Normalized LMS Algorithm and Data-selective Strategies for Adaptive Graph Signal Estimation

Marcelo Jorge Mendes Spelta[a], Wallace Alves Martins[a,b,*]

[a] *Electrical Engineering Program (PEE/Coppe), Federal University of Rio de Janeiro (UFRJ), 21941-972, Brazil.*
[b] *Interdisciplinary Centre for Security, Reliability and Trust (SnT), University of Luxembourg, L-1359, Luxembourg.*

Abstract

This work proposes a normalized least-mean-squares (NLMS) algorithm for online estimation of bandlimited graph signals (GS) using a reduced number of noisy measurements. As in the classical adaptive filtering framework, the resulting GS estimation technique converges faster than the least-mean-squares (LMS) algorithm while being less complex than the recursive least-squares (RLS) algorithm, both recently recast as adaptive estimation strategies for the GS framework. Detailed steady-state mean-squared error and deviation analyses are provided for the proposed NLMS algorithm, and are also employed to complement previous analyses on the LMS and RLS algorithms. Additionally, two different time-domain data-selective (DS) strategies are proposed to reduce the overall computational complexity by only performing updates when the input signal brings enough innovation. The parameter setting of the algorithms is performed based on the analysis of these DS strategies, and closed formulas are derived for an accurate evaluation of the update probability when using different adaptive algorithms. The theoretical results predicted in this work are corroborated with high accuracy by numerical simulations.

Keywords: Graph signal processing, Graph signal estimation, Data-selective algorithms, Adaptive filtering

[☆] This work was financed in part by the ERC project AGNOSTIC and in part by CNPq and Faperj, Brazilian research councils.
Accepted for publication on September 30, 2019
https://doi.org/10.1016/j.sigpro.2019.107326

[*] Corresponding author
Email address: wallace.alvesmartins@uni.lu (Wallace Alves Martins)

1. Introduction

Although traditional signal processing provides a useful framework for handling data defined on regular domains, such as time series and image grids, some modern problems are better represented by more general structures. For instance, irregular datasets arise from popular research fields, such as Internet of things and big data applications, and usually require performing processing tasks on their irregular domains [1]-[2]. In order to model these data points and their complex interactions, the signal processing community has recently turned its attention to a general mathematical structure called graph, and has been extending classical results to this irregular-structure domain, giving rise to the so-called graph signal processing (GSP) [2]-[21].

GSP involves the definition of a graph signal (GS) along with some analytical tools, such as its frequency-domain representation [3]-[5], bandlimited properties, as well as sampling [8]-[11] and reconstruction ideas [12]-[21]. In particular, the online reconstruction method proposed in [14] suggests the use of a procedure based on the least-mean-squares (LMS) algorithm [24]-[27] for a bandlimited GS estimation context, which represents the first attempt to merge the GSP field with the well-established adaptive filtering area [26]-[27]. This blending work is further extended in [15]-[20], where the authors present centralized and distributed GS reconstruction strategies, consolidating the use of adaptive methods in GSP. Besides the LMS algorithm, centralized and distributed versions of the recursive least-squares (RLS) algorithm [26]-[28] are also proposed for the same reconstruction scenarios.

The appeal for using adaptive filtering ideas in the GS estimation problem arises from the expected benefits of enabling online reconstruction and tracking time-varying graph signals in noisy environments. Although traditional interpolation methods like Kriging [29] can be used for handling signal inference in irregular domains with static reference values, adaptive algorithms seem more suitable for dealing with dynamic GSs due to their reduced complexity and benefit of online estimation. Moreover, recent papers extend the adaptive filtering concepts to cover time-varying sampling scenarios [17, 18], highlighting the flexibility and potential of the adaptive approach when employed in the GS estimation context.

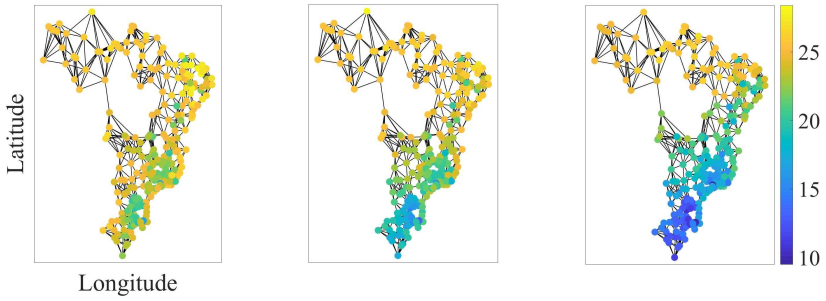
Like their classical counterparts [26]-[27], the LMS algorithm from [14] is much slower than the centralized RLS in [17]-[18], as verified in [21] for both static and time-varying reference GS. The RLS main drawback is its large computational complexity, which imposes a practical problem in applications with many nodes. Aiming at proposing an adaptive technique for GS estimation that has faster convergence speed than the LMS, while inducing a lower computational burden than the RLS, we propose a normalized least-mean-squares (NLMS) algorithm, which resembles its classical version [26]-[27], [30]-[31] and provides a trade-off between the GSP LMS and RLS algorithms. In particular, when considering the scenario with fixed and known sampling nodes, one observes that the proposed implementation of the NLMS algorithm results in a procedure that converges much faster than the LMS in [14], but with the very

same complexity.

Besides its derivation, this work provides a detailed theoretical analysis on the steady-state behavior of the proposed NLMS algorithm, presenting a range of values for the underlying convergence factor that guarantee the algorithm stability and ability to provide asymptotically unbiased GS estimates, as well as closed-form expressions for estimating the corresponding mean-squared error (MSE) and mean-squared deviation (MSD). Moreover, based on the NLMS study, we employ the same methodology and further complement the LMS and RLS analyses in [18] by obtaining more general steady-state expressions for the related figures of merit (FoMs) such as the MSE and MSD.

An important concern in several practical applications is power consumption. As some classical adaptive filtering algorithms implement the data-selection idea in order to obtain power savings [32]-[34], we propose tailoring this idea to the GSP context. Although some data-selection adaptive algorithm families like the set-membership [32]-[33] perform more complex computations involving the estimation of a solution set, in this first approach we propose the use of a simpler family called data-selective (DS) [34]-[36], which involves a more direct implementation based on a point update. Since the GS estimation problem presents some differences in comparison to the classical adaptive filtering problem, the novelty test at each algorithm iteration uses an error vector instead of a scalar value as in [34]. Thus, we propose two novelty tests: one based on the individual error component values, namely the component-wise error constraint strategy, and another that uses the vector squared ℓ_2-norm as a reference, the so-called ℓ_2-norm error constraint strategy. For both DS strategies we show how to define constraint parameters that allow an accurate estimate of the update probability when using not only the proposed GSP NLMS, but also the GSP LMS and RLS algorithms [18].

This paper is organized as follows: Section 2 presents some basic GSP concepts and highlights some aspects of classical adaptive filtering. Section 3 blends these two different areas by discussing the use of adaptive procedures, like the LMS and RLS algorithms, for the online reconstruction of bandlimited GS. The proposals of this work start to be detailed in Section 4 with the derivation and analysis of the NLMS algorithm for the GS context. In Section 5 we complement the analysis in [18] and provide important steady-state results about the GSP LMS and RLS algorithms. Section 6 describes two proposals of DS strategies with respect to time for GSP NLMS, LMS, and RLS algorithms; constraint parameters related to these DS strategies are also proposed based on the corresponding estimation of steady-state error variance, allowing accurate estimates of the underlying update probability of each algorithm. Section 7 describes some numerical simulations using a bandlimited approximation [21] of the Brazilian temperature dataset in [22]-[23] to validate the theoretical predictions from Sections 4-6. Additionally, Section 7 also compares the GSP adaptive algorithms' tracking performance when handling real-world data taken from the U.S. climate normals dataset [37]. At last, some conclusions are drawn in Section 8.



(a) January (b) April (c) July

Figure 1: GS representation of 1961-1990 monthly average temperatures (degree Celsius) from Brazilian weather stations [21]-[23].

2. Background on GSP and Adaptive Filtering

2.1. Graph Signal Processing

A graph structure $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is made up of N nodes or vertices $\mathcal{V} = \{v_1, v_2, ..., v_N\}$ linked by a set of edges $\mathcal{E} = \{\widehat{v_n v_m}\}$. For describing the connection strength between nodes v_n and $v_m \in \mathcal{V}$, each edge $\widehat{v_n v_m} \in \mathcal{E}$ is associated with a weight $a_{nm} \in \mathbb{R}$, which can be understood as a proximity or similarity metric. The $N \times N$ matrix formed by elements a_{nm} in its n^{th} row and m^{th} column is called the adjacency matrix $\mathbf{A}$ and stores information about the graph connections in a compact representation, being widely used in GSP approaches [4]-[6]. Although some problems might require the use of directed graphs (*digraphs*) for which $a_{nm} \neq a_{mn}$ for some $n, m \in \mathcal{N} = \{1, 2, \ldots, N\}$, this work only considers undirected graph structures, from which one concludes that $\mathbf{A}$ is symmetric. Additionally, an alternative representation for undirected graph structures is given by the Laplacian matrix $\mathbf{L} \in \mathbb{R}^{N \times N}$ defined as $\mathbf{L} = \mathbf{K} - \mathbf{A}$, where $\mathbf{K}$ is a diagonal matrix with diagonal entries $k_n = \sum_{m=1}^{N} a_{nm}$ [6].

Considering a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with N vertices, the graph signal (GS) $x : \mathcal{V} \to \mathbb{R}$ defined on the nodes of $\mathcal{G}$ can be represented as a vector $\mathbf{x} \in \mathbb{R}^N$, whose n^{th} entry, x_n, contains the function value at vertex $v_n \in \mathcal{V}$ [3]. An example of a practical GS representation is displayed in Figure 1, where the graph nodes indicate the geographical location of Brazilian weather stations and the color of each vertex v_n represents its average monthly temperature x_n according to the color scale [21]-[23].

Motivated by the classical signal processing framework, one can define the *graph Fourier transform* (GFT) of a GS $\mathbf{x} \in \mathbb{R}^N$ as its projection onto a set of orthonormal vectors $\{\mathbf{u}_n\} \subset \mathbb{R}^N$, where $n \in \mathcal{N}$. Those basis vectors are usually chosen as the orthonormal eigenvectors of either the adjacency matrix $\mathbf{A}$ [4]-[5] or the Laplacian matrix $\mathbf{L}$ [3], so that the information inherent to the graph structure is naturally embedded in the resulting frequency-domain representation. As we only consider undirected graphs here, the spectral decomposition of

$\mathbf{L}$ (or $\mathbf{A}$) assumes the form $\mathbf{U}\mathbf{\Lambda}\mathbf{U}^{\mathrm{T}}$, where $\mathbf{\Lambda} \in \mathbb{R}^{N\times N}$ is the diagonal eigenvalue matrix and $\mathbf{U} \in \mathbb{R}^{N\times N}$ is the orthonormal eigenvector matrix formed by $\{\mathbf{u}_n\}$. Thus, the GFT of a GS $\mathbf{x}$ is given by [3]-[4]

$$\mathbf{s} = \mathbf{U}^{\mathrm{T}}\mathbf{x}\,, \tag{1}$$

and for recovering $\mathbf{x}$ from its frequency-domain representation $\mathbf{s}$, one defines the *inverse graph Fourier transform* (IGFT) as

$$\mathbf{x} = \mathbf{U}\mathbf{s}\,. \tag{2}$$

A GS $\mathbf{x}_{\mathrm{o}} \in \mathbb{R}^N$ is *bandlimited* or *spectrally sparse (ssparse)* when its frequency-domain representation $\mathbf{s}$ has zero entries. If one selects $\mathcal{F}$ as an index subset of $\mathcal{N}$, a GS $\mathbf{x}_{\mathrm{o}}$ is called $\mathcal{F}$-ssparse if $\mathbf{s}$ is such that $\mathbf{s}_{\mathcal{N}\setminus\mathcal{F}}$ is a null vector [11], i.e., the components of $\mathbf{s}$ with indices in $\mathcal{N}\setminus\mathcal{F}$ are equal to zero, where $\mathcal{N}\setminus\mathcal{F}$ represents the complementary set of $\mathcal{F}$ with respect to N. In other words, this *frequency set* or *support* of $\mathcal{F}$ is mathematically defined as $\mathcal{F} = \{f \in \mathcal{N} \,|\, s_f \neq 0\}$ [14]. Thus, considering that $\mathbf{U}_{\mathcal{F}} \in \mathbb{R}^{N\times|\mathcal{F}|}$ and $\mathbf{s}_{\mathcal{F}} \in \mathbb{R}^{|\mathcal{F}|}$ are, respectively, the matrix whose columns are in $\{\mathbf{u}_n\}_{n\in\mathcal{F}}$ and the frequency-domain representation indexed by the elements in $\mathcal{F}$, from (2) one gets

$$\mathbf{x}_{\mathrm{o}} = \mathbf{U}_{\mathcal{F}}\mathbf{s}_{\mathcal{F}}\,. \tag{3}$$

In terms of sampling and reconstruction of bandlimited graph signals, some works [10, 19] describe these operations over a GS $\mathbf{x}_{\mathrm{o}}$ as the result of pre-multiplying it by a sampling matrix $\mathbf{D}_{\mathcal{S}} \in \mathbb{R}^{N\times N}$ and an interpolation matrix $\mathbf{\Phi} \in \mathbb{R}^{N\times N}$. Sampling is the operation of collecting only a limited number of values from the GS, whose reading positions are determined by the *sampling set* $\mathcal{S} \subseteq \mathcal{V}$. In this context, let $\mathbf{D}_{\mathcal{S}} \in \mathbb{R}^{N\times N}$ denote a diagonal matrix with entries d_n, where $d_n = 1$ if $v_n \in \mathcal{S}$ and $d_n = 0$ otherwise. Thus, one can write the sampled vector $\mathbf{x}_{\mathcal{S}} \in \mathbb{R}^N$ as

$$\mathbf{x}_{\mathcal{S}} = \mathbf{D}_{\mathcal{S}}\mathbf{x}\,. \tag{4}$$

Considering the representation in (3), in order to achieve perfect reconstruction for any $\mathcal{F}$-ssparse signal $\mathbf{x}_{\mathrm{o}}$—that is $\mathbf{x}_{\mathrm{o}} = \mathbf{\Phi}\mathbf{D}_{\mathcal{S}}\mathbf{x}_{\mathrm{o}}$—, the product $\mathbf{D}_{\mathcal{S}}\mathbf{U}_{\mathcal{F}}$ must be full rank, thus requiring $|\mathcal{S}| \geq |\mathcal{F}|$ [7, 10]. This fact indicates the importance of an adequate choice for the sampling set $\mathcal{S}$ and its connection to the graph structure $\mathbf{U}_{\mathcal{F}}$.

It is important, however, to remember that in many practical applications the sampled signal comes from noisy measurements and the impact of a proper selection of $\mathcal{S}$ is even bigger. In fact, many papers consider the use of optimal sampling strategies that minimize specific metrics of the reconstructed GS [10]-[11]. Particularly, this work employs online recovery strategies for reconstructing the original GS from its sampled noisy version, which can be seen as a time-varying interpolation matrix $\mathbf{\Phi}[k]$, and a few optimal sampling strategies have already been proposed in this context [14, 19], even covering time-varying probabilistic sampling [16, 18]-[19].

Hence, by taking into account the rich literature addressing sampling strategies, this paper does not propose new graph sampling schemes, it simply considers the existence of a possibly time-varying sampling set $\mathcal{S}[k]$, whose instantaneous node selection is not further explored. In particular, when handling stationary scenarios, we presume that $\mathcal{S}[k]$ converges to a static set $\mathcal{S}$ in steady-state. Moreover, we assume a previous knowledge of the graph structure and consider that $\mathcal{F}$ does not change with time and is *a priori* known, for the sake of simplicity. This assumption results in a constant matrix $\mathbf{U}_{\mathcal{F}}$.

2.2. Adaptive Signal Processing

In a supervised adaptive filter system identification configuration [27] one has a desired signal $d[k] \in \mathbb{R}$, an input $\mathbf{x}[k] \in \mathbb{R}^N$, and the parameter vector $\hat{\mathbf{h}}_{\mathrm{o}}[k] \in \mathbb{R}^N$, which is an estimate of the possibly time-varying unknown system parameters $\mathbf{h}_{\mathrm{o}}[k] \in \mathbb{R}^N$. From this simple configuration an instantaneous error signal is defined as

$$e[k] = d[k] - \mathbf{x}^{\mathrm{T}}[k]\hat{\boldsymbol{h}}_{\mathrm{o}}[k]\,, \tag{5}$$

i.e., it is the difference between the desired signal and the adaptive filter output. Based on the instantaneous error signal $e[k]$ in (5) it is possible to define different error metrics and, for each one of them, an alternative adaptive algorithm is derived following its minimization. This is the case for two particularly famous adaptive algorithms, the LMS and the RLS, which arise from the minimization of the mean square error via stochastic gradient and weighted least-square error, respectively [26]-[27].

For comparing the performance of different adaptive algorithms, some commonly used metrics are the mean-square error (MSE) and mean-squared deviation (MSD) defined as

$$\mathrm{MSE}[k] = \mathbb{E}[e^2[k]] \text{ and } \mathrm{MSD}[k] = \mathbb{E}[\|\hat{\boldsymbol{h}}_{\mathrm{o}}[k] - \mathbf{h}_{\mathrm{o}}[k]\|_2^2]\,, \tag{6}$$

in which $\mathbb{E}[g(\boldsymbol{w})]$ denotes the expected value of a generic function $g(\cdot)$ of a random vector $\boldsymbol{w}$ with realizations denoted as $\mathbf{w}$. Both MSE and MSD indicate how far, on average, the current estimate $\hat{\mathbf{h}}_{\mathrm{o}}[k]$ is from generating the desired output and from the unknown system parameters, respectively. Additionally, computational complexity is also a concern, since an improvement on convergence speed usually comes at the cost of increasing the number of arithmetic operations performed at each iteration.

3. Adaptive Estimation of Graph Signals

Consider the problem of estimating a bandlimited, or approximately bandlimited, reference GS $\mathbf{x}_{\mathrm{o}}[k] \in \mathbb{R}^N$ from a reduced set $\mathcal{S}[k] \subseteq \mathcal{V}$ of noisy measurements. In this context, the N-dimensional noisy reference (random) signal is

$$\boldsymbol{x}_w[k] = \mathbf{x}_{\mathrm{o}}[k] + \boldsymbol{w}[k]\,, \tag{7}$$

in which $\boldsymbol{w}[k]$ is the measurement random noise, assumed to be zero mean and with a covariance matrix $\mathbb{E}[\boldsymbol{w}[k]\boldsymbol{w}^{\mathrm{T}}[k]] = \mathbf{C}_w[k]$. Since only the node signals indexed by $\mathcal{S}[k]$ are acquired, the reference measurements at time instant k are in fact $\mathbf{D}_{\mathcal{S}[k]}\mathbf{x}_w[k]$.

The estimation error vector $\mathbf{e}[k] \in \mathbb{R}^N$ can be defined based on (3) and by considering only the acquired measurements, i.e.

$$\mathbf{e}[k] = \mathbf{D}_{\mathcal{S}[k]}(\mathbf{x}_w[k] - \mathbf{U}_{\mathcal{F}}\hat{\mathbf{s}}_{\mathcal{F}}[k])\,, \tag{8}$$

where $\hat{\mathbf{s}}_{\mathcal{F}}[k] \in \mathbb{R}^{|\mathcal{F}|}$ is the current frequency-domain estimate.

Although both standard and GSP adaptive scenarios are motivated by similar ideas, the corresponding instantaneous errors in (5) and (8) have distinct dimensions. Besides that, (8) replaces the influence of the input signal in (5) with the graph structure represented by $\mathbf{U}_{\mathcal{F}}$, taken here as time invariant. Thus, due to these slight differences, we expect the GS estimation algorithms to resemble their classical counterparts, yet presenting some peculiarities. To investigate these differences, we present the GSP LMS [14] and RLS [17] algorithms and establish the equivalent FoMs for evaluating the performance of adaptive methods for GS estimation.

3.1. LMS Algorithm

Among a large number of traditional adaptive filtering methods, the LMS algorithm [24]-[27] stands out as one of the most popular techniques due to its simple implementation and reduced computational complexity. Based on the theoretical Wiener filter formulation, the LMS method takes a practical approach by replacing the minimization of the MSE in (6) with the minimization of the instantaneous squared error in (5) to define its update equation.

In an attempt to obtain an LMS-based equivalent algorithm for the GS reconstruction context, [14] considers the error signal available as $\mathbf{e}[k]$ from (8) and, with a clear inspiration from the Wiener filter idea, defines a reference convex problem as

$$\min_{\tilde{\boldsymbol{s}}_{\mathcal{F}}} \mathbb{E}\left[\, \|\mathbf{D}_{\mathcal{S}[k]}(\boldsymbol{x}_w[k] - \mathbf{U}_{\mathcal{F}}\tilde{\boldsymbol{s}}_{\mathcal{F}})\|_2^2 \,\right]\,, \tag{9}$$

in which $\tilde{\boldsymbol{s}}_{\mathcal{F}}$ represents a free optimization variable corresponding to the algorithm update $\hat{\mathbf{s}}_{\mathcal{F}}[k+1]$.

Similarly to the original LMS, the GSP LMS algorithm in [14, 18] employs a stochastic gradient approach to solve (9) and finds an update expression for $\hat{\mathbf{s}}_{\mathcal{F}}[k+1]$. Then, from the IGFT (2) one easily obtains a vertex-domain estimate $\hat{\mathbf{x}}_{\mathrm{o}}[k]$ for the bandlimited GS $\mathbf{x}_{\mathrm{o}}[k]$, which corresponds to the GSP LMS update equation [14, 18]

$$\hat{\mathbf{x}}_{\mathrm{o}}[k+1] = \hat{\mathbf{x}}_{\mathrm{o}}[k] + \mu_{\mathrm{L}}\mathbf{U}_{\mathcal{F}}\mathbf{U}_{\mathcal{F}}^{\mathrm{T}}\mathbf{e}[k]\,, \tag{10}$$

where $\mu_{\mathrm{L}} \in \mathbb{R}_+$ is a design parameter called convergence factor whose purpose is to control the trade-off between increasing the convergence speed and reducing the steady-state error. An analysis about the range of μ_{L} that guarantees algorithm stability is presented in [14, 18].

In terms of traditional adaptive strategies, many LMS-based algorithms have been proposed in order to take advantage of unexplored features of the original method, mainly for enhancing its convergence speed. A particular LMS-based algorithm that is worth mentioning is the normalized least-mean-squares (NLMS) algorithm [26]-[27], which usually improves the convergence speed by using a time-varying convergence factor. We shall extend the idea of finding LMS-based strategies for GS estimation [14, 20] and propose the GSP NLMS algorithm in Section 4.

3.2. RLS Algorithm

An alternative approach to enhance convergence speed with respect to the LMS algorithm is to consider a different cost function, like the weighted least-squares (WLS) from which the original RLS algorithm is derived [26]-[27]. By following a similar idea, the authors in [18] propose the GSP RLS via a centralized version of the RLS method for online reconstruction of graph signals, which results in an algorithm with faster convergence but higher computational burden than the GSP LMS.

Instead of finding an instantaneous solution to the convex problem (9), the GSP RLS [18] evaluates its frequency-domain update estimate $\hat{\mathbf{s}}_{\mathcal{F}}[k+1]$ by minimizing the objective function

$$\min_{\tilde{\boldsymbol{s}}_{\mathcal{F}}} \sum_{l=1}^{k} \beta_{\mathrm{R}}^{k-l} \|\mathbf{D}_{\mathcal{S}[l]}(\mathbf{x}_w[l] - \mathbf{U}_{\mathcal{F}}\tilde{\boldsymbol{s}}_{\mathcal{F}})\|^2_{\mathbf{C}_w^{-1}[k]} + \beta_{\mathrm{R}}^{k}\|\tilde{\boldsymbol{s}}_{\mathcal{F}}\|^2_{\boldsymbol{\Pi}}, \tag{11}$$

where $\tilde{\boldsymbol{s}}_{\mathcal{F}}$ represents a free optimization variable, the forgetting factor β_{R} is in the range $0 \ll \beta_{\mathrm{R}} \leq 1$, and the regularization matrix $\boldsymbol{\Pi}$, which is usually taken as $\boldsymbol{\Pi} = \delta\mathbf{I}$ with a small $\delta > 0$, is included to account for ill-conditioning in the first iterations.

An online method for evaluating the WLS solution of (11) employs the ancillary variables $\boldsymbol{\Psi}[k] \in \mathbb{R}^{|\mathcal{F}|\times|\mathcal{F}|}$ and $\boldsymbol{\psi}[k] \in \mathbb{R}^{|\mathcal{F}|}$ defined recursively as

$$\begin{aligned} \boldsymbol{\Psi}[k] &= \beta_{\mathrm{R}}\boldsymbol{\Psi}[k-1] + \mathbf{U}_{\mathcal{F}}^{\mathrm{T}}\mathbf{D}_{\mathcal{S}[k]}\mathbf{C}_w^{-1}[k]\mathbf{D}_{\mathcal{S}[k]}\mathbf{U}_{\mathcal{F}}\,, \\ \boldsymbol{\psi}[k] &= \beta_{\mathrm{R}}\boldsymbol{\psi}[k-1] + \mathbf{U}_{\mathcal{F}}^{\mathrm{T}}\mathbf{D}_{\mathcal{S}[k]}\mathbf{C}_w^{-1}[k]\mathbf{D}_{\mathcal{S}[k]}\mathbf{x}_w[k]\,. \end{aligned} \tag{12}$$

From these variables, the solution $\hat{\mathbf{s}}_{\mathcal{F}}[k+1]$ of (11) satisfies $\boldsymbol{\Psi}[k]\,\hat{\mathbf{s}}_{\mathcal{F}}[k+1] = \boldsymbol{\psi}[k]$. Since $\boldsymbol{\Psi}[k]$ has full rank, one finds from (2) that the estimate $\hat{\mathbf{x}}_{\mathrm{o}}[k+1]$ for the RLS algorithm is

$$\hat{\mathbf{x}}_{\mathrm{o}}[k+1] = \mathbf{U}_{\mathcal{F}}\boldsymbol{\Psi}^{-1}[k]\boldsymbol{\psi}[k]. \tag{13}$$

In fact, it is worth mentioning that equations in (12) are more general than the ones in [17]-[18] because they allow the symmetric matrix $\mathbf{C}_w[k]$ to assume non-diagonal structures. This generic case is not covered in [17]-[18] because the diagonal assumption actually simplifies the task of designing optimal sampling strategies, which is one of the goals of those papers. Moreover, though the GSP

RLS algorithm in [17]-[18] suggests the use of (12) and (13) for computing the estimate $\hat{\mathbf{x}}_{\mathrm{o}}[k+1]$, Appendix A shows that these expressions are equivalent to

$$\begin{aligned} \boldsymbol{\Psi}[k] &= \beta_{\mathrm{R}}\boldsymbol{\Psi}[k-1] + \mathbf{U}_{\mathcal{F}}^{\mathrm{T}}\mathbf{D}_{\mathcal{S}[k]}\mathbf{C}_w^{-1}[k]\mathbf{D}_{\mathcal{S}[k]}\mathbf{U}_{\mathcal{F}}\,, \\ \hat{\mathbf{x}}_{\mathrm{o}}[k+1] &= \hat{\mathbf{x}}_{\mathrm{o}}[k] + \mathbf{U}_{\mathcal{F}}\boldsymbol{\Psi}^{-1}[k]\mathbf{U}_{\mathcal{F}}^{\mathrm{T}}\mathbf{D}_{\mathcal{S}[k]}\mathbf{C}_w^{-1}[k]\mathbf{e}[k]\,. \end{aligned} \tag{14}$$

In particular, by considering $\mathbf{C}_w^{-1}[k] = \mathbf{C}_w^{-1}$, $\mathcal{S}[k] = \mathcal{S}$, and the initialization $\boldsymbol{\Psi}[0] = \boldsymbol{\Pi}$ [17]-[18], we rewrite $\boldsymbol{\Psi}[k]$ in (14) as

$$\boldsymbol{\Psi}[k] = \beta_{\mathrm{R}}^k\boldsymbol{\Pi} + (\mathbf{U}_{\mathcal{F}}^{\mathrm{T}}\mathbf{D}_{\mathcal{S}}\mathbf{C}_w^{-1}\mathbf{D}_{\mathcal{S}}\mathbf{U}_{\mathcal{F}})\frac{(1-\beta_{\mathrm{R}}^k)}{(1-\beta_{\mathrm{R}})}\,. \tag{15}$$

Additionally, as $k \to \infty$, one has that $\beta_{\mathrm{R}}^k \to 0$. By defining $\mathbf{M}' = (\mathbf{U}_{\mathcal{F}}^{\mathrm{T}}\mathbf{D}_{\mathcal{S}}\mathbf{C}_w^{-1}\mathbf{D}_{\mathcal{S}}\mathbf{U}_{\mathcal{F}})^{-1}$, then when k increases, the GSP RLS update tends to the form (to be further explored)

$$\hat{\mathbf{x}}_{\mathrm{o}}[k+1] = \hat{\mathbf{x}}_{\mathrm{o}}[k] + (1-\beta_{\mathrm{R}})\mathbf{U}_{\mathcal{F}}\mathbf{M}'\mathbf{U}_{\mathcal{F}}^{\mathrm{T}}\mathbf{D}_{\mathcal{S}}\mathbf{C}_w^{-1}\mathbf{e}[k]\,. \tag{16}$$

3.3. Figures of Merit

The extension of the traditional MSE and MSD in (6) to the GSP estimation context is straightforward, being given by

$$\mathrm{MSE}_{\mathrm{G}}[k] = \mathbb{E}\{\|\boldsymbol{e}[k]\|_2^2\} \text{ and } \mathrm{MSD}_{\mathrm{G}}[k] = \mathbb{E}\{\|\Delta\hat{\boldsymbol{x}}_{\mathrm{o}}[k]\|_2^2\}, \tag{17}$$

where $\Delta\hat{\boldsymbol{x}}_{\mathrm{o}}[k] = \hat{\boldsymbol{x}}_{\mathrm{o}}[k] - \mathbf{x}_{\mathrm{o}}[k]$ is the difference between the current estimator $\hat{\boldsymbol{x}}_{\mathrm{o}}[k]$ and the original GS $\mathbf{x}_{\mathrm{o}}[k]$, and we use the subscript G for avoiding confusion with (6). Moreover, due to the property $\mathbf{U}_{\mathcal{F}}^{\mathrm{T}}\mathbf{U}_{\mathcal{F}} = \mathbf{I}$, if we consider (3) and define

$$\Delta\hat{\boldsymbol{s}}_{\mathcal{F}}[k] = \hat{\boldsymbol{s}}_{\mathcal{F}}[k] - \mathbf{s}_{\mathcal{F}}[k]\,, \tag{18}$$

from (17) we find that the $\mathrm{MSD}_{\mathrm{G}}$ is also given as

$$\mathrm{MSD}_{\mathrm{G}}[k] = \mathbb{E}\{\|\Delta\hat{\boldsymbol{s}}_{\mathcal{F}}[k]\|_2^2\}\,. \tag{19}$$

A disadvantage of using the scalar metrics in (17) is that they potentially hide the occurrence of large error entries in (8); an alternative FoM should be defined for estimating each component of the error vector. This more general FoM relies on statistics of $e_n[k]$, the n^{th} component of $\mathbf{e}[k]$ in (8), and provides a more accurate insight of the algorithm overall performance. Note that, from (7), (8), and (18), one has

$$e_n[k] = d_n[k](w_n[k] - \mathbf{u}_{n_{\mathcal{F}}}^{\mathrm{T}}\Delta\hat{\boldsymbol{s}}_{\mathcal{F}}[k])\,, \tag{20}$$

where $d_n[k] \in \{0,1\}$ depends on $\mathcal{S}[k]$, $w_n[k]$ is the n^{th} entry of $\boldsymbol{w}[k]$, and $\mathbf{u}_{n_{\mathcal{F}}}^{\mathrm{T}}$ is the n^{th} row of $\mathbf{U}_{\mathcal{F}}$.

If one works with an asymptotically unbiased estimator $\hat{\boldsymbol{s}}_{\mathcal{F}}[k]$ such that $\mathbb{E}[\Delta\hat{\boldsymbol{s}}_{\mathcal{F}}[k]]$ converges to the null vector in steady state, which holds true for

both GSP LMS and RLS algorithms [18], and by recalling that the noise vector $\boldsymbol{w}[k]$ has zero mean, then one has $\mathbb{E}[e_n[k]] \to 0$ as k grows to infinity.

By assuming that $\boldsymbol{w}[k]$ is uncorrelated with $\Delta\hat{\boldsymbol{s}}_{\mathcal{F}}[k]$ and that $\mathcal{S}[k]$ converges to a constant sampling set $\mathcal{S}$, then taking the expected value of the squared expression in (20) allows one to compute the steady-state error variance

$$\sigma^2_{e_n} = \lim_{k\to\infty} d_n[k](\,\mathbb{E}[w^2_n[k]] + \mathbf{u}^{\mathrm{T}}_{n_{\mathcal{F}}}\mathbb{E}[\Delta\hat{\boldsymbol{s}}_{\mathcal{F}}[k]\Delta\hat{\boldsymbol{s}}^{\mathrm{T}}_{\mathcal{F}}[k]]\mathbf{u}_{n_{\mathcal{F}}}), \tag{21}$$

thus yielding, from (17), the following steady-state $\mathrm{MSE_G}$:

$$\mathrm{MSE}^*_{\mathrm{G}} = \lim_{k\to\infty} \mathrm{MSE_G}[k] = \sum_{n=1}^{N} \sigma^2_{e_n}\,. \tag{22}$$

Based on the $\mathrm{MSD_G}$ in (19), one can also define

$$\mathrm{MSD}^*_{\mathrm{G}} = \lim_{k\to\infty} \mathrm{MSD}[k] = \lim_{k\to\infty} \mathrm{tr}\{\,\mathbb{E}[\Delta\hat{\boldsymbol{s}}_{\mathcal{F}}[k]\Delta\hat{\boldsymbol{s}}^{\mathrm{T}}_{\mathcal{F}}[k]]\,\}, \tag{23}$$

in which the matrix trace $\mathrm{tr}\{\cdot\}$ operator is employed to show the dependency of this FoM on the steady-state matrix $\mathbb{E}[\Delta\hat{\boldsymbol{s}}_{\mathcal{F}}[\infty]\Delta\hat{\boldsymbol{s}}^{\mathrm{T}}_{\mathcal{F}}[\infty]]$. Thus, this matrix plays a central role when computing the FoMs in (21), (22), and (23). In the analyses conducted in Subsection 4.4 and Section 5, we first estimate $\mathbb{E}[\Delta\hat{\boldsymbol{s}}_{\mathcal{F}}[\infty]\Delta\hat{\boldsymbol{s}}^{\mathrm{T}}_{\mathcal{F}}[\infty]]$ for each algorithm and then evaluate its respective FoMs according to (21), (22), and (23).

In particular, from the three FoMs covered in this subsection, only the $\mathrm{MSD}^*_{\mathrm{G}}$ in (23) has been previously considered in [14, 17, 18]. The error component variances $\sigma^2_{e_n}$ and $\mathrm{MSE_G}$ in (21) and (22), respectively, are natural extensions of standard FoMs appearing in the adaptive filtering literature and are employed in this work for providing a deeper understanding of the GS estimation problem.

4. NLMS Graph Signal Estimation Algorithm

In this section we propose and analyze the GSP NLMS algorithm for GS estimation.

4.1. Algorithm Derivation

Inspired by the traditional NLMS algorithm derivation [27], we search for a possibly time-varying factor $\mu_{\mathrm{L}} \in \mathbb{R}_+$ that improves the overall convergence rate of the GSP LMS algorithm described in Subsection 3.1. However, as in the GSP context one has to deal with an error vector $\mathbf{e}[k]$ instead of a scalar parameter, we generalize this idea of a convergence factor μ_{L} and adopt a symmetric convergence matrix $\mathbf{M} \in \mathbb{R}^{|\mathcal{F}|\times|\mathcal{F}|}$. Then, from (10) the frequency-domain update equation becomes

$$\hat{\mathbf{s}}_{\mathcal{F}}[k+1] = \hat{\mathbf{s}}_{\mathcal{F}}[k] + \mathbf{M}\mathbf{U}^{\mathrm{T}}_{\mathcal{F}}\mathbf{D}_{\mathcal{S}[k]}(\mathbf{x}_w[k] - \mathbf{U}_{\mathcal{F}}\hat{\mathbf{s}}_{\mathcal{F}}[k])\,. \tag{24}$$

Following the same reasoning for defining $\mathbf{e}[k]$ in (8), we can also define the *a posteriori* reconstruction error vector

$$\boldsymbol{\varepsilon}[k] = \mathbf{D}_{\mathcal{S}[k]}(\mathbf{x}_w[k] - \mathbf{U}_{\mathcal{F}}\hat{\mathbf{s}}_{\mathcal{F}}[k+1])\,. \tag{25}$$

In order to avoid confusion with the *a posteriori* error $\boldsymbol{\varepsilon}[k]$, we shall call $\mathbf{e}[k]$ as the *a priori* error.

According to equations (24) and (25), the *a posteriori* error $\boldsymbol{\varepsilon}[k]$ can be rewritten as function of the *a priori* error, i.e.

$$\boldsymbol{\varepsilon}[k] = \mathbf{D}_{\mathcal{S}[k]}(\mathbf{I} - \mathbf{U}_{\mathcal{F}}\mathbf{M}\mathbf{U}_{\mathcal{F}}^{\mathrm{T}})\mathbf{e}[k]\,. \tag{26}$$

Since an instantaneous estimate of how close the error vectors are to each other is given by $\Delta\tilde{e}^2[k] = \|\boldsymbol{\varepsilon}[k]\|_2^2 - \|\mathbf{e}[k]\|_2^2$, when using equation (26) we find that

$$\Delta\tilde{e}^2[k] = \mathbf{e}^{\mathrm{T}}[k]\mathbf{U}_{\mathcal{F}}(-2\mathbf{M} + \mathbf{M}\mathbf{U}_{\mathcal{F}}^{\mathrm{T}}\mathbf{D}_{\mathcal{S}[k]}\mathbf{U}_{\mathcal{F}}\mathbf{M})\mathbf{U}_{\mathcal{F}}^{\mathrm{T}}\mathbf{e}[k]. \tag{27}$$

In order to select the symmetric matrix $\mathbf{M}$ that minimizes $\Delta\tilde{e}^2[k]$, we take the derivative of (27) with respect to $\mathbf{M}$ [38], yielding

$$2\left[(\mathbf{U}_{\mathcal{F}}^{\mathrm{T}}\mathbf{D}_{\mathcal{S}[k]}\mathbf{U}_{\mathcal{F}})\mathbf{M} - \mathbf{I}\right]\mathbf{U}_{\mathcal{F}}^{\mathrm{T}}\mathbf{e}[k]\mathbf{e}^{\mathrm{T}}[k]\mathbf{U}_{\mathcal{F}} = \mathbf{0}\,, \tag{28}$$

so that the symmetric matrix $\mathbf{M}[k] = (\mathbf{U}_{\mathcal{F}}^{\mathrm{T}}\mathbf{D}_{\mathcal{S}[k]}\mathbf{U}_{\mathcal{F}})^{-1}$ minimizes the squared error difference $\Delta\tilde{e}^2[k]$. Based on this result, the frequency-domain update expression given by

$$\hat{\mathbf{s}}_{\mathcal{F}}[k+1] = \hat{\mathbf{s}}_{\mathcal{F}}[k] + (\mathbf{U}_{\mathcal{F}}^{\mathrm{T}}\mathbf{D}_{\mathcal{S}[k]}\mathbf{U}_{\mathcal{F}})^{-1}\mathbf{U}_{\mathcal{F}}^{\mathrm{T}}\mathbf{e}[k] \tag{29}$$

should be able to yield faster convergence to its steady-state value than the GSP LMS in [14, 18].

As in the original NLMS algorithm [27], we also include an additional fixed convergence factor $\mu_{\mathrm{N}} \in \mathbb{R}_+$, resulting in the GSP NLMS in Algorithm 1, in which we employ the vertex-domain update equation

$$\hat{\mathbf{x}}_{\mathrm{o}}[k+1] = \hat{\mathbf{x}}_{\mathrm{o}}[k] + \mu_{\mathrm{N}}\mathbf{U}_{\mathcal{F}}(\mathbf{U}_{\mathcal{F}}^{\mathrm{T}}\mathbf{D}_{\mathcal{S}[k]}\mathbf{U}_{\mathcal{F}})^{-1}\mathbf{U}_{\mathcal{F}}^{\mathrm{T}}\mathbf{e}[k]\,. \tag{30}$$

Algorithm 1 NLMS estimation of graph signals

1: $k \leftarrow 0$
2: **while** (true) **do**
3: $\quad\mathbf{e}[k] = \mathbf{D}_{\mathcal{S}[k]}(\mathbf{x}_w[k] - \hat{\mathbf{x}}_{\mathrm{o}}[k])$
4: $\quad\hat{\mathbf{x}}_{\mathrm{o}}[k+1] = \hat{\mathbf{x}}_{\mathrm{o}}[k] + \mu_{\mathrm{N}}\mathbf{U}_{\mathcal{F}}(\mathbf{U}_{\mathcal{F}}^{\mathrm{T}}\mathbf{D}_{\mathcal{S}[k]}\mathbf{U}_{\mathcal{F}})^{-1}\mathbf{U}_{\mathcal{F}}^{\mathrm{T}}\mathbf{e}[k]$
5: $\quad k \leftarrow k+1$
6: **end**

Remark: Alternatively, a different derivation of the GSP NLMS algorithm can be obtained by solving the following constrained convex problem in the frequency domain: to minimize the distance between the current $\hat{\mathbf{s}}_{\mathcal{F}}[k]$ and the

updated estimate $\hat{\mathbf{s}}_{\mathcal{F}}[k+1]$ (minimum disturbance principle), such that the Fourier transform of the *a posteriori* error $\boldsymbol{\varepsilon}[k]$ is equal to zero on the frequency support $\mathcal{F}$. Mathematically, one has

$$\begin{aligned} &\underset{\hat{\mathbf{s}}_{\mathcal{F}}[k+1]}{\text{minimize}} \quad \|\hat{\mathbf{s}}_{\mathcal{F}}[k+1] - \hat{\mathbf{s}}_{\mathcal{F}}[k]\|_2^2 \\ &\text{subject to} \quad \mathbf{U}_{\mathcal{F}}^{\mathrm{T}}\mathbf{D}_{\mathcal{S}[k]}(\mathbf{x}_w[k] - \mathbf{U}_{\mathcal{F}}\hat{\mathbf{s}}_{\mathcal{F}}[k+1]) = \mathbf{0}\,, \end{aligned} \tag{31}$$

whose solution $\hat{\mathbf{s}}_{\mathcal{F}}[k+1]$ is given by (29).

4.2. Stability and Convergence to Unbiased Solution

As the GSP NLMS in Algorithm 1 allows one to select different normalized convergence factors for controlling the trade-off between convergence speed and steady-state FoM values, it is essential to determine for which range of μ_{N} values $\hat{\boldsymbol{s}}_{\mathcal{F}}[k]$ is guaranteed to be asymptotically unbiased. This information is summarized, along with further results, in Proposition 1.

Proposition 1. *For time-invariant reference graph signal $\mathbf{x}_{\mathrm{o}}[k] = \mathbf{x}_{\mathrm{o}}$ and noise covariance matrix $\mathbf{C}_w[k] = \mathbf{C}_w$, the GSP NLMS algorithm converges to an unbiased solution when its factor μ_{N} is selected in the range $0 < \mu_{\mathrm{N}} < 2$. Additionally, by assuming that $\mathcal{S}[k]$ tends to a constant sampling set $\mathcal{S}$, for such μ_{N} the matrix $\mathbb{E}[\Delta\hat{\boldsymbol{s}}_{\mathcal{F}}[k]\Delta\hat{\boldsymbol{s}}_{\mathcal{F}}^{\mathrm{T}}[k]]$ converges to*

$$\mathbf{S}_{\mathrm{N}}^{*} = \mathbb{E}[\Delta\hat{\boldsymbol{s}}_{\mathcal{F}}[\infty]\Delta\hat{\boldsymbol{s}}_{\mathcal{F}}^{\mathrm{T}}[\infty]] = \frac{\mu_{\mathrm{N}}}{2-\mu_{\mathrm{N}}}\mathbf{M}\mathbf{U}_{\mathcal{F}}^{\mathrm{T}}\mathbf{D}_{\mathcal{S}}\mathbf{C}_w\mathbf{D}_{\mathcal{S}}\mathbf{U}_{\mathcal{F}}\mathbf{M}\,. \tag{32}$$

Proof. See Appendix B. □

Therefore, the μ_{N} parameter range that assures a stable behavior for the proposed NLMS algorithm is more straightforward than the range predicted in [14] for the GSP LMS algorithm, which depends on the eigenvalues of $\mathbf{U}_{\mathcal{F}}^{\mathrm{T}}\mathbf{D}_{\mathcal{S}[k]}\mathbf{U}_{\mathcal{F}}$. In fact, this well defined convergence range is also an advantage of the traditional NLMS algorithm in comparison to its LMS counterpart [26]-[27]. Furthermore, the convergence range of the traditional NLMS algorithm agrees with the predicted interval of $0 < \mu_{\mathrm{N}} < 2$ for the GSP NLMS algorithm.

4.3. Computational Complexity Analysis

In order to provide a fair comparison of the computational complexity among the GSP LMS, RLS, and NLMS algorithms, we estimate the amount of floating-point operations (FLOPs) required to evaluate the estimate $\hat{\mathbf{x}}_{\mathrm{o}}[k+1]$ at each iteration. As all algorithms present some common steps, we focus on the differences among them, which consist basically in how the update of $\hat{\mathbf{x}}_{\mathrm{o}}[k+1]$ is performed.

For the sake of comparison, we consider that the product between a $T_1 \times T_2$ matrix and a $T_2 \times T_3$ matrix results in $T_1T_2T_3$ multiplications and $T_1T_3(T_2-1)$ sums, or a total of $T_1T_3(2T_2-1)$ FLOPs. In particular, by taking two vectors $\mathbf{b}, \mathbf{c} \in \mathbb{R}^N$, with one vector having at most $|\mathcal{S}|$ non-zero elements, as the sampled

Table 1: Algorithms' complexity for computing $\hat{\mathbf{x}}_{\mathrm{o}}[k+1]$

Algorithm	$\hat{\mathbf{x}}_{\mathrm{o}}[k+1]$	FLOPs/iter.
LMS	(10)	$\lvert\mathcal{F}\rvert \left[2(\lvert\mathcal{S}\rvert + N) - 1\right] + N$
RLS	(14)	$\frac{1}{3}\lvert\mathcal{F}\rvert^3 + \lvert\mathcal{F}\rvert^2(2\lvert\mathcal{S}\rvert + 3) + 2\lvert\mathcal{F}\rvert\,(N\lvert\mathcal{S}\rvert + \lvert\mathcal{S}\rvert - 1) + \lvert\mathcal{F}\rvert N$
NLMS	(30)	$\frac{1}{3}\lvert\mathcal{F}\rvert^3 + \lvert\mathcal{F}\rvert^2(2\lvert\mathcal{S}\rvert + 1) + 2\lvert\mathcal{F}\rvert\,(N + \lvert\mathcal{S}\rvert - 1) + N$

error vector $\mathbf{e}[k]$ in (8), we assume that an inner product operation $\mathbf{b}^{\mathrm{T}}\mathbf{c}$ can be efficiently computed in $|\mathcal{S}|$ FLOPs. Then, the cost for computing $\mathbf{U}_{\mathcal{F}}^{\mathrm{T}}\mathbf{e}[k]$ is $|\mathcal{F}|(2|\mathcal{S}| - 1)$ operations. Similarly, due to the reduced complexity required by the sampling operation $\mathbf{D}_{\mathcal{S}[k]}\mathbf{b}$, which is estimated as 1 FLOP, it is straightforward to conclude that $\mathbf{D}_{\mathcal{S}[k]}\mathbf{U}_{\mathcal{F}}$ demands $|\mathcal{F}|$ FLOPs, while $\mathbf{U}_{\mathcal{F}}^{\mathrm{T}}\mathbf{D}_{\mathcal{S}[k]}\mathbf{U}_{\mathcal{F}}$ accounts for $|\mathcal{F}|^2(2|\mathcal{S}| - 1)$ operations. Likewise, when evaluating (14), the product of $\mathbf{U}_{\mathcal{F}}^{\mathrm{T}}\mathbf{D}_{\mathcal{S}[k]}\mathbf{C}_w^{-1}[k]$ by $\mathbf{e}[k]$ requires $|\mathcal{F}|(2|\mathcal{S}| - 1)$ FLOPs. In addition, it is assumed that the inversion of a $|\mathcal{F}| \times |\mathcal{F}|$ matrix like $\mathbf{U}_{\mathcal{F}}^{\mathrm{T}}\mathbf{D}_{\mathcal{S}[k]}\mathbf{U}_{\mathcal{F}}$ adds $\frac{1}{3}|\mathcal{F}|^3$ FLOPs per iteration via Cholesky factorization [39]. Thus, the computational complexity, in terms of FLOPs, for obtaining the estimate update when using the GSP LMS, RLS, and NLMS adaptive algorithms is summarized in Table 1.

One can observe in Table 1 that the computational complexity required for evaluating the GSP LMS update in (10) is significantly smaller in comparison to the other adaptive algorithms because it is obtained by simple matrix-vector products. On the other hand, the computation of the RLS and NLMS expressions in (14) and (30), respectively, is more complex since it requires matrix-matrix products and the inversion of a $|\mathcal{F}| \times |\mathcal{F}|$ matrix; yet, the proposed GSP NLMS algorithm requires less FLOPs than the GSP RLS algorithm. Additionally, it is worth mentioning that though the GSP RLS and NLMS algorithms are more complex than the GSP LMS, when the graph signal is properly represented by a few frequency components ($|\mathcal{F}| \ll N$), the order of complexity of both algorithms is also dominated by a linear term in N.

A scenario of practical interest occurs when the sampling set $\mathcal{S}[k]$ happens to be static and known *a priori*, as in the case of interpolation problems. In these cases, the evaluation of the GSP NLMS update in (30) is considerably simplified by rewriting the constant matrix $(\mathbf{U}_{\mathcal{F}}^{\mathrm{T}}\mathbf{D}_{\mathcal{S}}\mathbf{U}_{\mathcal{F}})^{-1}$ as $\mathbf{L}_{\mathrm{N}}\mathbf{L}_{\mathrm{N}}^{\mathrm{T}}$, where $\mathbf{L}_{\mathrm{N}} \in \mathbb{R}^{|\mathcal{F}|\times|\mathcal{F}|}$ is a lower triangular matrix obtained using the Cholesky decomposition. Then, after defining the ancillary matrix $\mathbf{B}_{\mathrm{N}} \in \mathbb{R}^{N\times|\mathcal{F}|}$ as $\mathbf{B}_{\mathrm{N}} = \mathbf{U}_{\mathcal{F}}\mathbf{L}_{\mathrm{N}}$, one verifies that the GSP NLMS update can be implemented in static sampling set scenarios, where $\mathcal{S}[k] = \mathcal{S}$ is known *a priori*, according to the expression

$$\hat{\mathbf{x}}_{\mathrm{o}}[k+1] = \hat{\mathbf{x}}_{\mathrm{o}}[k] + \mu_{\mathrm{N}}\mathbf{B}_{\mathrm{N}}\mathbf{B}_{\mathrm{N}}^{\mathrm{T}}\mathbf{e}[k]\,. \tag{33}$$

Therefore, by considering that matrix $\mathbf{B}_{\mathrm{N}}$ in (33) is a pre-evaluated structure stored for efficient algorithm implementation, one easily concludes from comparing (10) to (33) that the complexity required for the NLMS algorithm

resembles that of the LMS method in the static sampling set scenario, being both given by $|\mathcal{F}|\left[2(|\mathcal{S}|+N)-1\right]+N$ FLOPs per iteration.

4.4. Steady-State FoM Analysis

By assuming that $\mathcal{S}[k] \to \mathcal{S}$ and $\mathbf{C}_w[k] \to \mathbf{C}_w$, in this subsection we derive the steady-state values of the FoMs $\sigma^2_{e_n}$, MSE^*_{G}, and MSD^*_{G} discussed in Subsection 3.3, when using the GSP NLMS.

From (21) and (32), and by defining $\sigma^2_{w_n} = \mathbb{E}\{w^2_n[k]\}$, the steady-state value for $\sigma^2_{e_n}$ is given by

$$\sigma^2_{e_n} = d_n\left[\sigma^2_{w_n} + \frac{\mu_{\text{N}}}{2-\mu_{\text{N}}}\mathbf{u}^{\text{T}}_{n_{\mathcal{F}}}\mathbf{M}\mathbf{U}^{\text{T}}_{\mathcal{F}}\mathbf{D}_{\mathcal{S}}\mathbf{C}_w\mathbf{D}_{\mathcal{S}}\mathbf{U}_{\mathcal{F}}\mathbf{M}\mathbf{u}_{n_{\mathcal{F}}}\right]. \tag{34}$$

Moreover, according to (22), we find the MSE^*_{G} for the GSP NLMS by simply summing $\sigma^2_{e_n}$ for all $n \in \mathcal{N}$.

Finally, based on (23) and (32), the MSD^*_{G} is

$$\text{MSD}^*_{\text{G}} = \frac{\mu_{\text{N}}}{2-\mu_{\text{N}}}\text{tr}\left\{\mathbf{M}\mathbf{U}^{\text{T}}_{\mathcal{F}}\mathbf{D}_{\mathcal{S}}\mathbf{C}_w\mathbf{D}_{\mathcal{S}}\mathbf{U}_{\mathcal{F}}\mathbf{M}\right\}. \tag{35}$$

4.5. Remarks

Let us get a better feeling regarding the effect of matrix $\mathbf{M}[k]$ by comparing the GSP LMS and NLMS update equations. As $\{\mathbf{u}_n\}$ is a basis of $\mathbb{R}^N$, then there exists $\boldsymbol{\alpha}[k] \in \mathbb{R}^N$ such that $\mathbf{e}[k] = \mathbf{D}_{\mathcal{S}[k]}\mathbf{U}\boldsymbol{\alpha}[k]$. Without loss of generality, we can write $\mathbf{U} = [\mathbf{U}_{\mathcal{F}} \;\; \mathbf{U}_{\overline{\mathcal{F}}}]$ and $\boldsymbol{\alpha}[k] = [\boldsymbol{\alpha}^{\text{T}}_{\mathcal{F}}[k] \;\; \boldsymbol{\alpha}^{\text{T}}_{\overline{\mathcal{F}}}[k]]^{\text{T}}$, where $\overline{\mathcal{F}} = \mathcal{N} \setminus \mathcal{F}$. Note that $\boldsymbol{\alpha}[k]$ is the frequency-domain representation of $\mathbf{w}[k] + (\mathbf{x}_{\text{o}}[k] - \hat{\mathbf{x}}_{\text{o}}[k])$. In this case, vector $(\mathbf{x}_{\text{o}}[k] - \hat{\mathbf{x}}_{\text{o}}[k])$ is $\mathcal{F}$-ssparse, which means that $\boldsymbol{\alpha}_{\overline{\mathcal{F}}}[k]$ has only contributions from the measurement noise.

In the GSP LMS algorithm, the error signal $\mathbf{e}[k]$ is multiplied by matrix $\mu_{\text{L}}\mathbf{B}_{\text{L}} = \mu_{\text{L}}\mathbf{U}_{\mathcal{F}}\mathbf{U}^{\text{T}}_{\mathcal{F}}$, thus yielding, in the frequency domain, the correction term

$$\mu_{\text{L}}\mathbf{M}^{-1}[k]\boldsymbol{\alpha}_{\mathcal{F}}[k] + \mathbf{w}_{\text{L}}[k]\,, \tag{36}$$

where $\mathbf{w}_{\text{L}}[k] = \mu_{\text{L}}\left(\mathbf{U}^{\text{T}}_{\mathcal{F}}\mathbf{D}_{\mathcal{S}[k]}\mathbf{U}_{\overline{\mathcal{F}}}\right)\boldsymbol{\alpha}_{\overline{\mathcal{F}}}[k]$ is essentially noise after processing.[1] As for the GSP NLMS algorithm, the error signal is multiplied by matrix $\mu_{\text{N}}\mathbf{B}_{\text{N}}[k] = \mu_{\text{N}}\mathbf{U}_{\mathcal{F}}\mathbf{M}[k]\mathbf{U}^{\text{T}}_{\mathcal{F}}$, thus yielding, again in the frequency domain, the correction term

$$\mu_{\text{N}}\boldsymbol{\alpha}_{\mathcal{F}}[k] + \mathbf{w}_{\text{N}}[k]\,, \tag{37}$$

where $\mathbf{w}_{\text{N}}[k] = \mu_{\text{N}}\mathbf{M}[k]\left(\mathbf{U}^{\text{T}}_{\mathcal{F}}\mathbf{D}_{\mathcal{S}[k]}\mathbf{U}_{\overline{\mathcal{F}}}\right)\boldsymbol{\alpha}_{\overline{\mathcal{F}}}[k]$.

By comparing expressions (36) and (37) one can see that the estimation error within the frequency support $\mathcal{F}$ in the NLMS has a clean and direct impact on the correction of the previous estimate $\hat{\mathbf{s}}_{\mathcal{F}}[k]$, without distortions as in the

[1] Roughly speaking, $\mathbf{U}^{\text{T}}_{\mathcal{F}}\mathbf{D}_{\mathcal{S}[k]}\mathbf{U}_{\overline{\mathcal{F}}}$ tends to be close to $\mathbf{0}$, since $\mathbf{U}^{\text{T}}_{\mathcal{F}}\mathbf{U}_{\overline{\mathcal{F}}} = \mathbf{0}$.

LMS case, imposed by $\mathbf{M}^{-1}[k]$. The so-called normalization of the NLMS is responsible for this effect, which turns out to be a key aspect for enhancing the algorithm performance, as will be clear in Section 7.

In addition, it is noticeable that the proposed NLMS update equation in (30) resembles the RLS long-term expression in (16), thus indicating that the inclusion of $\mathbf{M}$ brings about some RLS-like features to the resulting algorithm. Particularly, when the covariance matrix is given by $\mathbf{C}_w = \sigma_w^2\mathbf{I}$, with $\sigma_w^2 > 0$, and $\mathcal{S}[k] = \mathcal{S}$, both algorithms present an equivalent performance for large k if the NLMS convergence factor μ_{N} and the RLS forgetting factor β_{R} follow the relation

$$\mu_{\mathrm{N}} = 1 - \beta_{\mathrm{R}}\,. \tag{38}$$

Differently from its traditional counterpart, whose normalization term is usually time-varying since it depends on the system input signal, in static sampling set scenarios the GSP NLMS algorithm relies on a fixed normalization term $\mathbf{M}$ for a constant graph structure, yielding an update expression that requires the same computational complexity as the GSP LMS algorithm. Besides, the GSP NLMS has a strong practical appeal since it presents a well-defined range of values for its convergence factor μ_{N} that guarantees the method stability, in opposition to the equivalent factor choice for the LMS algorithm [14].

5. GSP LMS and RLS Complementary Analysis

The analyses of the GSP LMS and RLS algorithms in [18] cover the possibility of signal reconstruction via sparse sampling and the MSD analysis, along with of optimal sampling proposals. Here, we assume that $\mathcal{S}[k] \to \mathcal{S}$ and extend those analyses to incorporate the steady-state FoMs $\sigma_{e_n}^2$ and $\mathrm{MSE}_{\mathrm{G}}$ from Subsection 3.3. Based on the accurate estimates for the error variances evaluated in this section, the GSP LMS and RLS algorithms can also be used with data-selection strategies, as we shall see in Section 6.

5.1. LMS Algorithm Error Analysis

From (10) and (18), one can write

$$\Delta\hat{\boldsymbol{s}}_{\mathcal{F}}[k+1] = (\mathbf{I} - \mu_{\mathrm{L}}\mathbf{U}_{\mathcal{F}}^{\mathrm{T}}\mathbf{D}_{\mathcal{S}}\mathbf{U}_{\mathcal{F}})\Delta\hat{\boldsymbol{s}}_{\mathcal{F}}[k] + \mu_{\mathrm{L}}\mathbf{U}_{\mathcal{F}}^{\mathrm{T}}\mathbf{D}_{\mathcal{S}}\boldsymbol{w}[k]\,, \tag{39}$$

thus implying that

$$\begin{aligned}\mathbb{E}[\Delta\hat{\boldsymbol{s}}_{\mathcal{F}}[k+1]\Delta\hat{\boldsymbol{s}}_{\mathcal{F}}^{\mathrm{T}}[k+1]] &= \mu_{\mathrm{L}}^2\mathbf{U}_{\mathcal{F}}^{\mathrm{T}}\mathbf{D}_{\mathcal{S}}\mathbf{C}_w\mathbf{D}_{\mathcal{S}}\mathbf{U}_{\mathcal{F}} + \\ &+(\mathbf{I}-\mu_{\mathrm{L}}\mathbf{U}_{\mathcal{F}}^{\mathrm{T}}\mathbf{D}_{\mathcal{S}}\mathbf{U}_{\mathcal{F}})\mathbb{E}[\Delta\hat{\boldsymbol{s}}_{\mathcal{F}}[k]\Delta\hat{\boldsymbol{s}}_{\mathcal{F}}^{\mathrm{T}}[k]](\mathbf{I}-\mu_{\mathrm{L}}\mathbf{U}_{\mathcal{F}}^{\mathrm{T}}\mathbf{D}_{\mathcal{S}}\mathbf{U}_{\mathcal{F}})\,.\end{aligned} \tag{40}$$

If μ_{L} is in the range that guarantees the algorithm stability [14], then matrix $\mathbb{E}[\Delta\hat{\mathbf{s}}_{\mathcal{F}}[k]\Delta\hat{\mathbf{s}}_{\mathcal{F}}^{\mathrm{T}}[k]]$ converges to $\mathbf{S}_{\mathrm{L}}^{*} \in \mathbb{R}^{|\mathcal{F}|\times|\mathcal{F}|}$ when $k \to \infty$. Thus, by defining $\mathbf{P}, \mathbf{R} \in \mathbb{R}^{|\mathcal{F}|\times|\mathcal{F}|}$ such that

$$\mathbf{P} = \mathbf{U}_{\mathcal{F}}^{\mathrm{T}}\mathbf{D}_{\mathcal{S}}\mathbf{U}_{\mathcal{F}} \quad \text{and} \quad \mathbf{R} = \mathbf{U}_{\mathcal{F}}^{\mathrm{T}}\mathbf{D}_{\mathcal{S}}\mathbf{C}_w\mathbf{D}_{\mathcal{S}}\mathbf{U}_{\mathcal{F}}\,, \tag{41}$$

one can write the steady-state form of expression (40) as

$$\mathbf{P}\mathbf{S}_{\mathrm{L}}^{*}+\mathbf{S}_{\mathrm{L}}^{*}\mathbf{P}-\mu_{\mathrm{L}}\mathbf{P}\mathbf{S}_{\mathrm{L}}^{*}\mathbf{P}=\mu_{\mathrm{L}}\mathbf{R}\,, \tag{42}$$

which is the generalized Lyapunov matrix equation [40], being equivalent to

$$[(\mathbf{I}\otimes\mathbf{P})+(\mathbf{P}\otimes\mathbf{I})-\mu_{\mathrm{L}}(\mathbf{P}\otimes\mathbf{P})]\,\mathrm{vec}(\mathbf{S}_{\mathrm{L}}^{*})=\mu_{\mathrm{L}}\,\mathrm{vec}(\mathbf{R}), \tag{43}$$

where $\otimes$ indicates the Kronecker product [41] and $\mathrm{vec}(\mathbf{S}_{\mathrm{L}}^{*})$ represents the vectorization of $\mathbf{S}_{\mathrm{L}}^{*}$, performed by stacking its columns into a single column vector. When the left-hand side matrix of (43) has full rank, $\mathrm{vec}(\mathbf{S}_{\mathrm{L}}^{*})$ is obtained by solving

$$\mathrm{vec}(\mathbf{S}_{\mathrm{L}}^{*})=\mu_{\mathrm{L}}\,[(\mathbf{I}\otimes\mathbf{P})+(\mathbf{P}\otimes\mathbf{I})-\mu_{\mathrm{L}}(\mathbf{P}\otimes\mathbf{P})]^{-1}\mathrm{vec}(\mathbf{R}). \tag{44}$$

After recovering matrix $\mathbf{S}_{\mathrm{L}}^{*}$ from its vectorized version, the n^{th} variance $\sigma_{e_n}^{2}$ for the GSP LMS algorithm is computed by replacing $\mathbf{S}_{\mathrm{L}}^{*}$ in (21), yielding

$$\sigma_{e_n}^{2}=d_n\,(\,\sigma_{w_n}^{2}+\mathbf{u}_{n_{\mathcal{F}}}^{\mathrm{T}}\mathbf{S}_{\mathrm{L}}^{*}\mathbf{u}_{n_{\mathcal{F}}})\,, \tag{45}$$

and from (22)

$$\mathrm{MSE}_{\mathrm{G}}^{*}=\sum_{i=1}^{N}d_n\left(\sigma_{w_n}^{2}+\mathbf{u}_{n_{\mathcal{F}}}^{\mathrm{T}}\mathbf{S}_{\mathrm{L}}^{*}\mathbf{u}_{n_{\mathcal{F}}}\right)\,. \tag{46}$$

Moreover, an additional result that comes straightforwardly from the knowledge of $\mathbf{S}_{\mathrm{L}}^{*}$ is the $\mathrm{MSD}_{\mathrm{G}}^{*}$ in (23), so that

$$\mathrm{MSD}_{\mathrm{G}}^{*}=\mathrm{tr}\{\mathbf{S}_{\mathrm{L}}\}\,. \tag{47}$$

Although a steady-state $\mathrm{MSD}_{\mathrm{G}}$ analysis for the LMS algorithm is presented in [14, 18], it requires μ_{L} to be relatively small. On the other hand, as the derivation of (47) does not rely on this assumption, it is more general, being valid for any μ_{L} that guarantees the algorithm stability.

5.2. RLS Algorithm Error Analysis

Similarly, the GSP RLS analysis starts by rewriting (16) as

$$\Delta\hat{\boldsymbol{s}}_{\mathcal{F}}[k+1]=\beta_{\mathrm{R}}\Delta\hat{\boldsymbol{s}}_{\mathcal{F}}[k]+(1-\beta_{\mathrm{R}})\mathbf{M}'\mathbf{U}_{\mathcal{F}}^{\mathrm{T}}\mathbf{D}_{\mathcal{S}}\mathbf{C}_{w}^{-1}\boldsymbol{w}[k]\,, \tag{48}$$

where we recall that $\mathbf{M}'=(\mathbf{U}_{\mathcal{F}}^{\mathrm{T}}\mathbf{D}_{\mathcal{S}}\mathbf{C}_{w}^{-1}\mathbf{D}_{\mathcal{S}}\mathbf{U}_{\mathcal{F}})^{-1}$. Thus, it implies that

$$\begin{aligned}\mathbb{E}[\Delta\hat{\boldsymbol{s}}_{\mathcal{F}}[k+1]\Delta\hat{\boldsymbol{s}}_{\mathcal{F}}^{\mathrm{T}}[k+1]]&=\beta_{\mathrm{R}}^{2}\mathbb{E}[\Delta\hat{\boldsymbol{s}}_{\mathcal{F}}[k]\Delta\hat{\boldsymbol{s}}_{\mathcal{F}}^{\mathrm{T}}[k]]+\\&+(1-\beta_{\mathrm{R}})^{2}\mathbf{M}'\mathbf{U}_{\mathcal{F}}^{\mathrm{T}}\mathbf{D}_{\mathcal{S}}\mathbf{C}_{w}^{-1}\mathbf{D}_{\mathcal{S}}\mathbf{C}_{w}\mathbf{D}_{\mathcal{S}}\mathbf{C}_{w}^{-1}\mathbf{D}_{\mathcal{S}}\mathbf{U}_{\mathcal{F}}\mathbf{M}'\,.\end{aligned} \tag{49}$$

Then, by considering the convergence of $\mathbb{E}[\Delta\hat{\mathbf{s}}_{\mathcal{F}}[k]\Delta\hat{\mathbf{s}}_{\mathcal{F}}^{\mathrm{T}}[k]]$ to $\mathbf{S}_{\mathrm{R}}^{*}\in\mathbb{R}^{|\mathcal{F}|\times|\mathcal{F}|}$ as k grows to infinity, one gets

$$\mathbf{S}_{\mathrm{R}}^{*}=\frac{1-\beta_{\mathrm{R}}}{1+\beta_{\mathrm{R}}}\cdot\mathbf{M}'\mathbf{U}_{\mathcal{F}}^{\mathrm{T}}\mathbf{D}_{\mathcal{S}}\mathbf{C}_{w}^{-1}\mathbf{D}_{\mathcal{S}}\mathbf{C}_{w}\mathbf{D}_{\mathcal{S}}\mathbf{C}_{w}^{-1}\mathbf{D}_{\mathcal{S}}\mathbf{U}_{\mathcal{F}}\mathbf{M}'\,, \tag{50}$$

allowing us to write

$$\sigma_{e_n}^2 = d_n \left(\sigma_{w_n}^2 + \mathbf{u}_{n_{\mathcal{F}}}^{\mathrm{T}} \mathbf{S}_{\mathrm{R}}^{*} \mathbf{u}_{n_{\mathcal{F}}}\right) , \tag{51}$$

$$\mathrm{MSE}_{\mathrm{G}}^{*} = \sum_{n=1}^{N} d_n \left(\sigma_{w_n}^2 + \mathbf{u}_{n_{\mathcal{F}}}^{\mathrm{T}} \mathbf{S}_{\mathrm{R}}^{*} \mathbf{u}_{n_{\mathcal{F}}}\right) , \tag{52}$$

$$\mathrm{MSD}_{\mathrm{G}}^{*} = \mathrm{tr}\left\{\mathbf{S}_{\mathrm{R}}^{*}\right\} . \tag{53}$$

6. Time-domain Data-selective Estimation Algorithms

Based on the assumption that not all acquired data brings novelty to the current system estimate, the data-selection (DS) idea [32]-[34] labels the received information according to its usefulness and updates the algorithm parameters only when the input data is classified as valuable. This section describes time-domain DS versions of the adaptive GSP LMS, RLS, and NLMS algorithms.

As the update decision on the traditional adaptive algorithms depends on the scalar error (5) and the GS estimation problem deals with the error vector $\mathbf{e}[k]$ in (8), it is necessary to first define how to measure data novelty in the GSP scenario. Thus, we consider two approaches: the first one performs component-wise comparisons between the error vector $\mathbf{e}[k]$ and a threshold vector $\bar{\boldsymbol{\gamma}} \in \mathbb{R}_{+}^{N}$, while the second one compares a squared ℓ_2-norm error-based metric with a scalar $\bar{\gamma} \in \mathbb{R}_{+}$. These strategies are respectively called component-wise error constraint (CW-EC) and ℓ_2-norm error constraint (ℓ_2N-EC). The main difference between these complementary strategies is that the CW-EC is more concerned with local innovation checking, while the ℓ_2N-EC method provides an average assessment of data novelty, being less sensitive to local changes. Thus, selecting the most suitable approach heavily depends on the application requirements. We now detail these strategies for GS estimation and suggest practical choices for the thresholds that allow to control the mean update rate.

6.1. Component-wise Error Constraint Strategy

The first update idea consists in defining a threshold vector $\bar{\boldsymbol{\gamma}} = [\bar{\gamma}_1 \quad \bar{\gamma}_2 \quad \dots \bar{\gamma}_N]^{\mathrm{T}}$ such that the n^{th} error component $e_n[k]$ from (8) is compared to its respective bound $\bar{\gamma}_n \in \mathbb{R}_{+}$. If all absolute components $|e_n[k]|$ are smaller than their respective $\bar{\gamma}_n$, the strategy assumes the input data does not bring enough innovation to the current system. In other words, considering that function abs$(\cdot)$ performs a component-wise modulus operation on its argument vector, when

$$\mathrm{abs}(\mathbf{e}[k]) \preceq \bar{\boldsymbol{\gamma}} \tag{54}$$

is true, there is no algorithm update. Otherwise, if any $e_n[k]$ has a larger absolute value than its respective $\bar{\gamma}_n$, we evaluate the new estimate according to (10), (13), or (30), depending on the choice of the LMS, RLS, or NLMS algorithms.

Although the constraint vector $\bar{\boldsymbol{\gamma}}$ can be defined in different ways, its choice influences the algorithm behavior in terms of the update probability/rate. In

order to provide a fair estimate of the update rate for the component-wise strategy, we consider the particular scenario in which the noise signal $\boldsymbol{w}[k]$ in (7) is Gaussian[2] and assume that each component $e_n[k]$ in (20) is modeled as a zero-mean Gaussian random variable (RV) with variance $\sigma_{e_n}^2$. Based on expressions (34), (45), and (51) for the NLMS, LMS, and RLS algorithms, respectively, we find the variances $\sigma_{e_n}^2$ for the particular algorithm and define $\bar{\boldsymbol{\gamma}}$ as

$$\bar{\boldsymbol{\gamma}} = \kappa \begin{bmatrix} \sigma_{e_1} & \sigma_{e_2} & \dots & \sigma_{e_N} \end{bmatrix}^{\mathrm{T}} , \tag{55}$$

in which the so-called update factor $\kappa \in \mathbb{R}_+$ is a design parameter included to fine tune the update rate.

The probability that all error entries will be in their respect intervals $[-\kappa\,\sigma_{e_n}, \kappa\,\sigma_{e_n}]$ is $\mathrm{erf}(\kappa/\sqrt{2})^{|\mathcal{S}|}$, since only $|\mathcal{S}|$ components of $\mathbf{e}[k]$ are non-zero, and the error function $\mathrm{erf}\big(\bar{\gamma}_n/(\sqrt{2}\sigma_{e_n})\big)$ describes the probability of $e_n[k]$ to fall in the interval $[-\bar{\gamma}_n, \bar{\gamma}_n]$ [42]. However, as the update rate is the complement of this value, the update probability P_{up} for the CW-EC DS estimation algorithm is

$$P_{\mathrm{up}} = 1 - \mathrm{erf}\left(\frac{\kappa}{\sqrt{2}}\right)^{|\mathcal{S}|} , \tag{56}$$

Alternatively, if the designer expects an update rate P_{up}, we find that

$$\kappa = \sqrt{2} \cdot \mathrm{erf}^{-1}\left(\sqrt[|\mathcal{S}|]{1 - P_{\mathrm{up}}} \right) . \tag{57}$$

The proposed CW-EC strategy in Algorithm 2 adds $2|\mathcal{S}|$ FLOPs per iteration ($|\mathcal{S}|$ modulus operations and $|\mathcal{S}|$ comparisons) to the algorithm complexity due to the test condition (54). However, it provides a considerable reduction of the overall complexity by avoiding unnecessary updates when condition (54) holds.

6.2. ℓ_2-Norm Error Constraint Strategy

An alternative to the CW-EC strategy consists in representing the instantaneous error vector $\mathbf{e}[k]$ by a single scalar value, which is directly compared to a scalar threshold $\bar{\gamma} \in \mathbb{R}_+$. In order to map $\mathbf{e}[k]$ into a scalar value we first define the normalized error vector $\bar{\mathbf{e}}[k] \in \mathbb{R}^N$ according to its individual components $\bar{e}_n$ described by

$$\bar{e}_n = \begin{cases} \dfrac{e_n[k]}{\sigma_{e_n}} & , \text{if } \sigma_{e_n} \neq 0 \,, \\ 0 & , \text{otherwise} \,, \end{cases} \tag{58}$$

where σ_{e_n} comes from (34), (45), or (51). Then, we select the squared ℓ_2-norm $\|\bar{\mathbf{e}}[k]\|_2^2$ for performing the scalar mapping. The ℓ_2N-EC strategy consists in verifying if the condition

$$\|\bar{\mathbf{e}}[k]\|_2^2 \leq \bar{\gamma} \tag{59}$$

[2]The Gaussian case was chosen here to simplify the mathematical development. But the proposed methodology can be extended to other types of distributions for the noise signal.

Algorithm 2 CW-EC data-selection strategy

Define update factor $\kappa \in \mathbb{R}_+^*$ via (57)
Evaluate $\bar{\boldsymbol{\gamma}} = \kappa \begin{bmatrix} \sigma_{e_1} & \sigma_{e_2} & \dots & \sigma_{e_N} \end{bmatrix}^{\mathrm{T}}$, where σ_{e_n} is given by (34), (45), or (51)
$k \leftarrow 0$
while (true) **do**
  $\mathbf{e}[k] = \mathbf{D}_{\mathcal{S}[k]}(\mathbf{x}_w[k] - \hat{\mathbf{x}}_{\mathrm{o}}[k])$
  if ($\mathrm{abs}(\mathbf{e}[k]) \preceq \bar{\boldsymbol{\gamma}}$) **then**
    $\hat{\mathbf{x}}_{\mathrm{o}}[k+1] = \hat{\mathbf{x}}_{\mathrm{o}}[k]$
  else
    Find $\hat{\mathbf{x}}_{\mathrm{o}}[k+1]$ using (30), (10), or (13)
  $k \leftarrow k+1$
end

holds true, in which case there is no algorithm update.

By choosing an update expression based on either the NLMS, LMS, or RLS algorithms, we once again consider the particular case of a Gaussian input noise and assume that each $e_n[k]$ is modeled as a zero-mean Gaussian RV with variance $\sigma_{e_n}^2$. As the square of a normal RV results in a chi-squared RV χ_1^2 with one-degree of freedom [43], then $\|\bar{\mathbf{e}}[k]\|_2^2$ is described by a $\chi_{|\mathcal{S}|}^2$ distribution, i.e., a chi-squared distribution with $|\mathcal{S}|$ degrees of freedom.

For an update factor $\kappa \in \mathbb{R}_+$, if we consider the threshold value

$$\bar{\gamma} = \kappa \, |\mathcal{S}| \, , \tag{60}$$

and remember the cumulative distribution function (CDF) of a chi-squared distribution with $|\mathcal{S}|$ degrees of freedom [43], the probability P_{up} for the ℓ_2N-EC strategy is estimated as

$$P_{\mathrm{up}} = \frac{\Gamma_{\mathrm{i}}(0.5\kappa|\mathcal{S}|)}{\Gamma(0.5|\mathcal{S}|)} \, , \tag{61}$$

where $\Gamma(\cdot)$ denotes the standard gamma function, and $\Gamma_{\mathrm{i}}(0.5\kappa|\mathcal{S}|) = \int_{0.5\kappa|\mathcal{S}|}^{\infty} t^{0.5|\mathcal{S}|-1} \mathrm{e}^{-t} \mathrm{d}t$ is an upper incomplete gamma function. Alternatively, if the designer expects a P_{up} update rate, we find that

$$\kappa = \frac{2}{|\mathcal{S}|} \Gamma_{\mathrm{i}}^{-1} \left(P_{\mathrm{up}} \cdot \Gamma(0.5|\mathcal{S}|) \right) \, . \tag{62}$$

Algorithm 3 summarizes the proposed ℓ_2N-EC DS strategy.

7. Numerical Simulations

We performed numerical experiments using real-world-based scenarios that require the estimation of temperature values according to noisy measurements from nearby weather stations. Since the underlying GSs have spatial correlation,

Algorithm 3 ℓ_2N-EC data-selection strategy

Define update factor $\kappa \in \mathbb{R}_+^*$ via (62)
Evaluate $\bar{\gamma} = \kappa\,|\mathcal{S}|$
$k \leftarrow 0$
while (true) **do**
  $\mathbf{e}[k] = \mathbf{D}_{\mathcal{S}[k]}(\mathbf{x}_w[k] - \hat{\mathbf{x}}_{\mathrm{o}}[k])$
  Obtain $\bar{\mathbf{e}}[k]$, whose entries $\bar{e}_n[k]$ are given by (58)
  if ($\|\bar{\mathbf{e}}[k]\|_2^2 \leq \bar{\gamma}$) **then**
    $\hat{\mathbf{x}}_{\mathrm{o}}[k+1] = \hat{\mathbf{x}}_{\mathrm{o}}[k]$
  else
    Find $\hat{\mathbf{x}}_{\mathrm{o}}[k+1]$ using (30), (10), or (13)
  $k \leftarrow k+1$
end

which is illustrated by the smooth temperature variations in Figure 1, we recast these problems into a bandlimited GS online reconstruction context, where we used the NLMS, LMS, and RLS adaptive algorithms discussed in this work to estimate the unknown temperature measurements.

For assessing the theoretical predictions from Sections 4-6, we first consider a partially synthetic graph signal in Subsection 7.1. Then, for comparing the behavior of the adaptive algorithms in tracking real data, Subsection 7.2 provides simulations with static and time-varying reference graph signals. All the experiments discussed in this section have been implemented in a MATLAB environment and the scripts are available at [44].

7.1. Brazilian Climate Normals Dataset

Based on the monthly average temperature dataset for Brazilian weather stations in the 1961-1990 period [22]-[23], we obtained a graph with $N = 299$ nodes where each vertex spatial location is described by its latitude and longitude coordinates. Using these geographical coordinates, the graph structure was formed by connecting each node to its 8 closest neighbors and assigning the edge weights $\{a_{nm}\}$ according to the Gaussian kernel weighting function [3]

$$a_{nm} = \begin{cases} \exp\Big(-\dfrac{d_{\mathrm{H}}(n,m)^2}{2\theta^2}\Big), & \text{if } \widehat{v_n v_m} \in \mathcal{E}, \\ 0, & \text{otherwise,} \end{cases} \tag{63}$$

where $d_{\mathrm{H}}(n,m)$ is the Haversine distance between vertices v_n and v_m, θ was taken as $2 \cdot 10^3$ and the condition $\widehat{v_n v_m} \in \mathcal{E}$ checks if the edge connecting nodes v_n and v_m is part of the set $\mathcal{E}$.

After verifying that the original temperature measurements comprise an *approximate bandlimited* GS, i.e., that the majority of their spectrum is concentrated on a limited number of frequencies, we assumed that a reconstruction

error of 2.5% satisfies the current application requirement. Based on this assumption, we selected the 200 most relevant frequencies of the GS frequency-domain representation of the July measurements (Figure 1) to define $\mathcal{F}$. The sampling set size was taken as $|\mathcal{S}| = 210$, and the indices of this set were obtained by a greedy approach that maximizes the minimum non-negative eigenvalue of $(\mathbf{U}_{\mathcal{F}}^{\mathrm{T}}\mathbf{D}_{\mathcal{S}\cup\{n\}}\mathbf{U}_{\mathcal{F}})$ at each internal iteration n, a procedure suggested in [10, 14] and discussed in [21], eventually yielding a static set $\mathcal{S}[k] = \mathcal{S}$.

We then took the indices in $\mathcal{F}$ of the original GS representation $\mathbf{s}$ from (1) to obtain $\mathbf{s}_{\mathcal{F}} \in \mathbb{R}^{|\mathcal{F}|}$ and defined an vertex-domain bandlimited GS $\mathbf{x}_{\mathrm{o}}$ as (3). This bandlimited approximation was performed with a didactic intention since we are more concerned in evaluating the behavior of the GSP NLMS, LMS, and RLS adaptive algorithms in this subsection, rather than in directly dealing with the original signal (as done in Subsection 7.2). For an explicit definition of the bandlimited reference $\mathbf{x}_{\mathrm{o}}$, the edge weights forming the adjacency matrix $\mathbf{A}$, the frequency set $\mathcal{F}$, the sampling set $\mathcal{S}$, and their respective matrices $\mathbf{U}_{\mathcal{F}}$ and $\mathbf{D}_{\mathcal{S}}$, see [44].

7.1.1. Noise Scenarios

In order to evaluate the adaptive GSP algorithms in different noise scenarios, we use a generic covariance matrix $\mathbf{C}_w$ as

$$\mathbf{C}_w = \mathrm{diag}\big(\sigma_{w_a}^2\,\mathbf{1} + \sigma_{w_b}^2\,\mathbf{r}_w\big)\,, \tag{64}$$

where $\mathbf{1} \in \mathbb{R}^N$ is a vector with all components equal to 1, $\mathbf{r}_w \in \mathbb{R}^N$ is a realization of a random vector whose entries follow a uniform distribution between $[0, 1]$, and $\sigma_{w_a}^2, \sigma_{w_b}^2 \in \mathbb{R}_+$ are variances that scale the elements of $\mathbf{1}$ and $\mathbf{r}_w$, respectively.

Although the theoretical predictions provided are valid for any symmetric $\mathbf{C}_w$, we focus on diagonal matrices because we do not expect the influence of correlated noise on more than one graph vertex measurement, since we consider sensor positions that are not close to each other. For $\mathbf{C}_w$ in (64), $\sigma_{w_a}^2$ represents the noise variance observed in all nodes, which might be caused by the same type of sensor device being used in all locations, while $\sigma_{w_b}^2$ accounts for the noise variance differences among nodes.

Thus, we defined the noise signal $\boldsymbol{w}[k]$ used for each simulation as zero-mean Gaussian according to three scenarios:

(i) $\sigma_{w_a}^2 = 0.001$ and $\sigma_{w_b}^2 = 0.000$;
(ii) $\sigma_{w_a}^2 = 0.010$ and $\sigma_{w_b}^2 = 0.000$; and
(iii) $\sigma_{w_a}^2 = 0.005$ and $\sigma_{w_b}^2 = 0.010$.

7.1.2. Convergence Speed and Complexity Comparison

We performed numerical simulations based on the noise scenarios (i) and (iii) to analyze the overall performance of the proposed GSP NLMS algorithm, when using both the general implementation (30) and the particular approach (33), in comparison to the LMS [14] and RLS [17] strategies. For each noise scenario we

used the three adaptive GSP algorithms, with respective convergence/forgetting factors adjusted in order to provide a similar $\text{MSD}_{\text{G}}^{*}$. At each simulation run we evaluated 5000 iterations, where we scaled the reference GS $\mathbf{x}_{\text{o}}$ by a 1.2 factor at $k = 2500$ to observe the algorithms' tracking abilities. We considered that: the algorithm has converged after reaching $1.025 \cdot \text{MSD}_{\text{G}}^{*}$ for the first time, the steady-state FoMs are computed using the last 1000 iterations of each run, and the update time uses the "*tic/toc*" MATLAB functions to provide the reader with a rough idea of how long it takes to compute $\hat{\mathbf{x}}_{\text{o}}[k+1]$ for each algorithm in our simulations. Based on the average values of a 1000-run ensemble, we obtained the numerical results presented in Table 2.

From Table 2 we observe that the GSP NLMS algorithm converges considerably (more than 10 times) faster than the LMS algorithm, but slightly (about twice) slower than the RLS algorithm. This convergence speed comparison is made clear by the $\text{MSD}_{\text{G}}[k]$ plots in Figure 2, where we only compare the NLMS and LMS methods in Figure 2a and display the three algorithms in Figure 2b. A particular point about Figure 2b is that the transition at $k = 2500$ indicates that the NLMS algorithm behaves like the RLS for large k, as pointed out in Subsection 4.5. Another conclusion from Table 2 is that the computation complexity for performing the GSP LMS update is noticeably smaller than the one required for computing the general implementation of the GSP NLMS in (30), which is still about 2.9 times faster than the RLS approach.[3] In particular, one verifies in Table 2 that the GSP NLMS approach in (33) provides the same fast converging characteristic of its general implementation while being as complex as the GSP LMS algorithm, however, it requires $\mathcal{S}[k]$ to be static and *a priori* known. Additionally, it is worth mentioning that the large difference in update time among the adaptive algorithms occurs because the current simulation scenario does not rely on the condition $|\mathcal{F}| \ll N$ ($|\mathcal{F}| \approx 0.7N$), where the dominant term in Table 1 becomes N.

7.1.3. Steady-State FoM Predictions

Next, we investigate the accuracy of the steady-state $\text{MSE}_{\text{G}}^{*}$ and $\text{MSD}_{\text{G}}^{*}$ predicted for the NLMS algorithm in Subsection 4.4 and for the LMS and RLS algorithms in Section 5. By using the noise scenarios (ii) and (iii) described in Subsection 7.1.1, we employed different convergence/forgetting factors ($\mu_{\text{N}}, \mu_{\text{L}}$, and β_{R}) to assess the theoretical predictions in diverse conditions. At each simulation run we evaluated 3000, 12000, and 1500 iterations for the NLMS, LMS, and RLS algorithms, respectively, where we considered the last 1000 iterations of each run to be part of the steady state. By taking an average of the $\text{MSE}_{\text{G}}[k]$ and $\text{MSD}_{\text{G}}[k]$ measurements at steady state for an ensemble of 1000 runs, we obtained the experimental results presented in Table 3. These results are com-

[3] Although these complexity results cannot be directly compared to the FLOPs estimation in Subsection 4.3, they corroborate it in terms of order of magnitude, illustrating that the GSP NLMS complexity represents a trade-off between the simpler GSP LMS and the more computationally complex GSP RLS.

Table 2: $\text{MSD}_{\text{G}}^{\star}$, iterations until convergence, and time for computing $\hat{\mathbf{x}}_{\text{o}}[k+1]$

Entry	Simulation Setup				Simulation Results		
	Alg.	Factor	$\mathbf{C}_w$	$\hat{\mathbf{x}}_{\text{o}}[k+1]$	$\text{MSD}_{\text{G}}^{\star}$	Converg.	Upd. Time
(L.I)	LMS	0.28	(1)	(10)	0.0313	1662 iter.	40 μs
(R.I)	RLS	0.93	(1)	(14)	0.0309	57 iter.	7041 μs
(N.I)	NLMS	0.07	(1)	(30)	0.0308	131 iter.	2474 μs
(N.I')	NLMS	0.07	(1)	(33)	0.0308	125 iter.	41 μs
(L.II)	LMS	0.28	(3)	(10)	0.3111	1416 iter.	41 μs
(R.II)	RLS	0.93	(3)	(14)	0.3041	57 iter.	7014 μs
(N.II)	NLMS	0.07	(3)	(30)	0.3055	114 iter.	2446 μs
(N.II')	NLMS	0.07	(3)	(33)	0.3058	111 iter.	41 μs
(L.III)	LMS	0.721	(3)	(10)	0.9987	488 iter.	40 μs
(R.III)	RLS	0.792	(3)	(14)	0.9729	18 iter.	7034 μs
(N.III)	NLMS	0.208	(3)	(30)	0.9765	34 iter.	2457 μs
(N.III')	NLMS	0.208	(3)	(33)	0.9779	33 iter.	41 μs

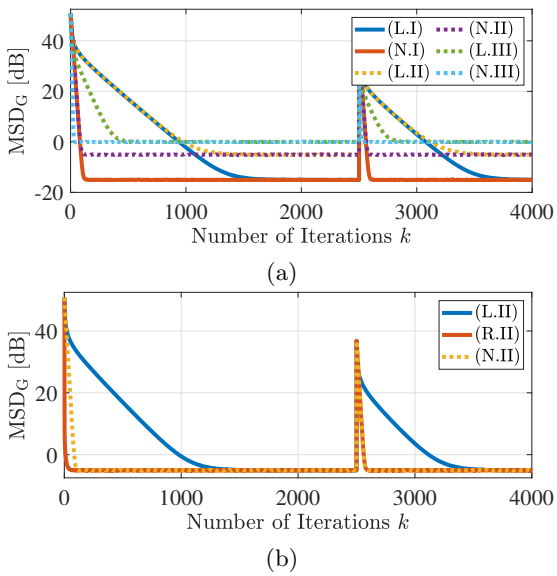


Figure 2: $\text{MSD}_{\text{G}}[k]$ behavior when using the simulation scenarios described in Table 2. (a) LMS and NLMS algorithms. (b) LMS, RLS, and NLMS algorithms.

Table 3: Theoretical and experimental $\text{MSE}_{\text{G}}^{*}$ and $\text{MSD}_{\text{G}}^{*}$, and their respective REs, for the NLMS, LMS, and RLS algorithms with different factors μ_{N}, μ_{L}, and β_{R}, respectively. Ensemble of 1000 runs and noise scenarios (ii) and (iii)

		$\mathbf{C}_w$ – (ii)						$\mathbf{C}_w$ – (iii)					
Algor.	Factor	$\text{MSE}_{\text{G}}^{*}$			$\text{MSD}_{\text{G}}^{*}$			$\text{MSE}_{\text{G}}^{*}$			$\text{MSD}_{\text{G}}^{*}$		
		Theory	Simul.	RE[%]	Theory	Simul.	RE[%]	Theory	Simul.	RE[%]	Theory	Simul.	RE[%]
NLMS	0.05	2.1513	2.1512	0.005	0.2217	0.2212	0.226	2.1464	2.1460	0.019	0.2202	0.2200	0.091
	0.10	2.2053	2.2052	0.005	0.4550	0.4549	0.022	2.2003	2.2004	-0.005	0.4520	0.4522	-0.044
	0.25	2.3857	2.3855	0.008	1.2350	1.2358	-0.065	2.3804	2.3801	0.013	1.2268	1.2264	0.033
	0.50	2.7667	2.7665	0.007	2.8817	2.8822	-0.017	2.7607	2.7605	0.007	2.8626	2.8623	0.010
LMS	0.20	2.2600	2.2599	0.004	0.2160	0.2159	0.046	2.2600	2.2598	0.009	0.2160	0.2161	-0.046
	0.50	2.5717	2.5711	0.023	0.6179	0.6179	0.000	2.5673	2.5674	-0.004	0.6171	0.6171	0.000
	1.00	3.4617	3.4638	-0.061	1.6803	1.6815	-0.071	3.4585	3.4595	-0.029	1.6793	1.6804	-0.065
RLS	0.95	2.1513	2.1516	-0.014	0.2217	0.2217	0.000	2.1513	2.1512	0.005	0.2217	0.2217	0.000
	0.90	2.2053	2.2050	0.014	0.4550	0.4551	-0.022	2.1999	2.1998	0.005	0.4501	0.4501	0.000
	0.75	2.3857	2.3857	0.000	1.2350	1.2362	-0.097	2.3794	2.3795	-0.004	1.2216	1.2221	-0.041

pared to the $\text{MSE}_{\text{G}}^{*}$ and $\text{MSD}_{\text{G}}^{*}$ theoretical predictions presented in: (34), (22), and (35) for the NLMS algorithm; (46) and (47) for the LMS algorithm; and (52) and (53) for the RLS algorithm. Additionally, in Table 3 we also include a relative error (RE) metric computed as

$$\text{Relative error (RE)} = \frac{\text{Theory value} - \text{Simul. result}}{\text{Simul. result}}\,. \tag{65}$$

From Table 3 we verify that the $\text{MSE}_{\text{G}}^{*}$ and $\text{MSD}_{\text{G}}^{*}$ predictions provided for the NLMS, LMS, and RLS algorithms are accurate across all different simulation scenarios. In particular, all results present an RE smaller than 0.25% with respect to their theoretical estimates. Although both the $\text{MSE}_{\text{G}}^{*}$ and $\text{MSD}_{\text{G}}^{*}$ predictions have been obtained in this work for all three adaptive GSP algorithms, it is worth mentioning that the $\text{MSD}_{\text{G}}^{*}$ for the LMS and RLS algorithms has been previously presented in [18]. However, the analysis for the LMS algorithm requires μ_{L} to be small and it presents an approximation that provides worse estimates of $\text{MSD}_{\text{G}}^{*}$ as μ_{L} increases. On the other hand, we observe from Table 3 that the accurate predictions for the LMS algorithm using (47) do not degrade with large μ_{L} factors.

7.1.4. Update Rate Steady-State Predictions

Finally, we performed a few numerical simulations to observe the behavior of the DS strategies presented in Section 6 and to verify the accuracy of update rate expressions (56) and (61). We will refer to the CW-EC strategy in Subsection 6.1 as DS scheme (I) and the ℓ_2N-EC strategy from Subsection 6.2 as DS scheme (II). By using the noise scenarios (ii) and (iii) described in Subsection 7.1.1, we employed different values of update factor κ for each DS scheme. For each of these environments we used either the NLMS, LMS, or RLS as the adaptive algorithm with factors $\mu_{\text{N}} = 0.1$, $\mu_{\text{L}} = 0.5$, and $\beta_{\text{R}} = 0.9$, in a total of 2500, 10000, and 2000 iterations for run, respectively. To compute P_{up} we only considered the last 1000 iterations of each run, and evaluated the average across an ensemble with 1000 runs. The scenario descriptions, theoretical predictions, and experimental results for the DS adaptive GSP strategies are in Table 4.

Table 4: Stationary P_{up} and RE for the adaptive GSP algorithms using DS strategies (I) and (II). Factors $\mu_{\text{N}} = 0.1$, $\mu_{\text{L}} = 0.5$, and $\beta_{\text{R}} = 0.9$ and total of 2500, 10000, and 2000 iterations for the NLMS, LMS, and RLS algorithms, respectively. Ensemble of 1000 runs and noise scenarios (ii) and (iii)

			$\mathbf{C}_w$ – (ii)						$\mathbf{C}_w$ – (iii)					
DS	κ	Theory	NLMS		LMS		RLS		NLMS		LMS		RLS	
		P_{up}[%]	P_{up}[%]	RE[%]	P_{up}[%]	RE[%]	P_{up}[%]	RE[%]	P_{up}[%]	RE[%]	P_{up}[%]	RE[%]	P_{up}[%]	RE[%]
(I)	3.00	43.319	43.314	0.012	43.300	0.044	43.325	-0.014	43.322	-0.007	43.343	-0.055	43.382	-0.145
(I)	3.50	9.310	9.322	-0.129	9.339	-0.311	9.319	-0.097	9.283	0.291	9.244	0.714	9.272	0.410
(I)	3.75	3.646	3.658	-0.328	3.640	0.165	3.635	0.303	3.647	-0.027	3.650	-0.110	3.678	-0.870
(II)	1.00	48.702	48.599	0.212	48.781	-0.162	48.744	-0.086	48.682	0.041	48.645	0.117	48.718	-0.033
(II)	1.10	15.276	15.301	-0.163	15.375	-0.644	15.214	0.408	15.317	-0.268	15.356	-0.521	15.296	-0.131
(II)	1.15	6.701	6.682	0.284	6.841	-2.046	6.724	-0.342	6.730	-0.431	6.791	-1.325	6.733	-0.475

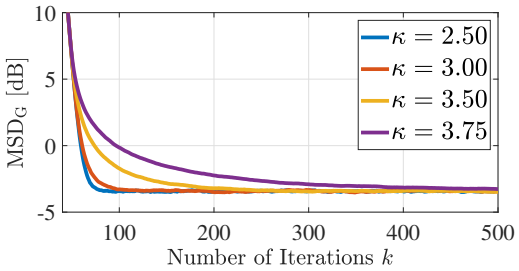


Figure 3: MSD_{G} of the NLMS algorithm when using different factors κ for the CW-EC DS strategy.

Based on Table 4 we conclude that the estimates (56) and (61) provide fair predictions about the update probability, where the largest RE evaluated is less than 2.5%. For both DS strategies we observe that a κ increase implies a reduction of the P_{up}, which is an interesting feature since it enhances the overall computational complexity reduction. However, the trade-off for increasing κ is a reduction in the algorithm convergence speed, as can be noted in the MSD_{G} behavior for the CW-EC strategy in Figure 3. Therefore, as Table 4 and Figure 3 indicate, one may reduce the overall complexity of the adaptive GSP algorithm by increasing the update factor of the DS strategy, at the cost of slowing down the convergence.

7.2. U.S. Climate Normals Dataset

For assessing the adaptive algorithms' performance with real data, we considered the dataset in [37], from which we extracted meteorological information of weather stations distributed on the United States' territory, as done in [18]. We collected the temperature hourly records from the first 95 hours of January 2010 and, after discarding the stations with missing inputs, we obtained a total of $N = 205$ stations in the US mainland. By employing all of these data for representing the graph signal in the current scenario, the graph structure is obtained through a procedure similar to the one described in Subsection 7.1, but using a 7-nearest-neighbor approach, as in [18]. According to this procedure, we found the graph structure illustrated in Figure 4, where it is shown

the temperature measurements at 01:00 from January 1$^{\text{st}}$ 2010.

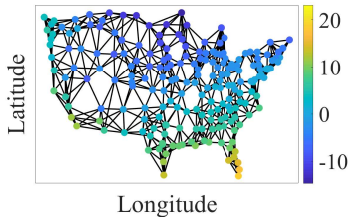


Figure 4: Graph structure inferred for the dataset [37], with the temperature (degree Celsius) hourly record from January 1$^{\text{st}}$ 2010 at 01:00 as the graph signal.

Assuming an acceptable reconstruction error of 5%, we found that $|\mathcal{F}| = 125$ resulted in a suitable approximation. The sampling set $\mathcal{S}$ was obtained as in Subsection 7.1, where we have used $|\mathcal{S}| = 130$. In order to first compare the GSP LMS, RLS, and NLMS algorithms in a static real data scenario, we have employed the temperature values at 01:00 January 1$^{\text{st}}$ 2010 with an incident noise following the noise scenario (ii) from Subsection 7.1.1. By using the factors $\mu_{\text{L}} = 1.5$ and $\beta_{\text{R}} = 0.5$ suggested in [18], and $\mu_{\text{N}} = 0.5$, the behavior of $\text{MSD}_{\text{G}}[k]$ is displayed for the adaptive algorithms in Figure 5. The undershoot behavior of the GSP LMS method in Figure 5 might be justified by a large value of μ_{L} taking the algorithm closer to its instability region; yet, the GSP LMS still requires a much longer time for settling in comparison to the proposed GSP NLMS algorithm, which is almost as fast as the GSP RLS procedure.

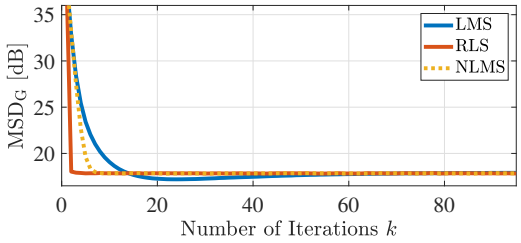


Figure 5: Adaptive algorithms performance when estimating a static real data from [37].

Additionally, for assessing the performance of these adaptive strategies in tracking real data, we have employed the GSP LMS, RLS, and NLMS algorithms for estimating unobserved weather stations using the dataset from 01:00 January 1$^{\text{st}}$ 2010 until 23:00 January 4$^{\text{th}}$ 2010. The estimation of a randomly picked unobserved node, which have not been sampled according to the index set $\mathcal{S}$, is presented in Figure 6. Based on this result, we observe that the behavior of the GSP NLMS resembles the GSP RLS algorithm (apart from a slight difference in the first iterations), while presenting a better performance in estimating most unobserved nodes in comparison to the GSP LMS method.

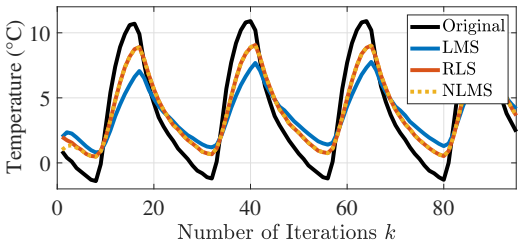


Figure 6: Original temperature signal and adaptive estimates across time at a randomly picked unobserved weather station from [37].

8. Conclusion

In this paper we proposed an adaptive strategy for online reconstruction of bandlimited graph signals based on the NLMS algorithm and two time-domain data-selective methods for reducing the overall computational complexity of general adaptive GSP algorithms. The proposed GSP NLMS algorithm presents much faster convergence than the LMS algorithm and is introduced with a comprehensive analysis that predicts some steady-state figures of merit. The same analysis methodology allows the estimation of steady-state metrics for the LMS and RLS algorithms, complementing prior results. These steady-state values are used for predicting the update probability of the proposed data-selective strategies when they are implemented with all three GSP adaptive algorithms. Simulation results indicate that the metric estimations provided in the analyses and the update probability predicted for data-selective techniques are accurate.

Appendix A. RLS Estimation Alternative Update Equations

In this part we represent $\boldsymbol{\Psi}[k]$ and $\boldsymbol{\psi}[k]$ from (12) and (13) with subscript indices k as $\boldsymbol{\Psi}_k$ and $\boldsymbol{\psi}_k$, respectively, for simplicity sake. Additionally, due to its initialization as a diagonal matrix, from (12) we find that $\boldsymbol{\Psi}_k$ is symmetric.

Based on (12), by taking $\mathbf{G}_k = \mathbf{D}_{\mathcal{S}[k]}\mathbf{U}_{\mathcal{F}}\boldsymbol{\Psi}_{k-1}^{-1}\mathbf{U}_{\mathcal{F}}^{\mathrm{T}}\mathbf{D}_{\mathcal{S}[k]}$, the matrix inversion lemma states that $\boldsymbol{\Psi}_k^{-1}$ can be written as

$$\boldsymbol{\Psi}_k^{-1} = \left[\mathbf{I} - \boldsymbol{\Psi}_{k-1}^{-1}\mathbf{U}_{\mathcal{F}}^{\mathrm{T}}\mathbf{D}_{\mathcal{S}[k]}(\beta_{\mathrm{R}}\mathbf{C}_w + \mathbf{G}_k)^{-1}\mathbf{D}_{\mathcal{S}[k]}\mathbf{U}_{\mathcal{F}}\right]\beta_{\mathrm{R}}^{-1}\boldsymbol{\Psi}_{k-1}^{-1}\,. \tag{A.1}$$

From (2) and $\hat{\mathbf{x}}_{\mathrm{o}}[k+1]$ in (13), it is clear that $\boldsymbol{\Psi}_k^{-1}\boldsymbol{\psi}_k$ is equal to $\hat{\mathbf{s}}_{\mathcal{F}}[k+1]$. Thus, by multiplying $\boldsymbol{\Psi}_k^{-1}$ in (A.1) and $\boldsymbol{\psi}_k$ from (12) we find $\hat{\mathbf{s}}_{\mathcal{F}}[k+1]$. Based on (1), $\hat{\mathbf{x}}_{\mathrm{o}}[k+1]$ is

$$\hat{\mathbf{x}}_{\mathrm{o}}[k+1] = \hat{\mathbf{x}}_{\mathrm{o}}[k] + \mathbf{U}_{\mathcal{F}}\boldsymbol{\Psi}_{k-1}^{-1}\mathbf{U}_{\mathcal{F}}^{\mathrm{T}}\mathbf{D}_{\mathcal{S}[k]}(\beta_{\mathrm{R}}\mathbf{C}_w + \mathbf{G}_k)^{-1}\mathbf{e}[k]\,. \tag{A.2}$$

When right-multiplying (A.1) by $\mathbf{U}_{\mathcal{F}}^{\mathrm{T}}\mathbf{D}_{\mathcal{S}[k]}\mathbf{C}_w^{-1}$ it follows that

$$\boldsymbol{\Psi}_k^{-1}\mathbf{U}_{\mathcal{F}}^{\mathrm{T}}\mathbf{D}_{\mathcal{S}[k]}\mathbf{C}_w^{-1} = \boldsymbol{\Psi}_{k-1}^{-1}\mathbf{U}_{\mathcal{F}}^{\mathrm{T}}\mathbf{D}_{\mathcal{S}[k]}(\beta_{\mathrm{R}}\mathbf{C}_w + \mathbf{G}_k)^{-1}\,,$$

which allows us to rewrite expression (A.2) as

$$\hat{\mathbf{x}}_{\mathrm{o}}[k+1] = \hat{\mathbf{x}}_{\mathrm{o}}[k] + \mathbf{U}_{\mathcal{F}}\boldsymbol{\Psi}_k^{-1}\mathbf{U}_{\mathcal{F}}^{\mathrm{T}}\mathbf{D}_{\mathcal{S}[k]}\mathbf{C}_w^{-1}\mathbf{e}[k]\,. \tag{A.3}$$

Appendix B. GSP NLMS Stability and Convergence Derivations

As we shall focus on steady-state values, let us assume a time-invariant reference GS such that $\mathbf{s}_{\mathcal{F}}[k] = \mathbf{s}_{\mathcal{F}}$ and a time-invariant noise covariance matrix such that $\mathbf{C}_w[k] = \mathbf{C}_w$.

Based on the definition of $\Delta\hat{\boldsymbol{s}}_{\mathcal{F}}[k]$ in (18) and the frequency-domain version of equation (30), it follows that

$$\Delta\hat{\boldsymbol{s}}_{\mathcal{F}}[k+1] = (1-\mu_{\mathrm{N}})\Delta\hat{\boldsymbol{s}}_{\mathcal{F}}[k] + \mu_{\mathrm{N}}\mathbf{M}[k]\mathbf{U}_{\mathcal{F}}^{\mathrm{T}}\mathbf{D}_{\mathcal{S}[k]}\boldsymbol{w}[k]\,, \tag{B.1}$$

thus yielding $\mathbb{E}\{\Delta\hat{\boldsymbol{s}}_{\mathcal{F}}[k+1]\} = (1-\mu_{\mathrm{N}})\,\mathbb{E}\{\Delta\hat{\boldsymbol{s}}_{\mathcal{F}}[k]\}$. This allows one to write

$$\mathbb{E}\{\Delta\hat{\boldsymbol{s}}_{\mathcal{F}}[k]\} = (1-\mu_{\mathrm{N}})^k\,\mathbb{E}\{\Delta\hat{\boldsymbol{s}}_{\mathcal{F}}[0]\}\,. \tag{B.2}$$

Considering that $\mathbb{E}\{\Delta\hat{\boldsymbol{s}}_{\mathcal{F}}[0]\}$ can be any vector, to guarantee that $\mathbb{E}\{\Delta\hat{\boldsymbol{s}}_{\mathcal{F}}[k]\}$ in (B.2) converges to a null vector as k increases, we must choose a parameter μ_{N} such that $|1-\mu_{\mathrm{N}}| < 1$. As a result, the interval that guarantees convergence to an unbiased solution is

$$0 < \mu_{\mathrm{N}} < 2\,. \tag{B.3}$$

In addition, by defining $\mathbf{S}_{\mathrm{N}}[k] = \mathbb{E}[\Delta\hat{\boldsymbol{s}}_{\mathcal{F}}[k]\Delta\hat{\boldsymbol{s}}_{\mathcal{F}}^{\mathrm{T}}[k]]$ and assuming that $\mathcal{S}[k]$ converges to a static sampling set $\mathcal{S}$, one has from (B.1) that

$$\mathbf{S}_{\mathrm{N}}[k+1] = (1-\mu_{\mathrm{N}})^2\mathbf{S}_{\mathrm{N}}[k] + \mu_{\mathrm{N}}^2\mathbf{M}\mathbf{U}_{\mathcal{F}}^{\mathrm{T}}\mathbf{D}_{\mathcal{S}}\mathbf{C}_w\mathbf{D}_{\mathcal{S}}\mathbf{U}_{\mathcal{F}}\mathbf{M}, \tag{B.4}$$

which is a difference equation that converges to a solution as long as $|1-\mu_{\mathrm{N}}| < 1$, i.e., the condition in (B.3) holds true. In this case, stability is guaranteed and $\mathbb{E}[\Delta\hat{\boldsymbol{s}}_{\mathcal{F}}[k]\Delta\hat{\boldsymbol{s}}_{\mathcal{F}}^{\mathrm{T}}[k]]$ approaches $\mathbf{S}_{\mathrm{N}}^{*} \in \mathbb{R}^{|\mathcal{F}|\times|\mathcal{F}|}$ as $k \to \infty$, given as

$$\mathbf{S}_{\mathrm{N}}^{*} = \mathbf{S}_{\mathrm{N}}[\infty] = \frac{\mu_{\mathrm{N}}}{2-\mu_{\mathrm{N}}}\mathbf{M}\mathbf{U}_{\mathcal{F}}^{\mathrm{T}}\mathbf{D}_{\mathcal{S}}\mathbf{C}_w\mathbf{D}_{\mathcal{S}}\mathbf{U}_{\mathcal{F}}\mathbf{M}\,. \tag{B.5}$$

References

[1] Y. Chen, S. Kar, J. M. F. Moura, The internet of things: Secure distributed inference, IEEE Signal Processing Magazine 35 (5) (2018) 64–75. doi:10.1109/MSP.2018.2842097.

[2] A. Sandryhaila, J. M. F. Moura, Big data analysis with signal processing on graphs: Representation and processing of massive data sets with irregular structure, IEEE Signal Processing Magazine 31 (5) (2014) 80–90. doi:10.1109/MSP.2014.2329213.

[3] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, P. Vandergheynst, The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains, IEEE Signal Processing Magazine 30 (3) (2013) 83–98. doi:10.1109/MSP.2012.2235192.

[4] A. Sandryhaila, J. M. F. Moura, Discrete signal processing on graphs, IEEE Transactions on Signal Processing 61 (7) (2013) 1644–1656. doi:10.1109/TSP.2013.2238935.

[5] A. Sandryhaila, J. M. F. Moura, Discrete signal processing on graphs: Frequency analysis, IEEE Transactions on Signal Processing 62 (12) (2014) 3042–3054. doi:10.1109/TSP.2014.2321121.

[6] A. Ortega, P. Frossard, J. Kovačević, J. M. F. Moura, P. Vandergheynst, Graph signal processing: Overview, challenges, and applications, Proceedings of the IEEE 106 (5) (2018) 808–828. doi:10.1109/JPROC.2018.2820126.

[7] S. K. Narang, A. Gadde, A. Ortega, Signal processing techniques for interpolation in graph structured data, in: 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, 2013, pp. 5445–5449. doi:10.1109/ICASSP.2013.6638704.

[8] A. Anis, A. Gadde, A. Ortega, Towards a sampling theorem for signals on arbitrary graphs, in: 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2014, pp. 3864–3868. doi:10.1109/ICASSP.2014.6854325.

[9] S. Chen, A. Sandryhaila, J. Kovačević, Sampling theory for graph signals, in: 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2015, pp. 3392–3396. doi:10.1109/ICASSP.2015.7178600.

[10] S. Chen, R. Varma, A. Sandryhaila, J. Kovačević, Discrete signal processing on graphs: Sampling theory, IEEE Transactions on Signal Processing 63 (24) (2015) 6510–6523. doi:10.1109/TSP.2015.2469645.

[11] L. F. O. Chamon, A. Ribeiro, Greedy sampling of graph signals, IEEE Transactions on Signal Processing 66 (1) (2018) 34–47. doi:10.1109/TSP.2017.2755586.

[12] S. Chen, A. Sandryhaila, J. M. F. Moura, J. Kovačević, Signal recovery on graphs: Variation minimization, IEEE Transactions on Signal Processing 63 (17) (2015) 4609–4624. doi:10.1109/TSP.2015.2441042.

[13] S. Chen, R. Varma, A. Singh, J. Kovačević, Signal recovery on graphs: Fundamental limits of sampling strategies, IEEE Transactions on Signal and Information Processing over Networks 2 (4) (2016) 539–554. doi:10.1109/TSIPN.2016.2614903.

[14] P. D. Lorenzo, S. Barbarossa, P. Banelli, S. Sardellitti, Adaptive least mean squares estimation of graph signals, IEEE Transactions on Signal and Information Processing over Networks 2 (4) (2016) 555–568. doi:10.1109/TSIPN.2016.2613687.

[15] P. D. Lorenzo, P. Banelli, S. Barbarossa, S. Sardellitti, Distributed adaptive learning of graph signals, IEEE Transactions on Signal Processing 65 (16) (2017) 4193–4208. doi:10.1109/TSP.2017.2708035.

[16] P. D. Lorenzo, P. Banelli, S. Barbarossa, Optimal sampling strategies for adaptive learning of graph signals, in: 2017 25th European Signal Processing Conference (EUSIPCO), 2017, pp. 1684–1688. doi:10.23919/EUSIPCO.2017.8081496.

[17] P. D. Lorenzo, E. Isufi, P. Banelli, S. Barbarossa, G. Leus, Distributed recursive least squares strategies for adaptive reconstruction of graph signals, in: 2017 25th European Signal Processing Conference (EUSIPCO), 2017, pp. 2289–2293. doi:10.23919/EUSIPCO.2017.8081618.

[18] P. D. Lorenzo, P. Banelli, E. Isufi, S. Barbarossa, G. Leus, Adaptive graph signal processing: Algorithms and optimal sampling strategies, IEEE Transactions on Signal Processing 66 (13) (2018) 3584–3598. doi:10.1109/TSP.2018.2835384.

[19] P. Lorenzo, S. Barbarossa, P. Banelli, Chapter 9 - sampling and recovery of graph signals, in: P. M. Djurić, C. Richard (Eds.), Cooperative and Graph Signal Processing, Academic Press, 2018, pp. 261 – 282. doi:https://doi.org/10.1016/B978-0-12-813677-5.00009-2.
URL http://www.sciencedirect.com/science/article/pii/B9780128136775000092

[20] P. D. Lorenzo, E. Ceci, Online recovery of time-varying signals defined over dynamic graphs, in: 2018 26th European Signal Processing Conference (EUSIPCO), 2018, pp. 131–135. doi:10.23919/EUSIPCO.2018.8553473.

[21] M. J. M. Spelta, W. A. Martins, Online temperature estimation using graph signals, in: XXXVI Simpósio Brasileiro de Telecomunicações e Processamento de Sinais - SBrT2018, 2018, pp. 154–158. doi:10.23919/EUSIPCO.2017.8081496.

[22] Instituto Nacional de Meteorologia (INMET) , Normais Climatológicas do Brasil, http://www.inmet.gov.br/portal/index.php?r=clima/normaisClimatologicas, [Online; accessed 02-April-2018].

[23] M. J. M. Spelta, Brazilian weather stations, http://github.com/mspelta/brazilian-weather-stations (2018).

[24] B. Widrow, M. E. Hoff, Adaptive switching circuits, WESCOM Conv. Rec. 4 (1960) 96–140. doi:10.1109/PROC.1976.10286.

[25] B. Widrow, J. M. McCool, M. G. Larimore, C. R. Johnson, Stationary and nonstationary learning characteristics of the LMS adaptive filter , Proceedings of the IEEE 64 (8) (1976) 1151–1162. doi:10.1109/PROC.1976.10286.

[26] A. H. Sayed, Adaptive filters, John Wiley & Sons, 2011.

[27] P. S. R. Diniz, Adaptive Filtering: Algorithms and Practical Implementation, Springer, 2013.

[28] E. Eleftheriou, D. Falconer, Tracking properties and steady-state performance of RLS adaptive filter algorithms, IEEE Transactions on Acoustics, Speech, and Signal Processing 34 (5) (1986) 1097–1110. doi:10.1109/TASSP.1986.1164950.

[29] N. Cressie, The origins of kriging, Mathematical Geology 22 (3) (1990) 239–252. doi:10.1007/BF00889887.
URL https://doi.org/10.1007/BF00889887

[30] J. Nagumo, A. Noda, A learning method for system identification, IEEE Transactions on Automatic Control 12 (3) (1967) 282–287. doi:10.1109/TAC.1967.1098599.

[31] D. T. M. Slock, On the convergence behavior of the LMS and the normalized LMS algorithms , IEEE Transactions on Signal Processing 41 (9) (1993) 2811–2825. doi:10.1109/78.236504.

[32] S. Werner, P. S. R. Diniz, Set-membership affine projection algorithm, IEEE Signal Processing Letters 8 (8) (2001) 231–235.

[33] S. Nagaraj, S. Gollamudi, S. Kapoor, Y.-F. Huang, BEACON: an adaptive set-membership filtering technique with sparse updates, IEEE Transactions on Signal Processing 47 (11) (1999) 2928–2941. doi:10.1109/78.796429.

[34] P. S. R. Diniz, On data-selective adaptive filtering, IEEE Transactions on Signal Processing 66 (16) (2018) 4239–4252. doi:10.1109/TSP.2018.2847657.

[35] D. Berberidis, V. Kekatos, G. B. Giannakis, Online censoring for large-scale regressions with application to streaming big data, IEEE Transactions on Signal Processing 64 (15) (2016) 3854–3867. doi:10.1109/TSP.2016.2546225.

[36] D. K. Berberidis, V. Kekatos, G. Wang, G. B. Giannakis, Adaptive censoring for large-scale regressions, in: 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2015, pp. 5475–5479. doi:10.1109/ICASSP.2015.7179018.

[37] National Oceanic and Atmospheric Administration (NOAA) , 1981-2010 U.S. Climate Normals, https://www.ncdc.noaa.gov/data-access/land-based-station-data/landbased-datasets/climate-normals/1981-2010-normals-data, [Online; accessed 23-June-2019].

[38] K. B. Petersen, M. S. Pedersen, The Matrix Cookbook (2012 (accessed November 9, 2018)).
URL https://www.math.uwaterloo.ca/~hwolkowi/matrixcookbook.pdf

[39] L. N. Trefethen, D. B. III, Numerical Linear Algebra, SIAM: Society for Industrial and Applied Mathematics, 1997.

[40] T. Damm, Direct methods and adi-preconditioned Krylov subspace methods for generalized lyapunov equations, Numerical Lin. Alg. with Applic. 15 (2008) 853–871.

[41] A. J. Laub, Matrix Analysis For Scientists And Engineers, SIAM: Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2004.

[42] L. C. Andrews, Special functions of mathematics for engineers, Oxford Univ. Press, 1998.

[43] R. V. Hogg, A. T. Craig, Introduction to Mathematical Statistics, Macmillan Publishing Co, Inc., 1978.

[44] M. J. M. Spelta, NLMS and DS Strategies for GSP , `http://github.com/mspelta/NLMS-and-DS-strategies-for-GSP` (2019).