Stationary time-vertex signal processing

Andreas Loukas* and Nathanaël Perraudin*
École Polytechnique Fédérale Lausanne, Switzerland

***Abstract*—The goal of this paper is to improve learning for multivariate processes whose structure is dependent on some *known* graph topology; especially when the number of available samples is much smaller than the number of variables. Typically, the graph information is incorporated into the learning process via a smoothness assumption postulating that the values supported on well-connected vertices exhibit small variations. We argue that smoothness is not enough. To capture the behavior of complex interconnected systems, such as transportation and biological networks, it is important to train expressive models, being able to reproduce a wide range of graph and temporal behaviors.**

Motivated by this need, this paper puts forth a novel definition of time-vertex wide-sense stationarity, or *joint stationarity* for short. We believe that the proposed definition is natural, at it intimately relates to existing definitions of stationarity in the time and vertex domains. We use joint stationarity to regularize learning and to reduce computational complexity in both estimation and recovery tasks. In particular, we show that for any jointly stationary process: (a) one can learn the covariance structure from $O(1)$ samples, and (b) can solve MMSE recovery problems, such as interpolation, denoising, forecasting, in complexity that is linear to the edges and timesteps. Experiments with three datasets suggest that joint stationarity can yield significant accuracy improvements in the reconstruction effort of under-sampled problems, even when the graph is only approximately known or the process is only close to stationary.

***Index Terms*—stationarity, multivariate time-vertex processes, harmonic analysis, graph signal processing, PSD estimation.**

I. Introduction

One of the main challenges when working with multivariate processes is to learn their statistical structure from few realizations of the process (samples). Concretely, suppose that we wish to estimate the first two moments of a process $\boldsymbol{X} \in \mathbb{R}^{N \times T}$, where N is the number of variables and T the number of timesteps. If no restricting assumptions are made (other than the first four moments are finite) then the number of samples needed to attain statistical significance is up to a logarithmic factor proportional to $O(NT)$, the degrees of freedom [1]. Assuming that the process is time wide-sense stationarity (TWSS) is very helpful as it reduces the degrees of freedom of the system –and thus the sample requirement– by a factor of T. Even a linear dependency on N however is often problematic in practice, when the number of variables is large and the ability to obtain multiple samples limited.

The goal of this paper is to improve learning for the specific cases when the multivariate process is supported on the vertex set and is statistically dependent on the edge set of some known graph topology. Whether examining epidemic spreading [2], how traffic evolves in the roads of a city [3], or neuronal activation patterns present in the brain [4], many of the high-dimensional processes one encounters are inherently constrained by some underlying network. This realization has been the driving force behind recent efforts to re-invent classical models by taking into account the graph structure, with advances in many problems, such as denoising [5] and semi-supervised learning [6], [7].

Yet, state-of-the-art models for processes (evolving) on graphs often fail to produce useful results when applied to real datasets. One of the main reasons for this shortcoming is that they model only a limited set of (smooth) spatio-temporal behaviors. The well-used graph Tikhonov and total variation priors for instance assume that the signal varies slowly or in a piece-wise constant manner over edges, without specifying any precise relations [8], [9]. Similarly, assuming that the graph Laplacian encodes the conditional correlations of variables, as is done with Gaussian Markov Random Fields [10], works well when the graph is not available, but becomes a rigid model when the graph is given [11]. To capture the behavior of complex networked systems, such as transportation and biological networks, it is important to train expressive models, being able to reproduce a wide range of graph and temporal behaviors.

Motivated by this need, this paper considers the statistical modeling of processes evolving on graphs. Our results are inspired by the recent introduction of a joint temporal and graph Fourier transform (JFT), a generalization of GFT appropriate for time-varying graph signals [12], [13], and the recent generalization of stationarity for graphs [14], [15], [16]. Our main contribution is a novel definition of time-vertex wide-sense stationarity, or *joint stationarity* for short. We believe that the proposed definition is natural, at it elegantly relates to existing definitions of stationarity in the time and vertex domains. We show that joint stationarity carries along important properties classically associated with stationarity: joint wide-sense stationary (JWSS) processes can be generated by filtering noise, and a joint Fourier transform diagonalizes their covariance. Furthermore, our definition is intimately linked with the familiar definitions for stationarity of multivariate time and graph processes.

We use the hypothesis of joint stationarity to regularize learning and to reduce computational complexity in both estimation and recovery tasks. Within our framework, one learns the covariance structure of a JWSS process from $O(1)$ samples and recovery (such as interpolation, denoising, forecasting) comes with a computational complexity that is close to linear on the number of edges and timesteps. In addition, we find that assuming joint stationarity aids in recovery even when only an approximation of the graph is known, or the process is only approximately jointly stationary. We therefore propose our model as good candidate for graph-related problems featuring

*A. Loukas and N. Perraudin contributed equally to this work.

a large number of variables with only a limited number of learning samples.

To test the utility of joint stationarity, we apply our methods on three diverse datasets: (a) a meteorological dataset depicting the hourly temperature of 32 weather stations over one month in Molene, France, (b) a traffic dataset depicting high resolution daily vehicle flow of 4 weekdays in the highways of Sacramento, and (c) simulated SIRS-type epidemics over Europe. Our experiments confirm that in the few samples regime, assuming joint stationarity yields an improvement in recovery performance as compared to time- or vertex-based methods, even when the graph is only approximately known and the data violate the strict conditions of our definition.

A. Related work

There exists an extensive literature on multivariate stationary processes, developing the original work of Wiener et al. [17], [18]. The reader may find interesting Bloomberg's book [19] focusing on the spectral relations. We focus on two main approaches that relate to our work, graphical models and signal processing on graphs.

Graphical models. In the context of graphical models, multivariate stationarity has been used jointly with a graph in the work of [20], [21]. Though relevant, we note that there is a key difference of these models with our approach: we assume that the graph is given, whereas in graphical models the graph structure (or more precisely the precision matrix) is the learned from the data. Knowing the graph allows us to search for more involved relations between the variables. As such, we are not restricted to the case that the conditional dependencies are given by the graph (and therefore that they are sparse), but allow non-adjacent variables to be conditionally dependent, modeling a wider set of behaviors. We also note that our approach is eventually more scalable. We refer to [11] for elements of connections between graphical models and graph signal processing.

Graph signal processing. The idea of studying the stationarity of a random vector with respect to a graph was first introduced by Girault et al. [15], [22] and then by Perraudin et al. [14]. While these contributions have different starting points, they both propose the same definition, i.e., the one we generalize in this contribution. Other recent contributions relating to stationarity on graphs are [16], [23]. Despite the relevance of these works, it is important to stress that this paper is the first to consider processes that are varying both in the vertex and time domains. In addition, the analysis presented here (particularly that of Section IV-C) is novel and can also be employed for the previously studied case of graph stationary processes [15], [22], [16]. We also note that the task of prediction using the two first statistical moments for time-evolving signal on graphs was also considered in [24], [25]. Nonetheless, there are a number of differences with these works, with the most important being that we define joint stationarity, and that we are not restricted to the causal case.

It should be noted that some results of this paper appeared in a conference paper [26].

II. Preliminaries

We consider signals supported on the vertices $\mathcal{V} = \{v_1, v_2, \dots, v_N\}$ of a weighted undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \boldsymbol{W}_G)$, with $\mathcal{E}$ the set of edges of cardinality $E = |\mathcal{E}|$ and $\boldsymbol{W}_G$ the weighted adjacency matrix.

Suppose that signal $\boldsymbol{x}_t$ is sampled at T successive regular intervals of unit length. The time-vertex signal $\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_T] \in \mathbb{R}^{N \times T}$ is then the matrix having graph signal $\boldsymbol{x}_t$ as its t-th column. Throughout this paper, we denote as $\boldsymbol{x} = \text{vec}(\boldsymbol{X})$ (without subscript) the vectorized representation of the matrix $\boldsymbol{X}$.

Harmonic time-vertex analysis. The frequency representation of a time-vertex signal $\boldsymbol{X}$ is given by the Joint Fourier Transform [13] (or JFT for short)

$$\hat{\boldsymbol{X}} = \text{JFT}\{\boldsymbol{X}\} \triangleq \text{GFT}\{\text{DFT}\{\boldsymbol{X}\}\} = \boldsymbol{U}_G^* \boldsymbol{X} (\boldsymbol{U}_T^*)^\intercal, \qquad (1)$$

with $\boldsymbol{U}_G$ and $\boldsymbol{U}_T$ being, respectively, the unitary Graph Fourier Transform (GFT) and Discrete Fourier Transform (DFT) matrices. The notation $\boldsymbol{U}_G^*$ denotes the transposed complex conjugate of $\boldsymbol{U}_G$, $\boldsymbol{U}_T^\intercal$ the transpose of $\boldsymbol{U}_T$, and $(\boldsymbol{U}_T^*)^\intercal$ the complex conjugate of $\boldsymbol{U}_T$. In vector form, we have that $\hat{\boldsymbol{x}} = \text{JFT}\{\boldsymbol{x}\} \triangleq \boldsymbol{U}_J^* \boldsymbol{x}$, where $\boldsymbol{U}_J = \boldsymbol{U}_T \otimes \boldsymbol{U}_G$ and operator $(\otimes)$ denotes the Kronecker product. As is often the case, we choose $\boldsymbol{U}_G$ to be the eigenvector matrix of the combinatorial[1] graph Laplacian matrix $\boldsymbol{L}_G = \text{diag}(\boldsymbol{W}_G \mathbf{1}_N) - \boldsymbol{W}_G$, where $\mathbf{1}_N$ is the all-ones vector of size N, and $\text{diag}(\boldsymbol{W}_G \mathbf{1}_N)$ is the diagonal degree matrix. On the other hand, matrix $\boldsymbol{U}_T$ is the eigenvector matrix of the Laplacian matrix $\boldsymbol{L}_T$ of a cyclic graph and

$$\boldsymbol{U}_T^*[t, \tau] = \frac{e^{-j\omega_\tau t}}{\sqrt{T}}, \quad \text{with} \quad \omega_\tau = \frac{2\pi(\tau - 1)}{T}, \qquad (2)$$

for $t, \tau = 1, 2, \dots, T$. Note that $\hat{\boldsymbol{X}}[n, \tau]$ is the Fourier coefficient associated with the joint frequency $[\lambda_n, \omega_\tau]$, where λ_n denotes the n-th graph eigenvalue and ω_τ the τ-th angular frequency. For an in-depth discussion of JFT and its properties, we refer the reader to [13], [27].

Joint time-vertex filtering. Filtering a time-vertex signal $\boldsymbol{x}$ with a *joint filter* $h(\boldsymbol{L}_G, \boldsymbol{L}_T)$ corresponds to element-wise multiplication in the joint frequency domain $[\lambda, \omega]$ by a function $h : [0 \lambda_{\max}] \times [-1,\ 1] \mapsto \mathbb{R}$ [28], [29], [13], [27]. When a joint filter $h(\boldsymbol{L}_G, \boldsymbol{L}_T)$ is applied to $\boldsymbol{x}$, the output is

$$h(\boldsymbol{L}_G, \boldsymbol{L}_T)\, \boldsymbol{x} = \boldsymbol{U}_J\, h(\boldsymbol{\Lambda}_G, \boldsymbol{\Omega})\, \boldsymbol{U}_J^* \boldsymbol{x}, \qquad (3)$$

where $\boldsymbol{\Lambda}_G \in \mathbb{R}^{N \times N}$ and $\boldsymbol{\Omega} \in \mathbb{R}^{T \times T}$ are diagonal matrices with $\boldsymbol{\Lambda}_G[n, n] = \lambda_n$ and $\boldsymbol{\Omega}[\tau, \tau] = \omega_\tau$, whereas $h(\boldsymbol{\Lambda}_G, \boldsymbol{\Omega})$ is a diagonal $NT \times NT$ matrix defined as

$$h(\boldsymbol{\Lambda}_G, \boldsymbol{\Omega}) = \text{diag}\left(\begin{bmatrix} h(\lambda_1, \omega_1) & \cdots & h(\lambda_1, \omega_T) \\ \vdots & \ddots & \vdots \\ h(\lambda_N, \omega_1) & \cdots & h(\lambda_N, \omega_T) \end{bmatrix} \right)$$

[1] Though we use the combinatorial Laplacian in our presentation, our results are applicable to any positive semi-definite matrix definition of a graph Laplacian or to the recently introduced shift operator [9].

and the $\text{diag}(\boldsymbol{A})$ operator creates a matrix with diagonal elements the vectorized form of $\boldsymbol{A}$. For convenience, we will often abuse notation and write $h(\theta_{n,\tau})$ to refer to $h(\lambda_n, \omega_\tau)$. Furthermore, we say that a joint filter is *separable*, if its joint frequency response h can be written as the product of a frequency response h_1 defined solely in the vertex domain and one h_2 in the time domain, i.e., $h(\theta) = h_1(\omega) \cdot h_2(\lambda)$.

III. Joint Time-Vertex Stationarity

Let $\boldsymbol{X} \in \mathbb{R}^{N \times T}$ be a discrete multivariate stochastic process (with finite number of time-steps T) that is indexed by the vertex v_i of graph $\mathcal{G}$ and time t. We refer to such processes as time-vertex processes, or *joint processes* for short.

Our objective is to provide a definition of stationarity that captures statistical invariance of the first two moments of a joint process $\boldsymbol{x} = \text{vec}(\boldsymbol{X}) \sim \mathcal{D}(\bar{\boldsymbol{x}}, \boldsymbol{\Sigma})$, i.e., the mean $\bar{\boldsymbol{x}} = \mathbf{E}[\boldsymbol{x}]$ and the covariance $\boldsymbol{\Sigma}$ under distribution $\mathcal{D}$. Crucially, the definition should do so in a manner that is faithful to the graph and temporal structure.

Typically, wide-sense stationarity is thought of as an invariance of the two first moments of a process with respect to translation. For the first moment things are straightforward: stationarity implies a constant mean $\mathbf{E}[\boldsymbol{x}] = c\mathbf{1}$, independently of the domain of interest. The second moment however is more complicated as it depends on the exact form translation takes in the particular domain. Unfortunately, for graphs translation is a non-trivial operation and three alternative translation operators exist: the generalized translation [30], the graph shift [9], and the isometric graph translation [22]. Due to this challenge, there are currently three alternative definitions of stationarity appropriate for graphs [14], [15], [16], one for each definition of translation.

The ambiguity associated with translation on graphs urges us to seek an alternative starting point for our definition. Fortunately, there exists an interpretation which holds promise: *up to its constant mean, a wide-sense stationary process corresponds to a white process filtered linearly on the underlying space*. This “filtering interpretation” of stationarity is well known classically[2] and is equivalent to asserting that the second moment can be expressed as $\boldsymbol{\Sigma} = h(\boldsymbol{L}_T)$, where $h(\boldsymbol{L}_T)$ is a linear filter. Thankfully, not only filtering is elegantly and uniquely defined for graphs [30], but also stating that a process is graph wide-sense stationary iff $\mathbf{E}[\boldsymbol{x}] = c\mathbf{1}_N$ and $\boldsymbol{\Sigma} = h(\boldsymbol{L}_G)$ is a graph filter, is generally consistent[3] with current definitions [14], [15], [16].

This motivates us to also express the definition of stationarity for joint processes in terms of joint filtering.

Definition 1 (Joint stationarity)**.** *A joint process $\boldsymbol{x} = vec(\boldsymbol{X})$ is called Jointly Wide-Sense Stationary (JWSS), if and only if*

(a) The first moment of the process is constant $\mathbf{E}[\boldsymbol{x}] = c\mathbf{1}_{NT}$.

(b) The covariance matrix of the process is a graph filter $\boldsymbol{\Sigma} = h(\boldsymbol{L}_G, \boldsymbol{L}_T)$, where $h(\cdot,\cdot)$ is a non-negative real function referred to as joint power spectral density (JPSD).

Let us examine Definition 1 in detail.

First moment condition. As in the classical case, the first moment of a JWSS process has to be constant over the time and the vertex sets, i.e., $\bar{\boldsymbol{X}}[i,t] = c$ for every $i = 1, 2, \ldots, N$ and $t = 1, 2, \ldots, T$. For alternative choices of the graph Laplacian with a null-space not spanned by the constant vector, the first moment condition should be modified to requiring that the expected value of a JWSS process is in the null space of the matrix $\boldsymbol{L}_T \oplus \boldsymbol{L}_G$ (see Remark 1 [16] for a similar observation on graph processes).

Second moment condition. According to the definition, the covariance matrix of a JWSS process takes the form of a joint filter $h(\boldsymbol{L}_G, \boldsymbol{L}_T)$, and is therefore diagonalizable by the JFT matrix $\boldsymbol{U}_J$. It may also be intesting to notice that the matrix $h(\boldsymbol{L}_G, \boldsymbol{L}_T)$ can be expressed as follows

$$\boldsymbol{\Sigma} = h(\boldsymbol{\Lambda}_G, \boldsymbol{\Omega}) = \begin{pmatrix} \boldsymbol{H}_{1,1} & \boldsymbol{H}_{1,2} & \cdots & \boldsymbol{H}_{1,T} \\ \boldsymbol{H}_{2,1} & \boldsymbol{H}_{2,2} & & \boldsymbol{H}_{2,T} \\ \vdots & & \ddots & \vdots \\ \boldsymbol{H}_{T,1} & \boldsymbol{H}_{1,2} & \cdots & \boldsymbol{H}_{T,T} \end{pmatrix}. \quad (4)$$

where

$$\boldsymbol{H}_{t_1,t_2} = \frac{1}{T} \sum_{\tau=1}^{T} h_{\omega_\tau}(\boldsymbol{L}_G)\, e^{j\omega_\tau(t_1 - t_2)} \quad (5)$$

and $h_{\omega_\tau}(\boldsymbol{L}_G)$ is the graph filter $h_{\omega_\tau} = h(\lambda, \omega_\tau)$. Being a covariance matrix, $h(\boldsymbol{L}_G, \boldsymbol{L}_T)$ must necessarily be positive-semidefinite; thus $h(\cdot,\cdot)$ is real (the eigenvalues of every hermitian matrix are real) and non-negative. Also equivalently, that every zero mean JWSS process $\boldsymbol{x} = \text{vec}(\boldsymbol{X})$ can be generated by joint filtering $\boldsymbol{x} = h(\boldsymbol{L}_G, \boldsymbol{L}_T)^{1/2}\boldsymbol{\varepsilon}$ a white process $\boldsymbol{\varepsilon}$ with zero mean and identity covariance. The following theorem exploits these facts to provide an interpretation of JWSS processes in the joint frequency domain.

Theorem 1 (Frequency interpretation)**.** *A joint process $\boldsymbol{X}$ over a connected graph $\mathcal{G}$ is Jointly Wide-Sense Stationary (JWSS) if and only if:*

(a) The joint spectral modes are in expectation zero

$$\mathbf{E}\left[\hat{\boldsymbol{X}}[n,\tau]\right] = 0 \quad \textit{if } \lambda_n \neq 0 \textit{ and } \omega_\tau \neq 0.$$

(b) The joint graph spectral modes are uncorrelated

$$\mathbf{E}\left[\hat{\boldsymbol{X}}[n_1,\tau_1]\hat{\boldsymbol{X}}[n_2,\tau_2]\right] = 0,$$

whenever $n_1 \neq n_2$ or $\tau_1 \neq \tau_2$.

(c) There exists a non-negative function $h(\cdot,\cdot)$, referred to as joint power spectral density (JPSD), such that

$$\mathbf{E}\left[\hat{\boldsymbol{X}}[n,\tau]^2\right] - \mathbf{E}\left[\hat{\boldsymbol{X}}[n,\tau]\right]^2 = h(\lambda_n, \omega_\tau),$$

for every $n = 1, 2, \ldots, N$ and $\tau = 1, 2, \ldots, T$.

(For clarity, this and other proofs of the paper have been moved to the appendix.)

[2] As the correlation between two instants t_1 and t_2 depends only on the difference between these two instants $\mathbf{E}[\boldsymbol{x}[t_1]\boldsymbol{x}[t_2]] - \mathbf{E}[\boldsymbol{x}[t_1]]\,\mathbf{E}[\boldsymbol{x}[t_2]] = \boldsymbol{\gamma}[t_1 - t_2]$, the covariance matrix has to be circulant, a property that is shared by linear filters.

[3] The only exception: for graphs with repeated eigenvalues, the conditions $\mathbf{E}[\boldsymbol{x}] = c\mathbf{1}$ and $\boldsymbol{\Sigma} = h(\boldsymbol{L}_G)$ are sufficient but not necessary for Girault's graph stationarity definition [15].

There are two, slightly technical, points that should be clarified here. First, for real processes $\boldsymbol{X}$, which are the focus of this paper, the function h forming the joint filter should be symmetric w.r.t. ω, meaning that $h(\lambda,\omega) = h(\lambda,-\omega)$. This property can be easily derived from the definition of the Fourier transform. Second, whenever the graph Laplacian features repeated eigenvalues, the degrees of freedom of h decrease, as necessarily $h(\lambda_1,\omega) = h(\lambda_2,\omega)$ when $\lambda_1 = \lambda_2$. This restriction is motivated by two observations: (a) For an eigenspace with multiplicity greater than one, there exists an infinite number of possible eigenvectors corresponding to the different rotations in the space and the JPSD is in general ill-defined. The condition $h(\lambda_1,\omega) = h(\lambda_2,\omega)$ when $\lambda_1 = \lambda_2$ deals with this ambiguity, as it ensures that the JPSD is the same independently of the choice of eigenvectors. (b) If one were to pick a ring graph and only one time step ($T = 1$), this condition ensures that joint stationarity is equivalent to classic stationarity in the periodic discrete case. We refer to [14, Section III B] for a detailed discussion.

We briefly present two additional properties of JWSS processes that will be useful in the rest of the paper.

Property 1 (White noise)**.** *White centered i.i.d. noise $\boldsymbol{w} \in \mathbb{R}^{NT} \sim \mathcal{D}(\boldsymbol{0}_{NT}, \boldsymbol{I}_{NT})$ is JWSS with constant JPSD for any graph.*

The proof follows easily by noting that the covariance of $\boldsymbol{w}$ is diagonalized by the joint Fourier basis of any graph $\boldsymbol{\Sigma}_{\boldsymbol{w}} = \boldsymbol{I} = \boldsymbol{U}_J \boldsymbol{I} \boldsymbol{U}_J^*$. This last equation tells us that the JPSD is constant, which implies that similar to the classical case, white noise contains all joint frequencies.

A second interesting property of JWSS processes is that stationarity is preserved through a filtering operation.

Property 2. *When a joint filter $f(\boldsymbol{L}_G, \boldsymbol{L}_T)$ is applied to a JWSS process $\boldsymbol{X}$ with JPSD h, the result $\boldsymbol{Y}$ remains JWSS with mean $cf(0,0)\boldsymbol{1}_{NT}$ and JPSD $f^2(\lambda,\omega)\, h(\lambda,\omega)$.*

A. Relations to classical definitions

We next provide an in depth examination of the relations between joint stationarity, the classical definition of time stationarity and that of vertex stationarity.

If no assumptions are made about the process, the covariance is simply

$$\boldsymbol{\Sigma} = \begin{pmatrix} \boldsymbol{\Sigma}_{1,1} & \boldsymbol{\Sigma}_{1,2} & \cdots & \boldsymbol{\Sigma}_{1,T} \\ \boldsymbol{\Sigma}_{2,1} & \boldsymbol{\Sigma}_{2,2} & & \\ \vdots & & \ddots & \vdots \\ \boldsymbol{\Sigma}_{T,1} & & \cdots & \boldsymbol{\Sigma}_{T,T} \end{pmatrix}.$$

When assuming that a process is JWSS, we in fact enforce that the statistical relation of variables at a given time-step $\boldsymbol{\Sigma}_{t_1,t_1}$ and those across different timesteps $\boldsymbol{\Sigma}_{t_1,t_2}$ should depend on the graph, as well as the time difference $t_1 - t_2$. The properties of the covariance matrix of a JWSS process can be decomposed into time and vertex dependencies.

1) JWSS $\subset$ TWSS. Similar to time stationary processes, the covariance $\boldsymbol{\Sigma}$ of a JWSS process has a block circulant structure, as $\boldsymbol{\Sigma}_{t_1,t_2} = \boldsymbol{\Sigma}_{\delta,1} = \boldsymbol{\Gamma}_\delta$, where $\delta = t_1 - t_2 + 1$. Hence the covariance matrix can be written as

$$\boldsymbol{\Sigma}_{\boldsymbol{x}} = \begin{pmatrix} \boldsymbol{\Gamma}_1 & \boldsymbol{\Gamma}_2 & \cdots & \boldsymbol{\Gamma}_T \\ \boldsymbol{\Gamma}_T & \boldsymbol{\Gamma}_1 & & \boldsymbol{\Gamma}_{T-1} \\ \vdots & & \ddots & \vdots \\ \boldsymbol{\Gamma}_2 & \boldsymbol{\Gamma}_3 & \cdots & \boldsymbol{\Gamma}_1 \end{pmatrix},$$

implying that correlations only depend on δ and not on any time localization. This property is shared by multivariate time wide sense stationary processes.

Definition 2 (Multivariate time stationarity)**.** *A joint process $\boldsymbol{X}$ is Time Wide-Sense Stationary (MTWSS), if and only if the following two properties hold*

(a) The expected value is constant as $\mathbf{E}[\boldsymbol{x}_t] = c\boldsymbol{1}$ for all t.
(b) For all t_1, t_2 the second moment satisfies $\boldsymbol{\Sigma}_{t_1,t_2} = \boldsymbol{\Sigma}_{\delta,1} = \boldsymbol{\Gamma}_\delta$, where $\delta = t_1 - t_2 + 1$.

Similarly to the univariate case, the Time Power Spectral Density (TPSD) is defined so as to encode the statistics of the process in the spectral domain:

$$\hat{\boldsymbol{\Gamma}}_\tau = \sum_{\delta=1}^{T} \boldsymbol{\Gamma}_\delta e^{-j\omega_\tau \delta} \tag{6}$$

We can also obtain the TPSD of a JWSS process by constructing a graph filter from h while fixing ω. Setting $h_{\omega_\tau}(\lambda) = h(\lambda, \omega_\tau)$, the TPSD of a JWSS process is $\hat{\boldsymbol{\Gamma}}_\tau = h_{\omega_\tau}(\boldsymbol{L}_G)$.

2) JWSS $\subset$ VWSS. It follows from Definition that, for a JWSS process, each block of $\boldsymbol{\Sigma}$ has to be a linear graph filter, i.e., $\boldsymbol{\Sigma}_{t_1,t_2} = \gamma_{t_1,t_2}(\boldsymbol{L}_G)$. Hence, the covariance matrix can be written as

$$\boldsymbol{\Sigma} = \begin{pmatrix} \gamma_{1,1}(\boldsymbol{L}_G) & \gamma_{1,2}(\boldsymbol{L}_G) & \cdots & \gamma_{1,T}(\boldsymbol{L}_G) \\ \gamma_{2,1}(\boldsymbol{L}_G) & \gamma_{2,2}(\boldsymbol{L}_G) & & \\ \vdots & & \ddots & \vdots \\ \gamma_{T,1}(\boldsymbol{L}_G) & & \cdots & \gamma_{T,T}(\boldsymbol{L}_G) \end{pmatrix}.$$

The concept of stationarity has been generalized to graph signals [14], [15], [16]. For no repeated eigenvalues, all the above state that a random signal is stationary on a graph if its expected value is constant on the vertex set, and the covariance matrix is jointly diagonalizable with the Laplacian, i.e., $\boldsymbol{\Sigma}_{\boldsymbol{x}_t} = h(\boldsymbol{L}_G)$. This notion of stationarity does not apply to time evolving processes as it does not characterize the correlation between different time-steps. As a result, we present here a generalization of this framework to timeseries on a graph.

Definition 3 (Multivariate vertex stationarity)**.** *A joint process $\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_T]$ is called Multivariate Vertex Wide-Sense Stationary (MVWSS), if and only if the following two properties hold independently:*

(a) The expected value is of the signal is constant $\mathbf{E}[\boldsymbol{x}_t] = c_t\boldsymbol{1}$ for all t.
(b) For all t_1 and t_2, we have $\boldsymbol{\Sigma}_{t_1,t_2}$, there exist a kernel γ_{t_1,t_2} such that $\boldsymbol{\Sigma}_{t_1,t_2} = \gamma_{t_1,t_2}(\boldsymbol{L}_G)$.

3) JWSS = TWSS $\cap$ VWSS. As shown next, the two definitions taken together are equivalent to that of joint stationarity.

Theorem 2. *A joint process $\boldsymbol{X}$ is JWSS if and only if it is both MTWSS and MVWSS.*

In other words, the set of processes that are JWSS are exactly those that are statistically invariant in the temporal and vertex domains.

IV. Joint Power Spectral Density estimation

The joint stationarity assumption can be very effective in overcoming the challenges associated with dimensionality. The main reason is that, for JWSS processes, the estimation variance is decoupled from the problem size. Concretely, suppose that we want to estimate the covariance matrix $\boldsymbol{\Sigma}$ of a joint process $\boldsymbol{x} = \operatorname{vec}(\boldsymbol{X})$ from K samples $\boldsymbol{x}_{(1)}, \boldsymbol{x}_{(2)}, \ldots, \boldsymbol{x}_{(K)}$. As we show in the following, if the process is JWSS such that $\boldsymbol{\Sigma} = h(\boldsymbol{L}_G, \boldsymbol{L}_T)$, estimation is possible from $K = O(1)$ samples! This is a sharp decrease from the classical and MTWSS settings, for which $K \approx NT$ and $K \approx N$ samples are necessary[4], respectively.

This section presents two JPSD estimators requiring constant number of samples. The first provides unbiased estimates at complexity that is $O(N^3 T \log(T))$. The second estimator, decreases further the estimation variance at a cost of a bounded bias, and can be approximated at complexity linear to ET.

A. Sample JPSD estimator

We define the "sample JPSD estimator" for every graph frequency λ_n and temporal frequency ω_τ as the estimate

$$\dot{h}(\lambda_n, \omega_\tau) \triangleq \sum_{k=1}^{K} \frac{\left|\mathrm{JFT}\{\boldsymbol{X}_{(k)}\}[n,\tau]\right|^2}{K}. \tag{7}$$

In case the process does not have zero mean, it should be centered by subtracting the constant signal $c\,\mathbf{1}_N \mathbf{1}_T^*$, where $c = \sum_{k,i,t} \boldsymbol{X}_{(k)}[i,t]$. For simplicity, suppose that the process is correctly centered. As the following theorem claims, the sample JPSD estimator is unbiased and its variance decreases linearly with the number of samples K.

Theorem 3. *For every distribution with bounded second and fourth order moments, the sample JPSD estimator $\dot{h}(\theta)$*

(a) is unbiased, i.e., $\mathbf{E}\left[\dot{h}(\theta)\right] = h(\theta)$, and

(b) has variance $\mathbf{Var}\left[\dot{h}(\theta)\right] = h^2(\theta)\,\dfrac{\gamma - 1}{K}$,

where constant γ depends only on the distribution of $\boldsymbol{x}$.

Proof. For any $\theta = [\lambda, \omega]$, the sample estimate is

$$\dot{h}(\theta) = h(\theta) \sum_{k=1}^{K} \frac{\hat{\varepsilon}_{(k)} \hat{\varepsilon}^*_{(k)}}{K}, \tag{8}$$

with $\hat{\varepsilon}_{(k)}$ being independent realizations of $\hat{\varepsilon}$, a zero mean complex random variable with unit variance. To see this, write $\boldsymbol{x} = h(\boldsymbol{L}_G, \boldsymbol{L}_T)^{1/2} \boldsymbol{\varepsilon}$, where the random vector $\boldsymbol{\varepsilon}$ has zero mean and identity covariance. Then, the complex random variable $\hat{\varepsilon}$ is the JFT coefficient of $\boldsymbol{\varepsilon}$ corresponding to frequencies λ and ω. The bias follows by noting that $\mathbf{E}\left[\hat{\varepsilon}_{(k)} \hat{\varepsilon}^*_{(k)}\right] = 1$, for every k. The variance is computed similarly by exploiting the fact that different terms in the sum are independent as they correspond to distinct realizations and setting $\gamma = \mathbf{E}\left[|\hat{\varepsilon}|^4\right]$. □

For the standard case of a Gaussian joint process, we provide an exact characterization of the distribution.

Corollary 1. *For every Gaussian JWSS process, the sample JPSD estimate follows a Gamma distribution with shape $K/2$ and scale $2h(\theta)/K$. The estimation error variance is equal to $\mathbf{Var}\left[\dot{h}(\theta)\right] = 2\,h^2(\theta)/K$.*

Proof. We continue in the context of the proof of Theorem 3. For a Gaussian distribution, $\hat{\varepsilon}$ is centered and scaled Gaussian and thus $\hat{\varepsilon}^2$ is a chi-squared random variable with 1 degree of freedom. Our estimate is therefore a scaled sum of i.i.d. chi-squared variables and corresponds to a Gamma distribution. The corollary then follows directly. □

Observe that the variance depends linearly on the fourth other moment of $|\hat{\varepsilon}|$ (see proof of Theorem 3) and is inversely proportional to the number of samples, but it is independent of N and T. In the following, we show how to achieve an even smaller variance by exploiting the properties of $h(\theta)$. In addition, we reduce the estimation accuracy by avoiding to perform an eigenvalue decomposition.

B. Convolutional JPSD estimator

When the number of available signals K is very small (even 1), we need an additional assumption on the correlations to obtain reasonable estimates. To this end, we next present a parametric JPSD estimator that allows us to trade off variance for bias.

Before delving into JWSS processes, it is helpful to consider the purely temporal case. For a TWSS process it is customary to assume that the autocorrelation function has support L that is a few times smaller than T. Then, cutting the signal into $\frac{T}{L}$ smaller parts and computing the average estimate, reduces the variance (by a factor of $\frac{T}{L}$), without sacrificing frequency resolution. This basic idea stems from two established methods used to estimate the PSD of a temporal signal, namely Bartlett's and Welch's methods [32], [33]. The act of averaging across different windows is in fact equivalent to a convolution in the spectral domain. Convolving the TPSD with a window, results in attenuation of the correlation for large delays, enforcing localization in the time domain.

Motivated by the observation that convolution with a window in the graph frequency domain also encourages localization in the vertex domain when the operation can be approximated by a polynomial with bounded order [30, Theorem 1 and Corollary 2], Perraudin and Vandergeynst proposed to reduce the estimation variance by convolving the sample GPSD [14]. In the following, we extend this idea to the joint domain. Concretely, Let $g(\theta)$ be a 2D window defined in the

[4] The number of samples needed for obtaining a good sample covariance matrix of an n-dimensional process is $O(n \log n)$ [1], [31].

joined frequency domain. We define our convolutional JPSD estimator as

$$\ddot{h}(\theta) \triangleq \frac{1}{c_g(\theta)} \sum_{\substack{n=1\\ \tau=1}}^{N,T} g(\theta - \theta_{n,\tau})^2 \, \dot{h}(\theta_{n,\tau}), \tag{9}$$

where, $c_g(\theta) \triangleq \sum_{n,\tau} g(\theta - \theta_{n,\tau})^2$ is a normalization factor that depends on the $\theta = (\lambda, \omega)$ frequency pair (since the graph eigenvalues are generally irregularly spaced). Moreover, $\dot{h}(\theta_{n,\tau})$ is the sample estimate defined in (7). Further implementation specifics, including a discussion on the choice of the 2D window g, are given in Section IV-C.

The convolutional JPSD estimator is a generalization of known PSD estimators for TWSS and GWSS processes. Denote by ϕ the dirac function. We have that: (a) For $g(\theta) = \phi(\lambda) \cdot g_T(\omega)$, we recover the classical TPSD estimator, applied independently for each λ. (b) For $g(\theta) = g_G(\lambda) \cdot \phi(\omega)$, we recover the GPSD estimator from [14] applied independently for each ω.

To provide a meaningful bias analysis, we introduce a Lipschitz continuity assumption on the JPSD, matching the intuition that localized phenomena tend to have a smooth representation in the frequency domain.

Theorem 4. *The convolutional JPSD estimator* $\ddot{h}(\theta)$

(a) has bias

$$\left|\mathbf{E}\left[\ddot{h}(\theta) - h(\theta)\right]\right| \leq \frac{\epsilon}{c_g(\theta)} \sum_{n=1, \tau=1}^{T,N} g(\theta - \theta_{n,\tau})^2 \, \|\theta - \theta_{n,\tau}\|_2 \,,$$

where ϵ *is the Lipschitz constant of* $h(\theta)$*, and*

(b) when the entries of $\hat{\boldsymbol{X}}$ *are independent random variables, its variance is*

$$\mathbf{Var}\left[\ddot{h}(\theta)\right] = \sum_{n,\tau} \frac{g(\theta - \theta_{n,\tau})^4}{c_g(\theta)^2} \, \mathbf{Var}\left[\dot{h}(\theta_{n,\tau})\right],$$

where $\mathbf{Var}\left[\dot{h}(\theta_{n,\tau})\right]$ *is the variance of the sample JPSD estimator at* $\theta_{n,\tau}$*.*

Proof. The derivations of the bias and variance are given in Lemmas 1 and 2, respectively. □

Let us consider as an example the case of a Gaussian JWSS process and a disc window with bandwidth B, i.e., $g_B(\theta) = 1$ if $\|\theta\|_2 \leq \frac{B}{2}$ and 0, otherwise. Though not necessarily localized in the graph domain, we choose here a disc window because it leads to simple and intuitive estimates.

Corollary 2. *For every* ϵ*-Lipschitz Gaussian JWSS process and disc window* $g_B(\theta)$*, the convolutional estimate has*

$$\left|\mathbf{E}\left[\ddot{h}(\theta) - h(\theta)\right]\right| \leq \frac{\epsilon B}{2} \quad \text{and} \quad \mathbf{Var}\left[\ddot{h}(\theta)\right] = \frac{2\, h_{\mathcal{S}}^2}{K|\mathcal{S}|}, \tag{10}$$

with set $\mathcal{S} = \{\theta_{n,\tau} \mid \|\theta_{n,\tau} - \theta\|_2 \leq B/2\}$ *and* $h_{\mathcal{S}}^2 = \sum_{\theta_{n,\tau} \in \mathcal{S}} \frac{h(\theta_{n,\tau})^2}{|\mathcal{S}|}$.

Proof. The results follow from Theorem 4 and Corollary 1 by noting that when a disc window is used: (a) $c_g(\theta) = |\mathcal{S}|$, and (b) $g(\theta - \theta_{n,\tau})^2 = 1$ for all n, τ in the window (there are $|\mathcal{S}|$ in total) and zero otherwise. The independence condition required by the variance clause of the theorem is satisfied since $\hat{\boldsymbol{x}}$ is Gaussian (as a rotation $\hat{\boldsymbol{x}} = \boldsymbol{U}_J^* \boldsymbol{x}$ of a Gaussian vector) with diagonal covariance. □

The above result suggests that, by selecting our window (bandwidth) we can trade off bias for variance. The trade-off is particularly beneficial as long as (a) the JPSD is smooth, and (b) the graph eigenevalues are clustered, such that $|\mathcal{S}| \gg B$. We also note that a special case of our results ($T = 1$) is novel also for the purely graph setting [14].

C. Fast implementation

Having defined the convolutional JPSD estimator, we turn to its computation. A straightforward implementation requires: $O(N^3)$ operations for computing the eigenbasis of our graph, $O(N^2 \times KT)$ for performing KT independent GFT, $O(T \log(T) \times KN)$ for KN independent FFT, and $O(N^2 T^2)$ for the convolution.

This section describes how to approximate a convolutional estimate using a number of operations that is linear to ET. Before describing the exact algorithm, we note two helpful properties of the estimator. First, we can compute $\ddot{h}(\theta)$ by obtaining estimates for each $\boldsymbol{X}_{(k)}$ independently and then averaging over k:

$$\dot{h}(\theta_{n,\tau}) = \frac{1}{K\, c_g(\theta)} \sum_{k} \sum_{n,\tau} g(\theta - \theta_{n,\tau})^2 \, |\mathrm{JFT}\{\boldsymbol{X}_{(k)}\}[n, \tau]|^2$$

As we will see in the following, the terms inside the outer sum can be approximated efficiently, avoiding the need for an expensive JFT. In addition, when the convolution window is separable, i.e., $g(\theta) = g_G(\lambda) \cdot g_T(\omega)$, as is assumed here, the joint convolution can be performed independently (and at any order) in the time and vertex domains

$$\ddot{h}(\theta) = \sum_{\tau} \frac{g_T(\omega - \omega_\tau)^2}{c_{g_T}(\omega)} \left(\sum_{n} \frac{g_G(\lambda - \lambda_n)^2}{c_{g_G}(\lambda)} \, \dot{h}(\theta_{n,\tau}) \right),$$

where $c_g(\theta) = c_{g_T}(\omega) \cdot c_{g_G}(\lambda)$. Exploiting this property, we treat the implementation of the two convolutions separately and the presented algorithms can be combined in any order.

Fast time convolution. This is the textbook case of TPSD estimation, that is solved by the Welch's method [33]. The method entails splitting each timeseries into equally sized overlapping segments, and averaging over segments the squared amplitude of the Fourier coefficients. The procedure is equivalent to an averaging (over time) of the squared coefficients of a Short Time Fourier Transform (STFT), with half overlapping windows w_T defined such that $\mathrm{DFT}\{w_T(t)\} = g_T(\omega)$ [34], [35]. Let L be the support of the autocorrelation, or equivalently the number of frequency bands. We suggest using the iterated sine window

$$w_T(t) \triangleq \begin{cases} \sin\left(0.5\pi \cos\left(\pi t / L\right)^2\right) & \text{if } t \in [-L/2, L/2] \\ 0 & \text{otherwise,} \end{cases}$$

as it turns the STFT into a tight operator. In order to get an estimate of $\ddot{h}$ at unknown frequencies, we interpolate between the L known points using splines [36].

Fast graph convolution. Inspired by the technique of [14], we perform the graph convolution using an approximated graph filtering operation [37] that scales linearly to the number of graph edges E. In particular,

$$\sum_{n=1}^{N} \frac{g_G(\lambda-\lambda_n)^2}{c_{g_G}(\lambda)}\, \dot{h}(\theta_{n,\tau}) = \frac{\|g_G(\boldsymbol{L}_G-\lambda \boldsymbol{I}_N)\, \boldsymbol{x}_\tau\|_2^2}{c_{g_G}(\lambda)}. \quad (11)$$

We suggest using the Gaussian window

$$g_G(\lambda_n-\lambda) \triangleq e^{-(\lambda_n-\lambda)^2/\sigma^2}, \quad (12)$$

with $\sigma^2 = 2(F+1)\lambda_{\max}/F^2$. As we did before, we only compute the above for $F = O(1)$ different values of λ and approximate the rest using splines. As the eigenvalues are not known, we need a stable way to estimate $c_{g_G}(\lambda)$. We obtain an unbiased estimate by filtering $Q = O(1)$ random Gaussian signals on the graph $\boldsymbol{\varepsilon} \in \mathbb{R}^N \sim \mathcal{N}(0, \boldsymbol{I}_N)$, such that

$$c_{g_G}(\lambda) = \mathbf{E}\left[\sum_{q=1}^{Q} \|g_G(\boldsymbol{L}_G-\lambda \boldsymbol{I}_N)\boldsymbol{\varepsilon}_{(q)}\|_2^2\right], \quad (13)$$

with variance equal to $2\sum_{n=1}^{N} g^4(\lambda_n-\lambda)/Q$. We omit the analysis, as it is similar to that in Theorem 3. According to our numerical evaluation, the approximation error introduced by the latter estimator and spectral filtering is almost negligible for smooth JPSD.

Complexity. The computational cost of the above methods is: (a) $O(TKF\times E + QF\times E) = O((TK+Q)EF)$ for the fast graph convolutions. Here, the TK and Q convolutions are performed in order to estimate the quantities at (11) and (13) for F different values of λ. (b) $O(NK \times T\log(L))$ for the fast time convolution, corresponding to NK STFT. Thus, in total the complexity of the fast convolutional JPSD estimator is $O(TKFE + QEF + NKT\log(L))$. Furthermore, when Q, F, K are constants, the complexity simplifies to $O(TE + NT\log(L))$. We remark that, though asymptotically superior, the fast implementation can be significantly slower when the number of variables is small. Our experiments demonstrate that it should be preferred for N larger than a few thousands (see Figure 2).

V. Recovery of JWSS Processes

This section considers the MMSE problem of recovering a hidden JWSS process $\boldsymbol{x} = \operatorname{vec}(\boldsymbol{X})$ from linear measurements $\boldsymbol{y}$ corrupted by a zero-mean JWSS process $\boldsymbol{w}$:

$$\begin{aligned} \min_{f:\mathbb{R}^N\to\mathbb{R}^N} \quad & \mathbf{E}\left[\|f(\boldsymbol{y})-\boldsymbol{x}\|_2^2\right] \\ \text{subject to} \quad & \boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}+\boldsymbol{w} \end{aligned} \quad \text{(P0)}$$

We remark that (a) for $\boldsymbol{A}$ binary diagonal and $\boldsymbol{w}=\boldsymbol{0}$, (P0) is an *interpolation* problem, (b) for $\boldsymbol{A}$ diagonal with $\boldsymbol{A}_{ii}=1$ if $i \leq Nt$ and zero otherwise and $\boldsymbol{w}=\boldsymbol{0}$ it corresponds to *forecasting*, and (c) for $\boldsymbol{A}=\boldsymbol{I}$ and $\boldsymbol{w}$ white noise (P0) is a *denoising* problem.

Since the solution of (P0) is in general distribution dependent, we additionally postulate that the function f is linear on $\boldsymbol{y}$, i.e., there exists a matrix $\boldsymbol{W}$ and a vector $\boldsymbol{b}$ such that $f(\boldsymbol{y}) = \boldsymbol{W}\boldsymbol{y}+\boldsymbol{b}$. The *minimum mean-squared linear estimate* is then known to be

$$\dot{\boldsymbol{x}} \;= \boldsymbol{\Sigma}_{\boldsymbol{xy}}\boldsymbol{\Sigma}_{\boldsymbol{y}}^{-1}(\boldsymbol{y}-\bar{\boldsymbol{y}})+\bar{\boldsymbol{x}}. \quad (14)$$

Above, $\boldsymbol{\Sigma}_{\boldsymbol{y}} = \boldsymbol{A}\boldsymbol{\Sigma}\boldsymbol{A}^* + \boldsymbol{\Sigma}_{\boldsymbol{w}}$ and $\boldsymbol{\Sigma}_{\boldsymbol{xy}} = \boldsymbol{\Sigma}\boldsymbol{A}^*$. Obtaining $\dot{\boldsymbol{x}}$ therefore entails solving a linear system in matrix $\boldsymbol{\Sigma}_{\boldsymbol{y}}$, that -naively approached- has $O(N^2T^2)$ complexity. In addition, especially in the noise-less case, the condition number of $\boldsymbol{\Sigma}_{\boldsymbol{y}}$ can be infinite, rendering direct inversion unfeasible.

We next discuss how to deal with these issues:

Decreasing the complexity. Thankfully, even if $\boldsymbol{\Sigma}_{\boldsymbol{y}}$ is not always sparse, we can approximate its multiplication by a vector very efficiently as (a) $\boldsymbol{A}$ usually is very sparse, and (b) per our assumption $\boldsymbol{\Sigma}$ and $\boldsymbol{\Sigma}_{\boldsymbol{w}}$ are joint filters and therefore they can be implemented at complexity that is (up to logarithmic factors) linear to the number of edges E and timesteps T [12], [29], [27]. Therefore, if we employ an iterative method such as the (preconditioned) conjugate gradient to compute the solution, the complexity of each iteration will be almost linear on the problem size.

Singular or badly conditioned $\boldsymbol{\Sigma}_{\boldsymbol{y}}$. In this case, we choose the solution with the minimal residual

$$\dot{\boldsymbol{x}} \;= \boldsymbol{\Sigma}_{\boldsymbol{xy}}\boldsymbol{\Sigma}_{\boldsymbol{y}}^{+}(\boldsymbol{y}-\bar{\boldsymbol{y}})+\bar{\boldsymbol{x}}. \quad (15)$$

Instead of solving the normal equations

$$\dot{\boldsymbol{x}} = \boldsymbol{\Sigma}_{\boldsymbol{xy}}(\boldsymbol{\Sigma}_{\boldsymbol{y}}^2)^{-1}\boldsymbol{\Sigma}_{\boldsymbol{y}}(\boldsymbol{y}-\bar{\boldsymbol{y}})+\bar{\boldsymbol{x}},$$

which has the effect of significantly increasing the condition number of our matrix, we suggest to employ the minimal residual conjugate gradient method for symmetric matrices [38]. The latter, though it is guaranteed to converge in at most NT iterations, has often much faster convergence. For badly conditioned covariance matrices, an alternative solution is to rewrite the problem as a regularized least squares problem

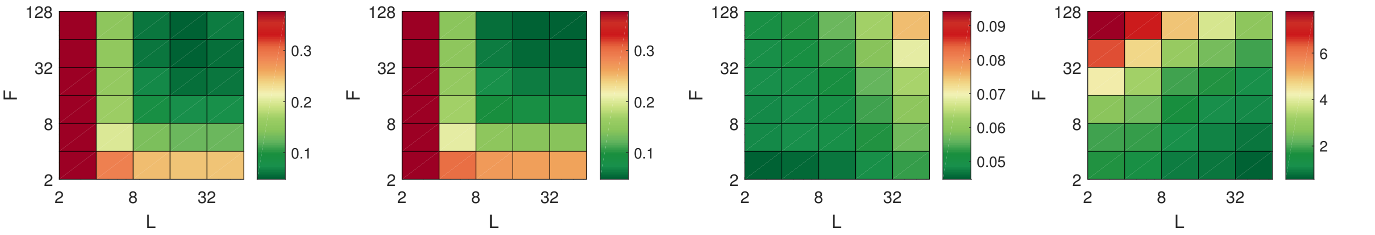
$$\min_{\boldsymbol{z}\in\mathbb{R}^N} \|\boldsymbol{A}\boldsymbol{z}-\boldsymbol{y}\|_2^2 + \|h_{\boldsymbol{w}}(\boldsymbol{L}_G,\boldsymbol{L}_T)^{1/2}\, h_{\boldsymbol{x}}(\boldsymbol{L}_G,\boldsymbol{L}_T)^{-1/2}(\boldsymbol{z}-\bar{\boldsymbol{x}})\|_2^2 \quad (16)$$

and solve it using a fast iterative shrinkage-thresholding algorithm (FISTA) scheme [39], [40], [41]. This problem was shown to converge to the correct solution when $\boldsymbol{w}$ is white noise [14]. There is a good reason for transforming the problem in this way: the FISTA convergence is a linear function of $2\,\|\boldsymbol{A}^*\boldsymbol{A}\|_2$, i.e., the Lipschitz constant of the gradient of $\|\boldsymbol{A}\boldsymbol{z}-\boldsymbol{y}\|_2^2$, but not the condition number of $h_{\boldsymbol{w}}(\boldsymbol{L}_G,\boldsymbol{L}_T)$ and $h_{\boldsymbol{x}}(\boldsymbol{L}_G,\boldsymbol{L}_T)$ [42]. As such, it convergences faster when the conditioning of $\boldsymbol{\Sigma}_{\boldsymbol{y}}$ is very poor and $\boldsymbol{A}$ is well behaved. Similarly, in the noise-less case one solves

$$\begin{aligned} \min_{\boldsymbol{z}\in\mathbb{R}^N} \quad & \|h_{\boldsymbol{x}}^{-1/2}(\boldsymbol{L}_G,\boldsymbol{L}_T)\;(\boldsymbol{z}-\bar{\boldsymbol{x}})\,\|_2^2 \\ \text{subject to} \quad & \boldsymbol{A}\boldsymbol{z}=\boldsymbol{y}, \end{aligned} \quad (17)$$

using a Douglas-Rachford scheme [43].

A special case. When matrix $\boldsymbol{A}$ is a joint filter and therefore $\boldsymbol{A} = a(\boldsymbol{L}_G,\boldsymbol{L}_T)$, the solution can be obtained by a single



(a) estimation error (b) bias (c) variance (d) execution time (sec)

Fig. 1: Influence of the parameters (window size L and number of graph filters F) on the (a) estimation error, (b) bias, (c) normalized std. dev., and (d) execution time. For improved visibility, the scale of (c) has been changed.

application of a Wiener-type filter $f(\boldsymbol{L}_G, \boldsymbol{L}_T)$ with

$$f(\lambda,\omega) = \frac{h_{\boldsymbol{x}}(\lambda,\omega)\,a(\lambda,\omega)}{a^2(\lambda,\omega)h_{\boldsymbol{x}}(\lambda,\omega) + h_{\boldsymbol{w}}(\lambda,\omega)}. \tag{18}$$

The most common case when this happens is when solving a denoising problem, since $\boldsymbol{A} = \boldsymbol{I}$ corresponds to the trivial joint filter with $a(\lambda,\omega) = 1$, for all λ and ω. Wiener filters were classically proposed in [44]. The first generalizations to graph signals appear in [45] and [22, pp 100] and were studied in more detail in [14].

VI. Experiments

A. Joint Power Spectral Density Estimation

The first step in our evaluation is to analyze the efficiency of JPSD estimation. Our objective is dual. First, we aim to study the role of the different method parameters into the estimation accuracy and computational complexity, essentially providing practical guidelines for their usage. In addition, we wish to illustrate the usefulness of the joint stationarity assumption in learning from few samples, even when the graph is only approximately known.

Variance-bias-complexity tradeoffs. To validate the analysis of Section IV-C for the computational and accuracy tradeoffs inherent to our JPSD estimation method, we performed numerical experiments with random geometric graphs ($N = 256$ vertices and average degree slightly above 7) and JWSS processes ($T = 128$ timesteps). For simplicity, we focus on the standard setting of a Gaussian process with smooth and separable JPSD that is exponentially decreasing with frequency: $h(\theta) = e^{-\lambda/\lambda_{max}}\, e^{-5\,\omega^2}$. In our experience, similar JPSD are commonly found in data with smooth spatio-temporal structure, such as for instance in meteorological datasets. We remark that the presented results were found consistent with those obtained for non-separable JPSD. In this section, we examine the relation between the real JPSD $\boldsymbol{H} = h(\boldsymbol{\Lambda}_G, \boldsymbol{\Omega})$ and the convolutional estimate $\ddot{\boldsymbol{H}} = \ddot{h}(\boldsymbol{\Lambda}_G, \boldsymbol{\Omega})$. We use the following metrics:

$$\overset{\textit{error}}{\frac{\tilde{\mathbf{E}}\left[\left\|\ddot{\boldsymbol{H}} - \boldsymbol{H}\right\|_F\right]}{\|\boldsymbol{H}\|_F}} \quad\Bigg|\quad \overset{\textit{bias}}{\frac{\left\|\tilde{\mathbf{E}}\left[\ddot{\boldsymbol{H}}\right] - \boldsymbol{H}\right\|_F}{\|\boldsymbol{H}\|_F}} \quad\Bigg|\quad \overset{\textit{variance}}{\frac{\tilde{\mathbf{E}}\left[\left\|\ddot{\boldsymbol{H}} - \tilde{\mathbf{E}}\left[\ddot{\boldsymbol{H}}\right]\right\|\right]_F}{\|\boldsymbol{H}\|_F}},$$

where $\tilde{\mathbf{E}}[\cdot]$ is the sample average from 20 independent experiments.

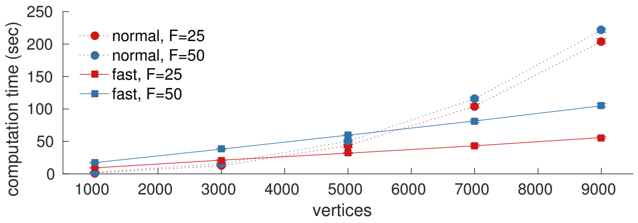
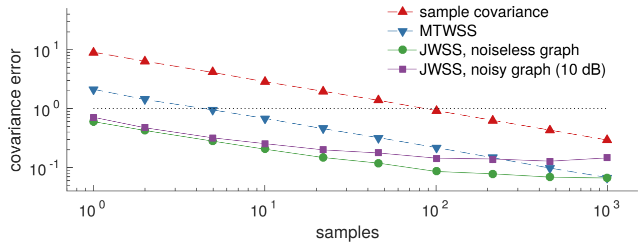


Fig. 2: Scalability of the convolutional JPSD estimator. The fast implementation should be favored when the graph is composed of more than a few thousands vertices. The approximation error inherent in the fast implementation was negligible in our experiments.

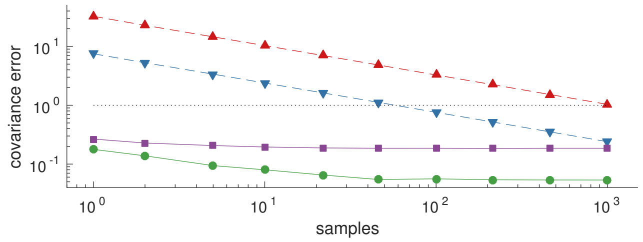
We remind the reader that there are two parameters influencing the performance of the convolutional JPSD estimator: the window size L corresponding to our assumption for the support length of the autocorellation in time, and the number of graph filters F used to capture power density in the graph spectral dimension. Figures 1 (a-d) report four key metrics for an exhaustive search of L, F combinations. We observe that large values of F and L generally reduce the estimation error (Figure 1a) because they result in reduced bias (Figure 1b). Nevertheless, setting the parameters to their maximum values is not suggested as the variance is increased (Figure 1c). In Figure 1d we see that, utilizing a large number of filters and number of windows (i.e., large F and small L), increases the average execution time.

Figure 2 delves further into the issue of scalability. In particular, we examine the min/median/max execution time of the convolutional JPSD estimator for increasing problem sizes when run in a desktop computer. Though the setting is identical to the previous experiments, here the number of vertices is varied from 1000 to 9000. We compare two implementations. The first, which naively performs the convolution in the spectral domain, uses the eigenvalue decomposition and therefore scales quadratically with the number of vertices. Due to its optimized code and simplicity, this should be the method of choice when N is small. For larger problems, we suggest using the fast implementation. As shown in the figure, this implementation scales linearly with N (here $E = O(N)$) when the number of filters F and timesteps T are held constant.

How to choose L and F? Having no computational constrains, one should choose the parameter combination that



(a) $N = 10, \ T = 10$



(b) $N = 100, \ T = 10$

Fig. 3: Even an approximate knowledge of the graph enables us to make good estimates of the covariance (and PSD) from very few samples. The joint stationarity prior becomes especially meaningful when the number of variables (N, T) increases.

minimized the Akaike information criterion (AIC) score $\text{AIC} = 2FL - 2\ln(\ddot{\ell})$, where $\ddot{\ell}$ is the distribution dependent estimated likelihood $\ddot{\ell} = \mathbf{P}(\boldsymbol{x}|\ddot{\boldsymbol{\Sigma}})$, and $\ddot{\boldsymbol{\Sigma}}$ is the estimated covariance based on the convolutional JPSD estimator with parameters L and F [46]. This procedure is often unfeasible as it is based on computing each model's log-likelihood and thus entails estimating one JPSD for each parameterization in consideration (as well as knowing the distribution type). We have found experimentally that setting $F = \min(N, 50)$ provides a good trade-off between computational complexity and error. On the other hand, we suggest setting L equal to an upper bound of the autocorrelation support.

Learning from few samples and a noisy graph. Figure 3 illustrates the benefit of a joint stationarity prior as compared to (a) a sample covariance estimator which makes no assumptions about the data, and (b) the multivariate TWSS process estimator with optimal bandwidth [17]. As expected, an accurate estimation is challenging when the number of samples is much smaller than the number of problem variables (NT), returning errors above one for the sample estimator. Introducing stationarity priors regularizes the estimation resulting in more stable estimates.

What is perhaps surprising is that, even when the graph (and $\boldsymbol{U}_G$) are known only approximately, estimating the second order moment of the distribution using the joint stationarity assumption is beneficial. To portray this phenomenon, we also plot the estimation error when using a noisy graph (we corrupted the weighted adjacency matrix by Gaussian noise, with SNR = 10 dB). Undoubtedly, introducing noise to the graph edges negatively affects estimation by introducing bias. Still, even with noise the proposed method significantly outperforms purely time-based methods when less than NT samples are available.

B. Recovery Performance on Three Datasets

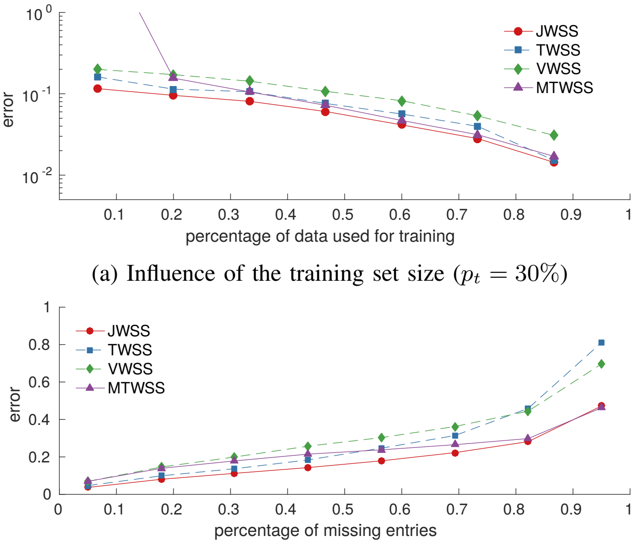
We apply our methods on three diverse datasets featuring multivariate processes evolving over graphs: (a) a weather dataset depicting the temperature of 32 weather stations over one month, (b) a traffic dataset depicting high resolution daily vehicle flow of 4 weekdays, and (c) SIRS-type epidemics in Europe. Our experiments aim to show that joint stationarity is a useful model, even in datasets which may violate the strict conditions of our definition, and that -especially when few samples are available- it can yield a significant improvement in recovery performance, as compared to time- or vertex-based methods, on real datasets.

Experimental setup. We split the K samples of each dataset into a *training set* of size p_tK and a *test set* of size $(1-p_t)K$, respectively. After estimating the JPSD from the training set, we attempt to recover the values of p_dNT variables randomly discarded from the test set. In each case, we report the RMSE for the recovered signal normalized by the ℓ_2-norm of the original signal. We compare our joint method to the state-of-the-art wiener filters assuming *univariate time/vertex stationarity* [14]. Univariate stationarity methods solve the statistical recovery problem under the assumption that signals at stationary in the time/vertex domains, but considering different vertices/timesteps as independent. These methods are known to outperform non-model based methods, such as Tikhonov regularization (ridge regression) and total-variation regularization (lasso) over the time or graph dimensions [8], [9]. We also compare to the more involved *multivariate TWSS model* where the values at different vertices are correlated and the covariance is block Circulant of size $NT \times NT$. The latter is only shown for the weather dataset as the large number of variables present in the other datasets (e.g., $\approx 10^8$ parameters for the traffic dataset) prohibited computation.

Molene dataset. The French national meteorological service has published in open access a dataset[5] with hourly weather observations collected during the Month of January 2014 in the region of Brest (France). The graph was built from the coordinates of the weather stations by connecting all the neighbors in a given radius with a weight function $\boldsymbol{W}_G[i_1, i_2] = \exp(-k\, d(i_1, i_2)^2)$, where $d(i_1, i_2)$ is the euclidean distance between the stations i_1 and i_2. Parameter k was adjusted to as obtain an average degree around 5 (k however is not a sensitive parameter). We split the data in $K = 15$ consecutive periods of $T = 48$ hours each. As sole pre-processing, we removed the mean (over time and stations) of the temperature. Since NT is here relatively small, we used the sample JPSD estimator.

We first test the influence of training set size p_t, while discarding $p_d = 30\%$ of the test variables. As seen in Figure 4a, the multivariate TWSS approach provides good recovery estimates when the when the number of samples

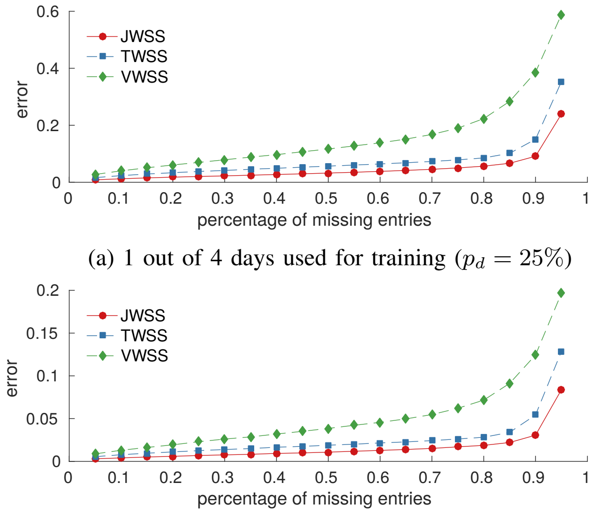
[5] Access to the raw data is possible directly from https://donneespubliques.meteofrance.fr/donnees_libres/Hackathon/RADOMEH.tar.gz



(a) Influence of the training set size ($p_t = 30\%$)

(b) Influence of the percentage of missing values ($p_d = 20\%$)

Fig. 4: Experiments with weather data. The joint approach becomes especially meaningful when the available data are few.



(a) 1 out of 4 days used for training ($p_d = 25\%$)

(b) 3 out of 4 days used for training ($p_d = 75\%$)

Fig. 5: Experiments on Sacramento highway flow. By exploiting both graph and temporal dimensions, the joint approach closely captures the subtle variations in traffic throughout each weekday.

is large, approaching that of joint stationarity, but suffers for small training sets (though not shown in the figure, the mean error was 9.8 when only $p_t = 10\%$ of the data was used for training). Due to their stricter modeling assumptions, disjoint stationarity methods returned relevant estimates when trained from very few samples, but exhibited larger bias. Figure 4b reports the achieved errors for recovery problems with progressively larger percentage $5\% \leq p_d \leq 95\%$ of discarded entries for a training percentage of $p_t = 20\%$. We can observe that the error trends are consistent across all cases.

Traffic dataset. The California department of transportation publishes high-resolution traffic flow measurements (number of vehicles per unit interval) from stations deployed in the highways of Sacramento[6]. We focused at 727 stations over four weekdays in the period 01-06 April 2016. Starting from the road connectivity network obtained by the OpenStreetMap.org, we constructed one timeseries for each highway segment by setting the flow over it to be a weighted average of all nearby stations, while abiding to traffic direction. This resulted in a graph of $N = 710$ vertices, and a total of $T = 24 \times 12$ measurements per day for $K = 4$ days. We used the convolutional JPSD estimator with parameters $L = T/2$ and $F = 75$, which were experimentally found to give good performance in the training set.

Figures 5a and 5b depict the mean recovery errors when the training sets where 1 and 3 days respectively. The strong temporal correlations present in highway traffic were useful in recovering missing values. Considering both the temporal and spatial dimension of the problem, resulted in very accurate estimates, with less that 0.04 error when p_d =50% of the data were removed and the PSD was estimated from one day.

SIRS epidemic. Our third dataset depicts the spread of an infectious disease over $N = 200$ major cities of Europe, as predicted by the Succeptible-Infected-Recovered-Susceptible (SIRS) model, one of the standard models used to study epidemics. Our intention is to examine the predictive power of the considered methods when dealing with different realizations of a non-linear and probabilistic process over a graph (the data are fictitious). We parameterized SIRS as follows: length of infection period: 2 days, length of immunity period: 10 days, probability of contagion across neighboring cities per day: 0.005, total period: $T = 180$ days. We generated a total of $K = 10$ infections, all having the same starting point. We also used the sample JPSD estimator. As shown in Figures 6a and 6b, the attained results were consistent with the weather and traffic datasets.

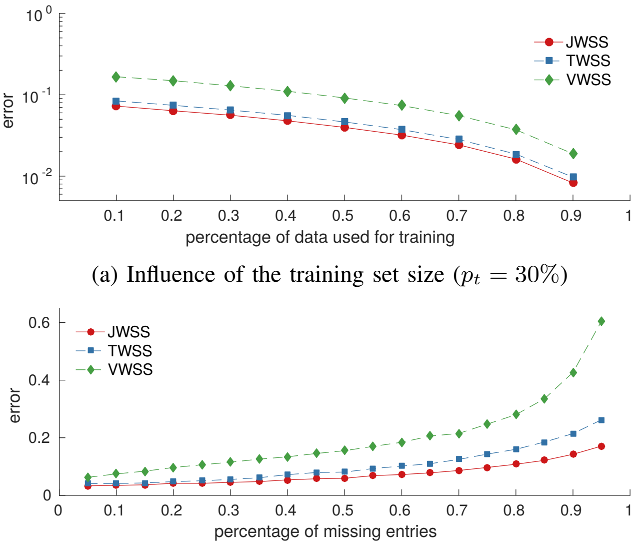
We remark that our simulations were done using the GSPBOX [47], the UNLocBoX [48], and the LTFAT [49]. The code reproducing our experiments is available at https://lts2.epfl.ch/stationary-time-vertex-signal-processing/.

VII. Conclusion

This paper proposed a novel definition of wide-sense stationarity appropriate for multivariate processes supported on the vertices of a graph. We showed that JWSS processes possess a number of familiar properties: they can be generated by filtering noise, and a joint Fourier transform diagonalizes their covariance. Furthermore, our model connects to time and vertex wide sense stationarity for multivariate processes.

Our model presents two key benefits. *First, the estimation and recovery of JWSS processes is very efficient, both in terms of sample and computational complexity.* In particular, the JPSD of a JWSS process can be estimated from very few (constant) number of samples at a complexity that is roughly

[6] The data correspond to the 3rd district of California and can be downloaded from http://pems.dot.ca.gov/



(a) Influence of the training set size ($p_t = 30\%$)

(b) Influence of the percentage of missing entries ($p_d = 90\%$)

Fig. 6: Experiments with the SIRS epidemic model.

linear to the number of graph edges and timesteps. After the PSD has been estimated, the linear MMSE recovery problems of interpolation, denoising, and forecasting can be solved in the same asymptotic complexity. *Second, joint stationarity is a volatile model, able to capture non-trivial statistical relations in the temporal and vertex domains.* Our experiments suggested that we can model real spatio-temporal processes as jointly stationary without significant loss. Specifically, the JWSS prior was found more expressive than (univariate) TWSS and VWSS priors, and improved upon the multivariate time stationarity prior when the dimensionality was large but the samples few.

Appendix

A. Deferred proofs

Proof of Theorem 1. By construction of the JFT basis, $\hat{\boldsymbol{X}}[0,0]$ captures the DC-offset of a signal, and condition (a) is equivalent to stating that $\mathbf{E}[\boldsymbol{x}] = c\mathbf{1}_{NT}$. Moreover, if the graph is connected and (a) holds, at least one of $\mathbf{E}\left[\hat{\boldsymbol{X}}[n_1,\tau_1]\right]$ and $\mathbf{E}\left[\hat{\boldsymbol{X}}[n_2,\tau_2]\right]$ must be zero when $n_1 \neq n_2$ or $\tau_1 \neq \tau_2$ and

$$\begin{aligned}
&\mathbf{E}\left[\hat{\boldsymbol{X}}[n_1,\tau_1]\hat{\boldsymbol{X}}[n_2,\tau_2]\right] \\
&= \mathbf{E}\left[\hat{\boldsymbol{X}}[n_1,\tau_1]\hat{\boldsymbol{X}}[n_2,\tau_2]\right] - \mathbf{E}\left[\hat{\boldsymbol{X}}[n_1,\tau_1]\right]\mathbf{E}\left[\hat{\boldsymbol{X}}[n_2,\tau_2]\right] \\
&= (\boldsymbol{U}_J^*\boldsymbol{\Sigma}\boldsymbol{U}_J)[(\tau_1-1)N+n_1,(\tau_2-1)N+n_2].
\end{aligned}$$

Therefore, condition (b) is equivalent to stating that $\boldsymbol{\Sigma} = \boldsymbol{U}_J\boldsymbol{D}\boldsymbol{U}_J^*$ for some diagonal matrix $\boldsymbol{D}$. In addition, (c) asserts that $\boldsymbol{D}[(\tau-1)N+n,(\tau-1)N+n] = h(\lambda_n,\omega_\tau)$ for every n,τ. Thus taken together, (b) and (c) state that $\boldsymbol{\Sigma} = \boldsymbol{U}_J\boldsymbol{D}\boldsymbol{U}_J^* = \boldsymbol{U}_J h(\boldsymbol{\Lambda}_G,\boldsymbol{\Lambda}_T)\boldsymbol{U}_J^* = h(\boldsymbol{L}_G,\boldsymbol{L}_T)$, which is the second moment condition of a JWSS process. □

Proof of Theorem 2. For the first movement, it is straightforward to see that $\mathbf{E}[\boldsymbol{X}[n,t]] = c$ if and only if both $\mathbf{E}[\boldsymbol{X}[n,t]] = c_t$ and $\mathbf{E}[\boldsymbol{X}[n,t]] = c_n\ \forall n,t$.
For the second moment, the covariance matrix of a JWSS process is by definition the linear operator associated to a joint filter $\boldsymbol{\Sigma} = h(\boldsymbol{L}_G,\boldsymbol{L}_T)$. Using (5), $\boldsymbol{\Sigma}_{t_1,t_2}$ can be written as

$$\boldsymbol{\Sigma}_{t_1,t_2} = \boldsymbol{U}_G\gamma_\delta(\boldsymbol{\Lambda})\boldsymbol{U}_G^* = \gamma_\delta(\boldsymbol{L}_G), \tag{19}$$

where $\delta = t_1 - t_2 + 1$ and

$$\gamma_\delta(\lambda) = \frac{1}{T}\sum_{\tau=1}^{T} h(\lambda,\omega_\tau)e^{j\omega_\tau\delta}. \tag{20}$$

Hence the process satisfies the (b) statement of Definition 2 (TWSS) and 3 (VWSS). Conversely if a process is TWSS and VWSS, we have $\boldsymbol{\Sigma}_{t_1,t_2} = \gamma_{t_1,t_2}(\boldsymbol{L}_G) = \gamma_\delta(\boldsymbol{L}_G)$ with $\delta = t_1 - t_2 + 1$. As a result, using (5), its covariance matrix can be written as a joint filter $h(\boldsymbol{L}_G,\boldsymbol{L}_T)$, where

$$h(\lambda_n,\omega_\tau) = \sum_{\delta=1}^{T}\gamma_\delta(\lambda_n)e^{j\omega_\tau\delta}, \tag{21}$$

and hence satisfies also the property of the second moment of JWSS processes. □

Proof of Property 2. The output of a filter $f(\boldsymbol{L}_J)$ can be written in vector form as $\boldsymbol{y} = f(\boldsymbol{L}_J)$. If the input signal $\boldsymbol{x}$ is JWSS, we can confirm that the first moment of the filter output is $\overline{f(\boldsymbol{L}_J)\boldsymbol{x}} = f(\boldsymbol{L}_J)\bar{\boldsymbol{x}} = f(0,0)\mathbf{E}[\boldsymbol{x}]$, which remain constant as $\mathbf{E}[\boldsymbol{x}]$ is constant by hypothesis. The computation of the second moment gives

$$\begin{aligned}
\boldsymbol{\Sigma}_{\boldsymbol{y}} &= \mathbf{E}\left[f(\boldsymbol{L}_J)\boldsymbol{x}\,(f(\boldsymbol{L}_J)\boldsymbol{x})^*\right] - \mathbf{E}[h(\boldsymbol{L}_J)\boldsymbol{x}]\,\mathbf{E}[(f(\boldsymbol{L}_J)\boldsymbol{x})^*] \\
&= f(\boldsymbol{L}_J)\mathbf{E}[\boldsymbol{x}\boldsymbol{x}^*]\,f(\boldsymbol{L}_J) - f(\boldsymbol{L}_J)\bar{\boldsymbol{x}}\bar{\boldsymbol{x}}^*f(\boldsymbol{L}_J)^* \\
&= f(\boldsymbol{L}_J)\boldsymbol{\Sigma}_{\boldsymbol{x}}f(\boldsymbol{L}_J)^* \\
&= \boldsymbol{U}_J\,\left(f^2(\theta)\,h_{\boldsymbol{X}}(\theta)\right)\,\boldsymbol{U}_J^*,
\end{aligned}$$

which satisfies the second moment condition of JWSS process. □

Lemma 1. *If function $h(\theta)$ is ϵ-Lipschitz, then the bias is bounded by*

$$\left|\mathbf{E}\left[\ddot{h}(\theta) - h(\theta)\right]\right| \leq \frac{\epsilon}{c_g(\theta)}\sum_{n=1,\tau=1}^{T,N} g(\theta-\theta_{n,\tau})^2\left\|\theta-\theta_{n,\tau}\right\|_2.$$

Proof. Since $h(\theta)$ is ϵ Lipschitz, we have $|h(\theta) - h(\theta_{n,\tau})| \leq \epsilon\left\|\theta-\theta_{n,\tau}\right\|_2$. Hence, we write

$$\begin{aligned}
\left|\mathbf{E}\left[\ddot{h}(\theta) - h(\theta)\right]\right| &= \left|\sum_{n,\tau=1}^{NT} g(\theta-\theta_{n,\tau})^2\frac{h(\theta_{n,\tau})}{c_g(\theta)} - h(\theta)\right| \\
&\leq |A\,h(\theta)| + \frac{\epsilon}{c_g(\theta)}\sum_{n,\tau=1}^{NT} g(\theta-\theta_{n,\tau})^2\left\|\theta-\theta_{n,\tau}\right\|_2
\end{aligned}$$

where by definition $A = \sum_{n,\tau=1}^{N,T}\frac{g^2(\theta-\theta_{n,\tau})}{c_g(\theta)} - 1 = 0$, and the claim follows. □

Lemma 2. *If $\boldsymbol{X}$ is a JWSS process such that the entries of $\hat{\boldsymbol{X}}$ are independent random variables, the convolutional JPSD* *estimate at θ has variance*

$$\mathbf{Var}\left[\ddot{h}(\theta)\right] = \sum_{n,\tau} \frac{g(\theta-\theta_{n,\tau})^4}{c_g(\theta)^2}\,\mathbf{Var}\left[\dot{h}(\theta_{n,\tau})\right], \tag{22}$$

where $\mathbf{Var}\left[\dot{h}(\theta_{n,\tau})\right]$ *is the variance of the sample JPSD estimator at* $\theta_{n,\tau}$*.*

Proof. Set $\alpha_{n,\tau} = g(\theta-\theta_{n,\tau})^2 h(\theta_{n,\tau})/c_g(\theta)$ and $\hat{\boldsymbol{E}}_{(k)} = \mathrm{mat}\big(\hat{\varepsilon}_{(k)}\big) = \mathrm{mat}\big(h(\boldsymbol{\Lambda}_G,\boldsymbol{\Omega})^{+}\hat{\boldsymbol{x}}_{(k)}\big)$, where $+$ denotes the pseudo-inverse and $\mathrm{mat}(\cdot)$ is the matricization operator. The centered random variable

$$\ddot{h}(\theta) - \mathbf{E}\left[\ddot{h}(\theta)\right] = \sum_{n,\tau} \frac{g(\theta-\theta_{n,\tau})^2}{c_g(\theta)}\big(\dot{h}(\theta_{n,\tau}) - h(\theta_{n,\tau})\big)$$

$$= \sum_{n,\tau}\alpha_{n,\tau}\left(\sum_{k}\frac{\hat{\boldsymbol{E}}_{(k)}[n,\tau]\hat{\boldsymbol{E}}_{(k)}[n,\tau]^*}{K} - 1\right) = \sum_{n,\tau}\alpha_{n,\tau}\, z_{n,\tau}$$

is a weighted sum of centered, identically distributed random variables $z_{n,\tau}$. Moreover, when the elements of $\hat{\boldsymbol{E}}_{(k)}$ are independent, so are the variables $z_{n,\tau}$. It follows that,

$$\begin{aligned}\mathbf{Var}\left[\ddot{h}(\theta)\right] &= \sum_{n,\tau}\alpha_{n,\tau}^2\,\mathbf{Var}\left[z_{n,\tau}^2\right] \\ &= \sum_{n,\tau}\frac{g(\theta-\theta_{n,\tau})^4}{c_g(\theta)^2}\,\mathbf{Var}\left[\dot{h}(\theta_{n,\tau})\right],\end{aligned}$$

which matches our claim. □

References

[1] M. Rudelson, "Random vectors in the isotropic position," *Journal of Functional Analysis*, vol. 164, no. 1, pp. 60–72, 1999.

[2] M. J. Keeling and K. T. Eames, "Networks and epidemic models," *Journal of the Royal Society Interface*, vol. 2, no. 4, pp. 295–307, 2005.

[3] P. Mohan, V. N. Padmanabhan, and R. Ramjee, "Nericell: rich monitoring of road and traffic conditions using mobile smartphones," in *Proceedings of the 6th ACM conference on Embedded network sensor systems*. ACM, 2008, pp. 323–336.

[4] W. Huang, L. Goldsberry, N. F. Wymbs, S. T. Grafton, D. S. Bassett, and A. Ribeiro, "Graph frequency analysis of brain signals," *arXiv preprint arXiv:1512.00037*, 2015.

[5] F. Zhang and E. R. Hancock, "Graph spectral image smoothing using the heat kernel," *Pattern Recognition*, vol. 41, no. 11, pp. 3328–3342, 2008.

[6] A. J. Smola and R. Kondor, "Kernels and regularization on graphs," in *Learning theory and kernel machines*. Springer, 2003, pp. 144–158.

[7] M. Belkin and P. Niyogi, "Semi-supervised learning on riemannian manifolds," *Machine learning*, vol. 56, no. 1-3, pp. 209–239, 2004.

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

[9] A. Sandryhaila and J. M. Moura, "Discrete signal processing on graphs," *IEEE transactions on signal processing*, vol. 61, pp. 1644–1656, 2013.

[10] A. Gadde and A. Ortega, "A probabilistic interpretation of sampling theory of graph signals," *arXiv preprint arXiv:1503.06629*, 2015.

[11] C. Zhang, D. Florêncio, and P. A. Chou, "Graph signal processing–a probabilistic framework," *Microsoft Res., Redmond, WA, USA, Tech. Rep. MSR-TR-2015-31*, 2015.

[12] E. Isufi, A. Loukas, A. Simonetto, and G. Leus, "Autoregressive moving average graph filtering," *IEEE Transactions on Signal Processing*, vol. 65, no. 2, pp. 274–288, 2017.

[13] A. Loukas and D. Foucard, "Frequency analysis of temporal graph signals," *arXiv preprint arXiv:1602.04434*, 2016.

[14] N. Perraudin and P. Vandergheynst, "Stationary signal processing on graphs," *IEEE Transactions on Signal Processing*, vol. 65, no. 13, pp. 3462 – 3477, 2017.

[15] B. Girault, "Stationary graph signals using an isometric graph translation," in *Signal Processing Conference (EUSIPCO), 2015 23rd European*. IEEE, 2015, pp. 1516–1520.

[16] A. G. Marques, S. Segarra, G. Leus, and A. Ribeiro, "Stationary graph processes and spectral estimation," *arXiv preprint arXiv:1603.04667*, 2016.

[17] N. Wiener and P. Masani, "The prediction theory of multivariate stochastic processes," *Acta Mathematica*, vol. 98, no. 1, pp. 111–150, 1957.

[18] ——, "The prediction theory of multivariate stochastic processes, ii," *Acta Mathematica*, vol. 99, no. 1, pp. 93–137, 1958.

[19] P. Bloomfield, *Fourier analysis of time series: an introduction*. John Wiley & Sons, 2004.

[20] F. R. Bach and M. I. Jordan, "Learning graphical models for stationary time series," *IEEE transactions on signal processing*, vol. 52, no. 8, pp. 2189–2199, 2004.

[21] R. Dahlhaus and M. Eichler, "Causality and graphical models in time series analysis," *Oxford Statistical Science Series*, pp. 115–137, 2003.

[22] B. Girault, "Signal processing on graphs-contributions to an emerging field," Ph.D. dissertation, Ecole normale supérieure de lyon, 2015.

[23] S. P. Chepuri and G. Leus, "Subsampling for graph power spectrum estimation," *arXiv preprint arXiv:1603.03697*, 2016.

[24] A. Loukas and N. Perraudin, "Predicting the evolution of stationary graph signals," *arXiv preprint arXiv:1607.03313*, 2016.

[25] J. Mei and J. M. Moura, "Signal processing on graphs: Causal modeling of unstructured data," *arXiv preprint arXiv:1503.00173*, 2015.

[26] N. Perraudin, A. Loukas, F. Grassi, and P. Vandergheynst, "Towards stationary time-vertex signal processing," *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017.

[27] F. Grassi, A. Loukas, N. Perraudin, and B. Ricaud, "A time-vertex signal processing framework," *ArXiv e-prints*, 2016.

[28] A. Loukas, A. Simonetto, and G. Leus, "Distributed autoregressive moving average graph filters," *IEEE Signal Processing Letters*, vol. 22, no. 11, pp. 1931–1935, 2015.

[29] E. Isufi, A. Loukas, A. Simonetto, and G. Leus, "Separable autoregressive moving average graph-temporal filters," in *Signal Processing Conference (EUSIPCO), 2016 24th European*. IEEE, 2016, pp. 200–204.

[30] D. I. Shuman, B. Ricaud, and P. Vandergheynst, "Vertex-frequency analysis on graphs," *Applied and Computational Harmonic Analysis*, vol. 40, no. 2, pp. 260–291, 2016.

[31] R. Vershynin, "How close is the sample covariance matrix to the actual covariance matrix?" *Journal of Theoretical Probability*, vol. 25, no. 3, pp. 655–686, 2012.

[32] M. S. Bartlett, "Periodogram analysis and continuous spectra," *Biometrika*, pp. 1–16, 1950.

[33] P. Welch, "The use of fast fourier transform for the estimation of power spectra: a method based on time averaging over short, modified periodograms," *IEEE Transactions on audio and electroacoustics*, pp. 70–73, 1967.

[34] K. Gröchenig, *Foundations of time-frequency analysis*. Springer Science & Business Media, 2013.

[35] H. G. Feichtinger and T. Strohmer, *Gabor analysis and algorithms: Theory and applications*. Springer Science & Business Media, 2012.

[36] C. De Boor, C. De Boor, E.-U. Mathématicien, C. De Boor, and C. De Boor, *A practical guide to splines*. Springer-Verlag New York, 1978, vol. 27.

[37] A. Susnjara, N. Perraudin, D. Kressner, and P. Vandergheynst, "Accelerated filtering on graphs using lanczos method," *arXiv preprint arXiv:1509.04537*, 2015.

[38] O. Axelsson, "Conjugate gradient type methods for unsymmetric and inconsistent systems of linear equations," *Linear algebra and its applications*, vol. 29, pp. 1–16, 1980.

[39] P. L. Combettes and V. R. Wajs, "Signal recovery by proximal forward-backward splitting," *Multiscale Modeling & Simulation*, vol. 4, no. 4, pp. 1168–1200, 2005.

[40] P. L. Combettes and J.-C. Pesquet, "Proximal splitting methods in signal processing," in *Fixed-point algorithms for inverse problems in science and engineering*. Springer, 2011, pp. 185–212.

[41] N. Komodakis and J.-C. Pesquet, "Playing with duality: An overview of recent primal? dual approaches for solving large-scale optimization problems," *IEEE Signal Processing Magazine*, vol. 32, no. 6, pp. 31–54, 2015.

[42] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM journal on imaging sciences*, vol. 2, no. 1, pp. 183–202, 2009.

[43] P. L. Combettes and J.-C. Pesquet, "A douglas–rachford splitting approach to nonsmooth convex variational signal recovery," *IEEE Journal of Selected Topics in Signal Processing*, vol. 1, no. 4, pp. 564–574, 2007.

[44] N. Wiener, *Extrapolation, interpolation, and smoothing of stationary time series*. MIT press Cambridge, MA, 1949, vol. 2.

[45] B. Girault, P. Goncalves, E. Fleury, and A. S. Mor, "Semi-supervised learning for graph to signal mapping: A graph signal wiener filter interpretation," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*. IEEE, 2014, pp. 1115–1119.

[46] H. Akaike, "A new look at the statistical model identification," *IEEE transactions on automatic control*, vol. 19, no. 6, pp. 716–723, 1974.

[47] N. Perraudin, J. Paratte, D. Shuman, V. Kalofolias, P. Vandergheynst, and D. K. Hammond, "GSPBOX: A toolbox for signal processing on graphs," *ArXiv e-prints*, Aug. 2014.

[48] N. Perraudin, D. Shuman, G. Puy, and P. Vandergheynst, "UNLocBoX A matlab convex optimization toolbox using proximal splitting methods," *ArXiv e-prints*, Feb. 2014.

[49] Z. Prusa, P. L. Sondergaard, N. Holighaus, C. Wiesmeyr, and P. Balazs, "The Large Time-Frequency Analysis Toolbox 2.0," in *Sound, Music, and Motion*, ser. Lecture Notes in Computer Science. Springer International Publishing, 2014, pp. 419–442.