{F}}^H \mathbf{P} \mathbf{U}_{\mathcal{F}})$, $\nu = \lambda_{\min}(\mathbf{U}_{\mathcal{F}}^H \mathbf{P} \mathbf{U}_{\mathcal{F}})$, and in (a) we have exploited $\delta \geq \nu$. Taking the limit of (50) as $n \rightarrow \infty$, and since $\|\mathbf{Q}\| < 1$ if conditions (9) [i.e., (11)] and (13) hold, we obtain

$$\lim_{n \rightarrow \infty} \mathbb{E}\|\tilde{\mathbf{s}}[n]\|^2 \leq \frac{\mu^2 c}{1 - \|\mathbf{Q}\|} \stackrel{(a)}{\leq} \frac{\mu c}{2\nu - \mu\delta^2}, \quad (52)$$

where (a) follows from (51). The upper bound (52) does not exceed $\mu c/\nu$ if $0 < \mu < \nu/\delta^2$. Thus, for sufficiently small values of the step-size, i.e., under Assumption 3, it holds

$$\lim_{n \rightarrow \infty} \mathbb{E}\|\tilde{\mathbf{s}}[n]\|^2 = O(\mu). \quad (53)$$

Also, from (50), the transient component of $\mathbb{E}\|\tilde{\mathbf{s}}[n]\|^2$ vanishes as \mathbf{Q}^n , for $n \rightarrow \infty$. Thus, from (51), the convergence rate of Algorithm 1 (i.e., $\|\mathbf{Q}\|$) is well approximated by (15) when $\mu \ll 2\nu/\delta^2$. Finally, to derive (14), we recast (43) as:

$$\begin{aligned} \mathbb{E}\|\tilde{\mathbf{s}}[n+1]\|_{\Phi}^2 &= \mathbb{E}\|\tilde{\mathbf{s}}[n]\|_{\Phi}^2 - \mu \mathbb{E}\|\tilde{\mathbf{s}}[n]\|_{\mathbf{H}\Phi + \Phi\mathbf{H}}^2 \\ &\quad + \mu^2 \mathbb{E}\|\tilde{\mathbf{s}}[n]\|_{\mathbf{H}\Phi\mathbf{H}}^2 + \mu^2 \text{Tr}(\Phi \mathbf{U}_{\mathcal{F}}^H \mathbf{P} \mathbf{C}_v \mathbf{U}_{\mathcal{F}}) \quad (54) \end{aligned}$$

Taking the limit of (54) as $n \rightarrow \infty$ (assuming that convergence conditions are satisfied), since $\lim_{n \rightarrow \infty} \mathbb{E}\|\tilde{\mathbf{s}}[n+1]\|^2 = \lim_{n \rightarrow \infty} \mathbb{E}\|\tilde{\mathbf{s}}[n]\|^2$ and $\lim_{n \rightarrow \infty} \mathbb{E}\|\tilde{\mathbf{s}}[n]\|_{\mathbf{H}\Phi\mathbf{H}}^2 = O(\mu)$ [cf. (53)], we obtain the following expression:

$$\lim_{n \rightarrow \infty} \mathbb{E}\|\tilde{\mathbf{s}}[n]\|_{\mathbf{H}\Phi + \Phi\mathbf{H}}^2 = \mu \text{Tr}(\Phi \mathbf{U}_{\mathcal{F}}^H \mathbf{P} \mathbf{C}_v \mathbf{U}_{\mathcal{F}}) + O(\mu^2) \quad (55)$$

Using now $\Phi = \mathbf{H}^{-1}$ in (55), we obtain (14).

APPENDIX B PROOF OF THEOREM 2

From (33), (29), and (30), we obtain:

$$\begin{aligned} \hat{\mathbf{x}}_c[n] &= \mathbf{U}_{\mathcal{F}} \left(\sum_{l=1}^n \beta^{n-l} \mathbf{U}_{\mathcal{F}}^H \mathbf{D}_{S[l]} \mathbf{C}_v^{-1} \mathbf{U}_{\mathcal{F}} + \beta^n \mathbf{\Pi} \right)^{-1} \\ &\quad \times \left(\sum_{l=1}^n \beta^{n-l} \mathbf{U}_{\mathcal{F}}^H \mathbf{D}_{S[l]} \mathbf{C}_v^{-1} \mathbf{y}[l] \right) \quad (56) \end{aligned}$$

Exploiting now Assumption 4, the relations (5), (33), and (11) [i.e., (9)], for sufficiently large n , the long-term behavior of recursion (56) is well approximated by:

$$\hat{\mathbf{x}}_c[n] = \mathbf{x}^o + \mathbf{U}_{\mathcal{F}} \bar{\Psi}^{-1} \sum_{l=1}^n \beta^{n-l} \mathbf{U}_{\mathcal{F}}^H \mathbf{D}_{S[l]} \mathbf{C}_v^{-1} \mathbf{v}[l], \quad (57)$$

where we have neglected the term $\beta^n \mathbf{\Pi}$ at large values of n . Thus, from (57), we have

$$\begin{aligned} & \lim_{n \rightarrow \infty} \mathbb{E} \|\hat{\mathbf{x}}_c[n] - \mathbf{x}^o\|^2 \\ & \stackrel{(a)}{=} \lim_{n \rightarrow \infty} \mathbb{E} \sum_{l=1}^n \beta^{2(n-l)} \mathbf{v}^H[l] \mathbf{C}_v^{-1} \mathbf{D}_{S[l]} \mathbf{U}_{\mathcal{F}} \bar{\Psi}^{-2} \mathbf{U}_{\mathcal{F}}^H \mathbf{D}_{S[l]} \mathbf{C}_v^{-1} \mathbf{v}[l] \\ & \stackrel{(b)}{=} \frac{1}{1 - \beta^2} \text{Tr} \left(\mathbf{U}_{\mathcal{F}}^H \text{diag}(\mathbf{p}) \mathbf{C}_v^{-1} \mathbf{U}_{\mathcal{F}} \bar{\Psi}^{-2} \right) \\ & \stackrel{(c)}{=} \frac{1 - \beta}{1 + \beta} \text{Tr} \left[\left(\mathbf{U}_{\mathcal{F}}^H \text{diag}(\mathbf{p}) \mathbf{C}_v^{-1} \mathbf{U}_{\mathcal{F}} \right)^{-1} \right] \end{aligned}$$

where in (a) we used $\mathbf{U}_{\mathcal{F}}^H \mathbf{U}_{\mathcal{F}} = \mathbf{I}$; in (b) we exploited Assumption 2, the uncorrelatedness of the observation noise, the relation $\mathbb{E} \{ \mathbf{D}_{S[l]} \mathbf{C}_v^{-1} \mathbf{D}_{S[l]} \} = \text{diag}(\mathbf{p}) \mathbf{C}_v^{-1}$, and $\lim_{n \rightarrow \infty} \sum_{l=1}^n \beta^{2(n-l)} = 1/(1 - \beta^2)$; finally, in (c) we have used (33). This concludes the proof.

ACKNOWLEDGMENT

The authors would like to thank the authors of [26] for sharing the code to reproduce the results obtained by their algorithm.

REFERENCES

- [1] 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.
- [2] 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.
- [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. Sandryhaila and J. M. F. Moura, "Discrete signal processing on graphs: Frequency analysis," *IEEE Trans. Signal Process.*, vol. 62, no. 12, pp. 3042–3054, Jun. 2014.
- [5] S. K. Narang and A. Ortega, "Perfect reconstruction two-channel wavelet filter banks for graph structured data," *IEEE Trans. Signal Process.*, vol. 60, no. 6, pp. 2786–2799, Jun. 2012.
- [6] S. K. Narang and A. Ortega, "Compact support biorthogonal wavelet filterbanks for arbitrary undirected graphs," *IEEE Trans. Signal Process.*, vol. 61, no. 19, pp. 4673–4685, Oct. 2013.
- [7] 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.
- [8] I. Z. Pesenson, "Sampling in Paley-Wiener spaces on combinatorial graphs," *Trans. Amer. Math. Soc.*, vol. 360, no. 10, pp. 5603–5627, 2008.
- [9] X. Zhu and M. Rabbat, "Approximating signals supported on graphs," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, Mar. 2012, pp. 3921–3924.
- [10] 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.
- [11] S. Sardellitti, S. Barbarossa, and P. Di Lorenzo, "On the graph Fourier transform for directed graphs," *IEEE J. Sel. Topics Signal Process.*, vol. 11, no. 6, pp. 796–811, Sep. 2017.
- [12] S. Narang, A. Gadde, and A. Ortega, "Signal processing techniques for interpolation in graph structured data," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, May 2013, pp. 5445–5449.
- [13] M. Tsitsvero, S. Barbarossa, and P. Di Lorenzo, "Signals on graphs: Uncertainty principle and sampling," *IEEE Trans. Signal Process.*, vol. 64, no. 18, pp. 4845–4860, Sep. 2016.
- [14] X. Wang, P. Liu, and Y. Gu, "Local-set-based graph signal reconstruction," *IEEE Trans. Signal Process.*, vol. 63, no. 9, pp. 2432–2444, May 2015.
- [15] A. G. Marques, S. Segarra, G. Leus, and A. Ribeiro, "Sampling of graph signals with successive local aggregations," *IEEE Trans. Signal Process.*, vol. 64, no. 7, pp. 1832–1843, Apr. 2016.
- [16] F. Gama, A. G. Marques, G. Mateos, and A. Ribeiro, "Rethinking sketching as sampling: A graph signal processing approach," arXiv:1611.00119, 2016.
- [17] R. Lafferty and J. Kondor, "Diffusion kernels on graphs and other discrete structures," in *Proc. 19th Int. Conf. Mach. Learn.*, 2002, pp. 315–322.
- [18] O. Chapelle, B. Scholkopf, and A. Zien, "Semi-supervised learning (Chapelle, O. *et al.*, Eds.; 2006) [Book Reviews]," *IEEE Trans. Neural Netw.*, vol. 20, no. 3, pp. 542–542, Mar. 2009.
- [19] M. Belkin, P. Niyogi, and V. Sindhwani, "Manifold regularization: A geometric framework for learning from labeled and unlabeled examples," *J. Mach. Learn. Res.*, vol. 7, pp. 2399–2434, 2006.
- [20] D. Romero, M. Ma, and G. B. Giannakis, "Kernel-based reconstruction of graph signals," *IEEE Trans. Signal Process.*, vol. 65, no. 3, pp. 764–778, Feb. 2017.
- [21] L. F. Chamon and A. Ribeiro, "Greedy sampling of graph signals," *IEEE Trans. Signal Process.*, vol. 66, no. 1, pp. 34–47, Jan. 2018.
- [22] S. Chen, R. Varma, A. Singh, and J. Kovačević, "Signal recovery on graphs: Fundamental limits of sampling strategies," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 2, no. 4, pp. 539–554, Dec. 2016.
- [23] A. Anis, A. Gadde, and A. Ortega, "Efficient sampling set selection for bandlimited graph signals using graph spectral proxies," *IEEE Trans. Signal Process.*, vol. 64, no. 14, pp. 3775–3789, Jul. 2016.
- [24] P. Di Lorenzo, S. Barbarossa, P. Banelli, and S. Sardellitti, "Adaptive least mean squares estimation of graph signals," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 2, no. 4, pp. 555–568, Dec. 2016.
- [25] 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.
- [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] X. Wang, M. Wang, and Y. Gu, "A distributed tracking algorithm for reconstruction of graph signals," *IEEE J. Sel. Topics Signal Process.*, vol. 9, no. 4, pp. 728–740, Jun. 2015.
- [28] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. Trends Mach. Learn.*, vol. 3, no. 1, pp. 1–122, 2011.
- [29] A. H. Sayed, *Adaptive Filters*. New York, NY, USA: Wiley, 2011.
- [30] S. P. Chepuri and G. Leus, "Sparsity-promoting sensor selection for non-linear measurement models," *IEEE Trans. Signal Process.*, vol. 63, no. 3, pp. 684–698, Feb. 2015.
- [31] S. P. Chepuri and G. Leus, "Sparse sensing for distributed detection," *IEEE Trans. Signal Process.*, vol. 64, no. 6, pp. 1446–1460, Mar. 2016.
- [32] Y. Fang, K. A. Loparo, and X. Feng, "Inequalities for the trace of matrix product," *IEEE Trans. Automat. Control*, vol. 39, no. 12, pp. 2489–2490, Dec. 1994.
- [33] M. Avriel, W. E. Diewert, S. Schaible, and I. Zang, *Generalized Convexity*. Philadelphia, PA, USA: SIAM, 2010.
- [34] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, MA, USA: Cambridge Univ. Press, 2004.
- [35] F. Scutari, F. Facchinei, and L. Lampariello, "Parallel and distributed methods for constrained nonconvex optimization—Part 1: Theory," *IEEE Trans. Signal Process.*, vol. 65, no. 8, pp. 1929–1944, Apr. 2017.
- [36] S. Schaible and J. Shi, "Recent developments in fractional programming: Single-ratio and max-min case," in *Nonlinear Analysis and Convex Analysis*, W. Takahashi and T. Tanaka, Eds. Yokohama, Japan: Yokohama Publ., pp. 493–506, 2004.
- [37] W. Dinkelbach, "On nonlinear fractional programming," *Manage. Sci.*, vol. 13, no. 7, pp. 492–498, 1967.
- [38] F. S. Cattivelli, C. G. Lopes, and A. H. Sayed, "Diffusion recursive least-squares for distributed estimation over adaptive networks," *IEEE Trans. Signal Process.*, vol. 56, no. 5, pp. 1865–1877, May 2008.

- [39] G. Mateos, I. D. Schizas, and G. B. Giannakis, "Distributed recursive least-squares for consensus-based in-network adaptive estimation," *IEEE Trans. Signal Process.*, vol. 57, no. 11, pp. 4583–4588, Nov. 2009.
- [40] K. Sun, D.-Z. Zheng, and Q. Lu, "A simulation study of OBDD-based proper splitting strategies for power systems under consideration of transient stability," *IEEE Trans. Power Syst.*, vol. 20, no. 1, pp. 389–399, Feb. 2005.
- [41] "1981–2010 U.S. Climate Normals," [Online]. Available: <https://www.ncdc.noaa.gov/data-access/land-based-station-data/land-based-datasets/climate-normals/1981-2010-normals-data>
- [42] M. A. Kramer, E. D. Kolaczyk, and H. E. Kirsch, "Emergent network topology at seizure onset in humans," *Epilepsy Res.*, vol. 79, no. 2, pp. 173–186, 2008.
- [43] M. Gavish and D. L. Donoho, "Optimal shrinkage of singular values," *IEEE Trans. Inf. Theory*, vol. 63, no. 4, pp. 2137–2152, Apr. 2017.
- [44] A. H. Sayed and V. H. Nascimento, *Energy Conservation and the Learning Ability of LMS Adaptive Filters*. New York, NY, USA: Wiley, 2003.
- [45] A. H. Sayed *et al.*, "Adaptation, learning, and optimization over networks," *Found. Trends Mach. Learn.*, vol. 7, no. 4/5, pp. 311–801, 2014.



Paolo Di Lorenzo (S'10–M'13) received the M.Sc. and the Ph.D. degrees in electrical engineering from Sapienza University of Rome, Rome, Italy, in 2008 and 2012, respectively. In 2010, he held a visiting research appointment with the Department of Electrical Engineering, University of California at Los Angeles, Los Angeles, CA, USA. From May 2015 to February 2018, he was an Assistant Professor with the Department of Engineering, University of Perugia, Perugia, Italy. He is currently an Assistant Professor with the Department of Information Engineering, Electronics, and Telecommunications, Sapienza University of Rome. He has participated in the FP7 European research projects FREEDOM, SIMTISYS, and TROPIC. His research interests include signal processing theory and methods, adaptation and learning, distributed optimization, machine learning, and graph signal processing. He is currently an Associate Editor for the *EURASIP Journal on Advances in Signal Processing*. He was the recipient of the three best student paper awards, respectively, at IEEE SPAWC'10, EURASIP EUSIPCO'11, and IEEE CAMSAP'11. He is also the recipient of the 2012 GTTI (Italian national group on telecommunications and information theory) award for the Best Ph.D. thesis in information technologies and communications.



Paolo Banelli (S'90–M'99) received the Laurea degree (*cum laude*) in electronics engineering and the Ph.D. degree in telecommunications from the University of Perugia, Perugia, Italy, in 1993 and 1998, respectively. In 2005, he was appointed as an Associate Professor with the Department of Electronic and Information Engineering, University of Perugia, where he had been an Assistant Professor since 1998. In 2001, he joined the SpinComm group, as a Visiting Researcher, in the Electrical and Computer Engineering Department, University of Minnesota, Minneapolis, MN, USA. His research interests include signal processing for wireless communications, with emphasis on multicarrier transmissions, signal processing for biomedical applications, spectrum sensing for cognitive radio, and waveform design for 5G communications, and recently graph signal processing. He was a member (during 2011–2013) of the IEEE Signal Processing Society's Signal Processing for Communications and Networking Technical Committee. In 2009, he was the General Co-Chair of the IEEE International Symposium on Signal Processing Advances for Wireless Communications. He was an Associate Editor for the IEEE TRANSACTIONS ON SIGNAL PROCESSING (during 20013–2016). He is currently an Associate Editor for the *EURASIP Journal on Advances in Signal Processing*.



Elvin Isufi (S'13) was born in Albania in 1989. He received the Master of Science degree (*cum laude*) in electrical and telecommunication engineering from the University of Perugia, Perugia, Italy, in 2014. From November 2013 to August 2014, he was a visiting member at the Circuits and Systems Group, Delft University of Technology, where he worked on his master's thesis. Since November 2014, he is working toward the Ph.D. degree in signal processing on graphs with Delft University of Technology, Delft, The Netherlands. His research interests include graph signal processing, graph and time signal processing, network anomaly detection, and network coding for underwater routing.



Sergio Barbarossa (S'84–M'88–F'12) received the M.Sc. and Ph.D. degrees in electrical engineering from Sapienza University of Rome, Rome, Italy, in 1984 and 1988, respectively. He was a Research Engineer with Selenia SpA (during 1984–1986) and the Environmental Institute of Michigan (in 1988). He was a Visiting Professor with the University of Virginia (in 1995 and 1997) and the University of Minnesota (in 1999). He is currently a Full Professor with Sapienza University of Rome. He is the author of a research monograph titled *Multiantenna Wireless Communication Systems* (Artech House, 2005). His current research focuses on topological data analysis, signal processing over graphs, mobile-edge computing, and 5G networks. He is a EURASIP Fellow. He is currently an Associate Editor for the IEEE TRANSACTIONS ON SIGNAL AND INFORMATION PROCESSING OVER NETWORKS. He was an IEEE Distinguished Lecturer during 2012–2013. He was the Scientific Coordinator of various European projects on wireless sensor networks, femtocell networks, and mobile cloud computing. He was the General Chairman of the IEEE Workshop on Signal Processing Advances in Wireless Communications (SPAWC) in 2003, and the Technical Co-Chair of SPAWC in 2013. He was an Associate Editor for the IEEE TRANSACTIONS ON SIGNAL PROCESSING (during 1998–2000 and during 2004–2006) and the IEEE SIGNAL PROCESSING MAGAZINE. He was the recipient of the 2010 EURASIP Technical Achievements Award and the 2000 and 2014 IEEE Best Paper Awards from the IEEE Signal Processing Society. He is the coauthor of papers that received the Best Student Paper Award at ICASSP 2006, SPAWC 2010, EUSIPCO 2011, and CAMSAP 2011.



Geert Leus (F'12) received the M.Sc. and Ph.D. degrees in electrical engineering from the KU Leuven, Leuven, Belgium, in June 1996 and May 2000, respectively. He is currently an "Antoni van Leeuwenhoek" Full Professor with the Faculty of Electrical Engineering, Mathematics and Computer Science, Delft University of Technology, Delft, The Netherlands. His research interests include the broad area of signal processing. He received the 2002 IEEE Signal Processing Society Young Author Best Paper Award and the 2005 IEEE Signal Processing Society Best Paper Award. He is a Fellow of the EURASIP. He was a Member-at-Large of the Board of Governors of the IEEE Signal Processing Society, the Chair for the IEEE Signal Processing for Communications and Networking Technical Committee, a member of the IEEE Sensor Array and Multichannel Technical Committee, and the Editor-in-Chief for the *EURASIP Journal on Advances in Signal Processing*. He was also on the editorial boards of the IEEE TRANSACTIONS ON SIGNAL PROCESSING, the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, the IEEE SIGNAL PROCESSING LETTERS, and the *EURASIP Journal on Advances in Signal Processing*. He is currently an Associate Editor for the *Foundations and Trends in Signal Processing* and the Editor-in-Chief of the *EURASIP Journal on Advances in Signal Processing*.