

Towards stationary time-vertex signal processing

Nathanaël Perraudin* and Andreas Loukas* and Francesco Grassi and Pierre Vandergheynst
 École Polytechnique Fédérale Lausanne, Switzerland

Abstract—Graph-based methods for signal processing have shown promise for the analysis of data exhibiting irregular structure, such as those found in social, transportation, and sensor networks. Yet, though these systems are often dynamic, state-of-the-art methods for signal processing on graphs ignore the dimension of time, treating successive graph signals independently or taking a global average. To address this shortcoming, this paper considers the statistical analysis of time-varying graph signals. We introduce a novel definition of *joint (time-vertex) stationarity*, which generalizes the classical definition of time stationarity and the more recent definition appropriate for graphs. Joint stationarity gives rise to a scalable Wiener optimization framework for joint denoising, semi-supervised learning, or more generally inverting a linear operator, that is provably optimal. Experimental results on real weather data demonstrate that taking into account graph and time dimensions jointly can yield significant accuracy improvements in the reconstruction effort.

I. INTRODUCTION

Whether examining opinion dichotomy in social networks [1], how traffic evolves in the roads of a city [2], or neuronal activation patterns present in the brain [3], much of the high-dimensional data one encounters exhibit complex non-euclidean properties. This realization has been the driving force behind recent efforts to re-invent the mathematical models used for data analysis. Within the field of signal processing, one of the main research thrusts has been to extend harmonic analysis to graph signals, i.e., signals supported on the vertices of irregular graphs. The key breakthrough in the field has been the introduction of a notion of frequency appropriate for graph signals and of the associated graph Fourier transform (GFT). Because it enables us to process signals taking into account complex relations between variables, the GFT has led to advances in problems such as denoising [4] and semi-supervised learning [5], [6].

Yet, state-of-the-art graph frequency based methods often fail to produce useful results when applied to real datasets. One of the main reasons underlying this shortcoming is that they ignore the time dimension, for example by treating successive signals independently or performing a global average [3], [7], [8]. On the contrary, many of the systems to which graph signal processing is applied to are dynamic. Consider for instance a sensor network, and suppose that we want to infer the weather conditions on a mountain given temperature measurements from a small set of weather stations. Approaches that do not take into account the temporal evolution of weather will be biased by seasonal variations and unable to provide insights about transient phenomena. Moreover, when the weather dynamics are slow and predictable, taking into account the time dimension, e.g., by imposing a smoothness

prior, can yield accuracy improvements in the reconstruction effort.

Motivated by this need, this paper considers the *statistical analysis of time-evolving graph signals*. 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 [9], and the recent generalization of stationarity for graphs [7], [10], [11]. 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, as it elegantly generalizes existing definitions of stationarity in the time and vertex domains. We show that joint stationarity carries along important properties classically associated with stationarity. Moreover, our definition leads to a Wiener framework for solving denoising and interpolating time-varying graph signals, that yields superior performance compared to state-of-the-art methods in time or vertex domains. The proposed framework is composed out of two key components: a scalable joint power spectral density estimation method, and an optimization framework suitable for deconvolution under additive error. The latter is shown to be optimal in the mean-squared error sense. Experiments with a real weather dataset illustrate the superior performance of our method, ultimately demonstrating that joint stationarity is a useful assumption in practice.

II. PRELIMINARIES

Our objective is to model and predict the evolution of graph signals, i.e., signals supported on the vertices $\mathcal{V} = \{v_1, v_2, \dots, v_N\}$ of a weighted undirected graph $G = (\mathcal{V}, \mathcal{E}, \mathbf{W}_G)$, with \mathcal{E} the set of edges and \mathbf{W}_G the weighted adjacency matrix. A more convenient matrix representation of G is the (combinatorial¹) Laplacian matrix $\mathbf{L}_G = \text{diag}(\mathbf{W}_G \mathbf{1}_N) - \mathbf{W}_G$, where $\mathbf{1}_N$ is the all-ones vector of size N , and $\text{diag}(\mathbf{W}_G \mathbf{1}_N)$ is the diagonal degree matrix.

Harmonic vertex analysis. In the context of graph signal processing, the importance of the Laplacian matrix stems from it giving rise to a graph-specific notion of frequency. The *Graph Fourier Transform* (GFT) of a graph signal $\mathbf{x} \in \mathbb{R}^N$ is defined as $\text{GFT}\{\mathbf{x}\} = \mathbf{U}_G^* \mathbf{x}$, where \mathbf{U}_G is the eigenvector matrix of \mathbf{L}_G and thus $\mathbf{L}_G = \mathbf{U}_G \mathbf{\Lambda}_G \mathbf{U}_G^*$. The GFT allows us to extend filtering to graphs [13], [14], [15]. Filtering a signal \mathbf{x} with a graph filter $h(\mathbf{L}_G)$ corresponds to element-wise multiplication in the spectral domain

$$h(\mathbf{L}_G) \mathbf{x} \triangleq \text{GFT}^{-1}\{h(\mathbf{\Lambda}_G) \circ \text{GFT}\{\mathbf{x}\}\} = \mathbf{U}_G h(\mathbf{\Lambda}_G) \mathbf{U}_G^* \mathbf{x},$$

¹Though we use the combinatorial Laplacian in our presentation, our results are applicable to any positive semi-definite matrix representation of a graph or to the recently introduced shift operator [12].

where the scalar function $h : \mathbb{R}_+ \mapsto \mathbb{R}$, referred to as the graph frequency response, has been applied to each diagonal entry of Λ_G . It is often convenient to represent the diagonal of matrix $h(\Lambda_G)$ as a vector, in which case we write $\mathbf{h} = \text{diag}(h(\Lambda_G))$. The notation U_G^* denotes the transposed complex conjugate of U_G , U_G^T the transpose of U_G , and \bar{U}_G the complex conjugate of U_G . We will also use the notion of *graph localization* [15], [7], a generalization of the translation operator used in the classical setting appropriate for graphs². Localizing a filter with frequency response h onto vertex v_i reads

$$\mathcal{T}_i^G h \triangleq h(\mathbf{L}_G) \delta_i, \quad (1)$$

where δ_i is a Kronecker delta centered at vertex v_i . For a sufficiently regular function h , this operation localizes the filter around v_i [15, Theorem 1 and Corollary 2]. Evaluated at the i_2 -th vertex, the above expression becomes

$$\mathcal{T}_{i_1}^G h(i_2) = \sum_{n=1}^N h(\lambda_n) \bar{\mathbf{u}}_n(i_1) \mathbf{u}_n(i_2), \quad (2)$$

where we use the notation $\mathbf{u}_n(i) = [U_G]_{i,n}$ and $\lambda_n = [\Lambda_G]_{n,n}$. Note that in the expression above, the localization operator takes precedence over indexing and $\mathcal{T}_{i_1}^G h(i_2) = [\mathcal{T}_{i_1}^G h](i_2)$; this convention is used throughout this paper. The concept of localization is intimately linked to that of *translation* in the time domain. If $\mathcal{T}_\tau^T h$ is the localization operator taken on a cycle graph of T vertices (representing time), localization is equivalent to translation

$$\mathcal{T}_\tau^T h(t) = \mathcal{T}_0^T h(t - \tau), \quad \text{for all } t, \tau = 1, \dots, T. \quad (3)$$

In simple words, for a cyclic graph, the localization operator computes the inverse Fourier transform of the frequency response h , and translates it to vertex v_i . We can verify this using the fact that the complex exponential Fourier basis form the eigenvector set of all cyclic graphs [16], which together with (2) implies that

$$\begin{aligned} \mathcal{T}_{t_1}^T h(t_2) &= \frac{1}{T} \sum_{\tau=1}^T h(\omega_\tau) e^{-2\pi j \frac{(\tau-1)t_1}{T}} e^{2\pi j \frac{(\tau-1)t_2}{T}} \\ &= \frac{1}{T} \sum_{\tau=1}^T h(\omega_\tau) e^{2\pi j \frac{(\tau-1)(t_2-t_1)}{T}} = \mathcal{T}_0^T h(t_2 - t_1). \end{aligned} \quad (4)$$

Above $\mathcal{T}_0^T h = U_T \mathbf{h}$ is the inverse Fourier transform of \mathbf{h} and U_T is the orthonormal Fourier basis. In the case of irregular graphs, localization differs further from translation because the shape of the localized filter adapts to the graph and varies as a function of its topology. Additional insights about the localization operator can be found in [15], [13], [17], [7].

Harmonic time-vertex analysis. Suppose that a graph signal \mathbf{x}_t is sampled at T successive regular intervals of unit length. The time-varying graph signal $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_T] \in \mathbb{R}^{N \times T}$ is then the matrix having graph signal \mathbf{x}_t as its t -th column. Equivalently, $\mathbf{X} = [\mathbf{x}^1, \mathbf{x}^2, \dots, \mathbf{x}^N]^T$ holds N

²Stationarity is classically defined as the invariance of statistical moments of a signal with respect to translation. This definition however cannot be directly generalized to graphs, which do not possess regular structure and thus lack of an isometric translation operator.

temporal signals $\mathbf{x}^i \in \mathbb{R}^T$, one for each vertex v_i . Throughout this paper, we denote as $\mathbf{x} = \text{vec}(\mathbf{X})$ (without subscript) the vectorized representation of the matrix \mathbf{X} .

The frequency representation of \mathbf{X} is given by the joint (time-vertex) Fourier transform (or JFT for short)

$$\text{JFT}\{\mathbf{X}\} = U_G^* \mathbf{X} \bar{U}_T, \quad (5)$$

where, once more, U_G is the graph Laplacian eigenvector matrix, whereas \bar{U}_T is the complex conjugate of the DFT matrix divided by $1/\sqrt{T}$. In fact, matrix U_T is the eigenvector matrix of the lag operator (or Laplacian matrix) $\mathbf{L}_T = U_T \Lambda_T U_T^*$. Denote by Ω the diagonal matrix of angular frequencies (i.e., $\Omega_{tt} = \omega_t = 2\pi t/T$). In case \mathbf{L}_T is the lag operator, we have $\Lambda_T = e^{-j\Omega}$, where $j = \sqrt{-1}$. When \mathbf{L}_T is the Laplacian, we have $\Lambda_T = \text{real}(\mathbf{I} - e^{-j\Omega})$. Expressed in vector form, the joint Fourier transform becomes $\text{JFT}\{\mathbf{x}\} = U_J^* \mathbf{x}$, where $U_J = U_T \otimes U_G$ is unitary, and operator (\otimes) denotes the kroneker product. The inverse joint Fourier transforms in matrix and vector form are, respectively, $\text{JFT}^{-1}\{\mathbf{X}\} = U_G \mathbf{X} U_T^T$ and $\text{JFT}^{-1}\{\mathbf{x}\} = U_J \mathbf{x}$. For an in-depth discussion of JFT and its properties, we refer the reader to [9].

Leveraging the definition of the JFT, filtering and localization can also be extended to the joint (time-vertex) domain. A *joint filter* $h(\mathbf{L}_J)$ is a function defined in the joint spectral domain $h : \mathbb{R}_+ \times \mathbb{R} \mapsto \mathbb{R}$ that is evaluated at the graph eigenvalues λ_G and the angular frequencies ω . The output of a joint filter is

$$h(\mathbf{L}_J) \mathbf{x} \triangleq U_J h(\Lambda_G, \Omega) U_J^* \mathbf{x}, \quad (6)$$

where $h(\Lambda_G, \Omega)$ is a $NT \times NT$ diagonal matrix with $[h(\Lambda_G, \Omega)]_{k,k} = h(\lambda_n, \omega_\tau)$ and $k = N(\tau - 1) + n$. Equivalently, if we define the matrix \mathbf{H} of dimension $N \times T$ as $\mathbf{H}_{n,\tau} = h(\lambda_n, \omega_\tau)$ for every graph frequency λ_n and temporal frequency ω_τ , we have

$$h(\mathbf{L}_J) \mathbf{x} = \text{vec}(\text{JFT}^{-1}\{\mathbf{H} \circ \text{JFT}\{\mathbf{X}\}\}), \quad (7)$$

with (\circ) being the element-wise multiplication (Hadamard product). In an analogy to (1), we define the *joint localization operator* as

$$\mathcal{T}_{i,t}^J h \triangleq \text{mat}(h(\mathbf{L}_J) (\delta_t \otimes \delta_i)) \quad (8)$$

$$= \text{JFT}^{-1}\{\mathbf{H} \circ \text{JFT}\{\delta_i \delta_t^T\}\} \quad (9)$$

where $\text{mat}(\cdot)$ is the matricization operator, such that $\text{mat}(\text{vec}(\mathbf{X})) = \mathbf{X}$. In order to link (8) with graph localization (1) and the classical translation operator, we observe the following relations

$$\begin{aligned} \mathcal{T}_{i_1, t_1}^J h(i_2, t_2) &= \frac{1}{T} \sum_{\tau=1}^{N, T} h(\lambda_n, \omega_k) \bar{\mathbf{u}}_n(i_1) \mathbf{u}_n(i_2) e^{2\pi j \frac{(\tau-1)(t_2-t_1)}{T}} \\ &= \mathcal{T}_{i_1, 0}^J h(i_2, t_2 - t_1) \end{aligned} \quad (10)$$

$$\begin{aligned} &= \frac{1}{T} \sum_{n=1}^N \left[\sum_{\tau=1}^T h(\lambda_n, \omega_\tau) e^{2\pi j \frac{(\tau-1)(t_2-t_1)}{T}} \right] \bar{\mathbf{u}}_n(i_1) \mathbf{u}_n(i_2) \\ &= \sum_{n=1}^N [\mathcal{T}_{t_1}^T \mathbf{H}_{n,\cdot}] (t_2) \bar{\mathbf{u}}_n(i_1) \mathbf{u}_n(i_2) \end{aligned} \quad (11)$$

$$\begin{aligned}
&= \frac{1}{T} \sum_{\tau=1}^T \left[\sum_{n=1}^N h(\lambda_n, \omega_\tau) \bar{\mathbf{u}}_n(i_1) \mathbf{u}_n(i_2) \right] e^{2\pi j \frac{(\tau-1)(t_2-t_1)}{T}} \\
&= \frac{1}{T} \sum_{\tau=1}^T [\mathcal{T}_{i_1}^G \mathbf{H}_{\cdot, \tau}] (i_2) e^{2\pi j \frac{(\tau-1)(t_2-t_1)}{T}}. \quad (12)
\end{aligned}$$

The above equations provide three key insights about the joint localization operator:

- 1) From (10), we observe that the localization operator performs a translation along the time dimension.
- 2) From (11), it follows that joint localization consist of first localizing (translating) independently in time each line of the matrix \mathbf{H} and then localizing independently on the graph each column of the resulting matrix. Joint localization is thus equivalent to a successive application of a graph and a time localization operator.
- 3) Furthermore, according to (12), the successive localization in time and graph can be performed in any order.

When the filter is separable, i.e., when the joint frequency response can be written as the product of a frequency response defined solely in the vertex domain and one in the time domain $h(\lambda, \omega) = h_1(\lambda)h_2(\omega)$, the joint localization is simply

$$h(\mathbf{L}_J) (\boldsymbol{\delta}_t \otimes \boldsymbol{\delta}_i) = \text{vec}(h(\mathbf{L}_G)(\boldsymbol{\delta}_t \otimes \boldsymbol{\delta}_i^T)h(\mathbf{L}_T)). \quad (13)$$

Nevertheless, for this work, we assume that the filter is not separable as it is a too restrictive hypothesis.

III. JOINT TIME-VERTEX STATIONARITY

Let \mathbf{X} be a discrete multivariate stochastic process with finite number of time-steps T that is indexed by vertex v_i and time t . We refer to such processes as *joint time-vertex processes*, or *joint processes* for short.

To put our results in context, let us first review the established definitions of stationarity over time and vertex domains, respectively. Our definition will emerge us a consequence of both. We note that, although our exposition is self-contained, the reader will benefit from familiarizing with previous work on stationarity on graphs [7].

Definition 1 (Time stationarity). *A joint process \mathbf{X} is Time Wide-Sense Stationary (TWSS), if and only if the following two properties hold independently for each vertex v_i :*

- 1) *The expected value is constant over the time domain*

$$\mathbb{E}[\mathbf{x}^i] = c_i \mathbf{1}_T.$$

- 2) *There exists a function γ_i , for which*

$$[\boldsymbol{\Sigma}_{\mathbf{x}^i}]_{t, \tau} = \left[\mathbb{E}[\mathbf{x}^i \mathbf{x}^{i*}] - \mathbb{E}[\mathbf{x}^i] \mathbb{E}[\mathbf{x}^{i*}] \right]_{t, \tau} = \mathcal{T}_t^T \gamma_i$$

Function γ_i is the autocorrelation function of signal \mathbf{x}^i in the Fourier domain, and is also referred to as Time Power Spectral Density (TPSD).

We remind the reader that on a cyclic graph, localizing the TPSD is equivalent to translating the autocorrelation. Thus, using (9) we recover the classical definition, where the autocorrelation function depends only on the time difference: $[\boldsymbol{\Sigma}_{\mathbf{x}^i}]_{t, \tau} = \mathcal{T}_0^T \gamma_i(t - \tau)$. Simply put, assuming time stationarity is equivalent to asserting that the statistics of the two first

moments are independent of the time. We also observe that the TPSD is the Fourier transform of the autocorrelation, agreeing with the Wiener-Khintchine Theorem [18]. To summarize, TPSD encodes the statistics of the signal in the spectral domain.

This consideration allows us to generalize the concept of stationarity to graph signals. Please refer to the work of Perraudin and Vandergheynst [7] for a more detailed study. We express a variation of their definition in the following.

Definition 2 (Vertex stationarity). *A joint process $\mathbf{X} = [\mathbf{x}_1 \mathbf{x}_2 \dots \mathbf{x}_T]$ is called Vertex Wide-Sense (or second order) Stationary (VWSS), if and only if the following two properties hold independently for each time t :*

- 1) *The expected value is in the null space of the Laplacian*

$$\mathbf{L}_G \mathbb{E}[\mathbf{x}_t] = \mathbf{0}_N.$$

- 2) *There exists a graph filter $s_t(\mathbf{L}_G)$, for which*

$$[\boldsymbol{\Sigma}_{\mathbf{x}_t}]_{i, \cdot} = [\mathbb{E}[\mathbf{x}_t \mathbf{x}_t^*] - \mathbb{E}[\mathbf{x}_t] \mathbb{E}[\mathbf{x}_t^*]]_{i, \cdot} = \mathcal{T}_i^G s_t.$$

Function s_t is the autocorrelation function of signal \mathbf{x}_t in the graph Fourier domain and is also referred to as Vertex Power Spectral Density (VPSD).

Considering that the null space of \mathbf{L}_T in both the normalized and the combinatorial case is the span of the constant eigenvector $\mathbf{1}_T$, the first condition of the above definition is analogous to the corresponding condition of the time stationarity definition. Moreover, the condition for the second moment is a natural generalization of the second condition of time stationarity where, instead of imposing translation invariance, we suppose invariance under the localization operator. This second condition is in fact equivalent to a generalization of the Wiener-Khintchine theorem and implies that $\boldsymbol{\Sigma}_{\mathbf{x}_t}$ is jointly diagonalizable with \mathbf{L}_G (in TWSS the covariance is Toeplitz and thus also diagonalizable with the DFT matrix \mathbf{U}_T).

We now unify the TWSS and GWSS in order to leverage both the time and vertex domain statistics.

Definition 3 (Joint stationarity). *A process \mathbf{X} is called Jointly (or time-vertex) Wide-Sense Stationary (JWSS), if and only if its vector form $\mathbf{x} = \text{vec}(\mathbf{X})$ satisfies the following properties:*

- 1) *The expected value is in the null space of the Laplacian*

$$\mathbf{L}_J \mathbb{E}[\mathbf{x}] = \mathbf{0}_{NT}.$$

- 2) *There exists a joint filter $h(\mathbf{L}_J)$, for which*

$$[\boldsymbol{\Sigma}_{\mathbf{x}}]_{k, \cdot} = [\mathbb{E}[\mathbf{x} \mathbf{x}^*] - \mathbb{E}[\mathbf{x}] \mathbb{E}[\mathbf{x}^*]]_{k, \cdot} = \text{vec}(\mathcal{T}_{i, t}^J h),$$

where $k = N(t - 1) + i$.

Function h is the autocorrelation function of signal \mathbf{x} in the joint Fourier domain and is also referred to as time-vertex power spectral density or Joint Power Spectral Density (JPSD) for short.

The definition above is in fact equivalent to stating that the mean is constant, and the covariance matrix $\boldsymbol{\Sigma}_{\mathbf{x}}$ is jointly diagonalizable with the joint Laplacian \mathbf{L}_J . The latter statement is (also) a generalization the Wiener-Khintchine theorem and is proven next.

Theorem 1. A process \mathbf{X} is JWSS if and only if 1) $\mathbf{L}_J \mathbb{E}[\mathbf{x}] = \mathbf{0}_{NT}$, and 2) its covariance matrix is jointly diagonalizable by the joint Fourier basis \mathbf{U}_J .

Proof. To prove an equivalence relation between the two definitions (i.e., the JWSS definition and the one stated by the theorem) we will prove a one-to-one equivalence between their respective conditions. Clearly the first conditions of both definitions are identical. The second condition $[\boldsymbol{\Sigma}_{\mathbf{x}}]_{k,\cdot} = \text{vec}(\mathcal{T}_{i,t}^J h)$ of the joint stationarity definition together with (8) assert that the covariance being a joint filter $\boldsymbol{\Sigma}_{\mathbf{x}} = h(\mathbf{L}_J)$ for some function h . We therefore have that $\boldsymbol{\Sigma}_{\mathbf{x}} = \mathbf{U}_J h(\boldsymbol{\Lambda}_G, \boldsymbol{\Omega}) \mathbf{U}_J^*$ with $h(\boldsymbol{\Lambda}_G, \boldsymbol{\Omega})$ diagonal, which implies our claim. \square

Interestingly, assuming joint stationarity is equivalent to assuming stationarity in both domains at the same time.

Theorem 2. If a joint process \mathbf{X} is JWSS, then it is both TWSS and GWSS.

Proof. It is straightforward to see that $\mathbf{L}_J \mathbb{E}[\mathbf{x}] = \mathbf{0}_{NT}$ if and only if both $\mathbf{L}_T \mathbb{E}[\mathbf{x}_t] = \mathbf{0}_N$ and $\mathbf{L}_G \mathbb{E}[\mathbf{x}^i] = \mathbf{0}_T$, hold for all t and i . We still need to show that the second-order moment properties of TWSS and VWSS are equivalent to that of JWSS. If a process is joint stationary, then from (12) we have that, for each vertex v_i

$$\begin{aligned} [\boldsymbol{\Sigma}_{\mathbf{x}^i}]_{t_1, t_2} &= [\mathcal{T}_{i, t_1}^J h](i, t_2) \\ &= \frac{1}{T} \sum_{\tau=1}^T [\mathcal{T}_i^G \mathbf{H}_{\cdot, \tau}](i) e^{2\pi j \frac{(\tau-1)(t_2-t_1)}{T}} \end{aligned}$$

which is equivalent to asserting that \mathbf{x}^i is stationary in time with TPSD $\gamma_i(\omega_\tau) = [\mathcal{T}_i^G \mathbf{H}_{\cdot, \tau}](i)$. Similarly, using (11), we find that for each time t

$$[\boldsymbol{\Sigma}_{\mathbf{x}_t}]_{i_1, i_2} = [\mathcal{T}_{i_1, t}^J h](i_2, t) = \sum_{n=1}^N [\mathcal{T}_t^T \mathbf{H}_{n, \cdot}](t) \bar{\mathbf{u}}_n(i_1) \mathbf{u}_n(i_2)$$

meaning that process \mathbf{x}_t is stationary with VPSD $s_t(\lambda_n) = [\mathcal{T}_t^G \mathbf{H}_{n, \cdot}](t)$. \square

Example 1 (White *i.i.d.* noise). White *i.i.d.* noise $\mathbf{w} \in \mathbb{R}^{NT}$ is JWSS for any graph. Indeed, the first moment $\mathbb{E}[\mathbf{w}]$ is constant for any time and vertex. Moreover, due to being an identity matrix, the covariance of \mathbf{w} is diagonalized by the joint Fourier basis of any graph $\boldsymbol{\Sigma}_{\mathbf{w}} = \mathbf{I} = \mathbf{U}_J \mathbf{I} \mathbf{U}_J^*$. This last equation tells us that the JPSD is constant, which implies that similar to the classical case, white noise contains all joint (time-vertex) frequencies.

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

Theorem 3. When a joint filter $f(\mathbf{L}_J)$ is applied to a JWSS process \mathbf{X} , the result \mathbf{Y} remains JWSS with mean $f(0, 0) \mathbb{E}[\mathbf{X}]$ and JPSD that satisfies

$$h_{\mathbf{Y}}(\lambda, \omega) = f^2(\lambda, \omega) \cdot h_{\mathbf{X}}(\lambda, \omega). \quad (14)$$

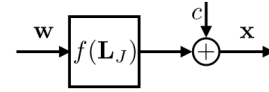
Proof. The output of a filter $f(\mathbf{L}_J)$ can be written in vector form as $\mathbf{y} = f(\mathbf{L}_J) \mathbf{x}$. If the input signal \mathbf{x} is JWSS, we can confirm that the first moment of the filter output is zero,

$\mathbb{E}[f(\mathbf{L}_J) \mathbf{x}] = f(\mathbf{L}_J) \mathbb{E}[\mathbf{x}] = f(0, 0) \mathbb{E}[\mathbf{x}]$. The last equality follows from the fact that by definition $\mathbb{E}[\mathbf{x}]$ is in the null space of \mathbf{L}_J . The computation of the second moment gives

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

which is, from Theorem 1, JWSS as it is diagonalizable by \mathbf{U}_J . \square

As the following diagram illustrates, Theorem 3 provides a simple way to artificially produce JWSS signals with a prescribed PSD f^2 by simply filtering white noise with the joint filter $f(\mathbf{L}_J)$.



The resulting signal will be stationary with PSD f^2 and this holds for white noise abiding to any distribution (not only Gaussian). In the sequel, we assume for simplicity that the signal is centered at 0, i.e., $\mathbb{E}[\mathbf{x}] = \mathbf{0} \cdot \mathbf{1}$.

Whenever it is clear from the context, in the following we simply refer to the TPSD, VPSD, and JPSD as PSD.

IV. JOINT POWER SPECTRAL DENSITY ESTIMATION

As the JPSD is central in our method, we need a reliable way to compute it. Since we take into account the correlation both in the time and in the vertex domain, the actual size of the covariance matrix $\boldsymbol{\Sigma}_{\mathbf{x}}$ is $NT \times NT$. In many cases, this matrix is not computable nor can be even stored. Additionally, if attempt to estimate it using classical covariance estimation methods, the number of samples necessary for obtaining a reasonable estimation accuracy can be prohibitive. The number of samples needed for obtaining a good sample covariance matrix of an n -dimensional process is generally not known, but for distributions with finite second moment it has been shown to be $O(n \log n)$ by Rudelson [19], [20]. In our case, this theorem implies that we need $O(NT \log(NT))$ signals, of NT variables each, to obtain a good estimate of the statistics of a joint process.

To circumvent this issue, we leverage the time-vertex structure of the data. The basic idea behind our approach stems from two established methods used to estimate the TPSD of a temporal signal, namely Bartlett's and Welch's methods [21], which are summarized below.

TPSD estimation methods. In Bartlett's method, the signal (timeseries) is first cut into equally sized segments without overlap. Then, the Fourier transform of each segment is computed. Finally, the PSD is obtained by averaging over segments the squared amplitude of the Fourier coefficients. Welch's method [22] is a generalization that works with overlapping segments. We can see the TPSD estimation of both methods as the averaging over time of the squared coefficients of a Short Time Fourier Transform (STFT). We remind the

reader that STFT is used to extract the frequency content of a temporal signal at a given time, by first selecting a part of the signal using a window and then compute the discrete Fourier transform. More concretely, for a discrete signal \mathbf{s} of length T , the circular discrete sampled STFT of \mathbf{s} at the m -th (out of M) frequency band, and under window \mathbf{g} is

$$\text{STFT}\{\mathbf{s}\}(k, m) \triangleq \sum_{t=1}^T \mathbf{s}(t) \overline{\mathbf{g}(t_k)} e^{-2\pi j \frac{(t-1)(m-1)}{M}},$$

where $t_k = \text{mod}(t - a(k-1), T) + 1$, scalar a is the shift in time between two successive windows [23, equation 1], and $\text{mod}(t, T)$ finds the remainder after division by T i.e., $\text{mod}(t, T) = t - T \lfloor \frac{t}{T} \rfloor$. Note that $k = 0, 1, \dots, \lfloor \frac{T}{a} \rfloor - 1$ is the time band centered at ka and that $m = 1, \dots, M$ is the frequency band index. For additional insights about this transform, we refer the reader to [24], [25].

Joint PSD estimation. Based on the idea that the Bartlett method is an average of STFT coefficients, we propose to use the GFT of the STFT as a tool to estimate the joint PSD. Consider a time window \mathbf{g} and a time-vertex signal \mathbf{X} . We first define the coefficients' tensor as

$$\begin{aligned} \mathbf{C}_{n,k,m} &\triangleq \sum_{i=1}^N [\mathbf{U}_G]_{i,n} \text{STFT}\{\mathbf{x}^i\}(k, m) \\ &= \sum_{i=1}^N [\mathbf{U}_G]_{i,n} \sum_{t=1}^T \mathbf{X}_{i,t} \overline{\mathbf{g}(t_k)} e^{-2\pi j \frac{(t-1)(m-1)}{M}}. \end{aligned}$$

A usual parameter for M is the support size of \mathbf{g} . Then, for half-overlapping windows, we set a to $M/2$. For any discrete vertex frequency λ_n and time frequency $\omega_m = 2\pi m/M$, our JPSD estimator is

$$\tilde{h}(\lambda_n, \omega_m) \triangleq \frac{a}{T \|\mathbf{g}\|_2^2} \sum_{k=0}^{\lfloor T/a \rfloor - 1} \mathbf{C}_{n,k,m}^2 \quad (15)$$

In order to get an estimate of h at $\omega \neq \omega_m$, we interpolate between the known points. Alternatively, with sufficient computation power, one may set $M = T$. Though alternative choices are possible, we suggest using the iterated sine window

$$g(t) = \sin\left(0.5\pi \cos(\pi t/M)^2\right) \chi_{[-M/2, M/2]}(t),$$

where $\chi_{[-M/2, M/2]}(t) = 1$ if $t \in [-M/2, M/2]$ and 0 otherwise, as it turns the STFT into a tight operator for $M = 2a$. We defer an error analysis of the estimator for the longer version of this paper.

Other PSD estimation methods. In case $T \ll N$, two problems arise with the aforementioned method. First, we cannot compute the graph Fourier basis \mathbf{U}_G , and second the number of sample in time might not be sufficient to average over time. To circumvent this problem, one can do the average over the graph vertex using a technique similar to the PSD estimator of [7]. This technique will be studied in future work.

V. OPTIMIZATION FRAMEWORK

We can leverage our definition of stationarity to generalize the optimization framework of [7], useful for denoising, interpolating, and more generally deconvolving stationary processes. Concretely, suppose that our measurements \mathbf{y} are generated by a linear model

$$\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{w}, \quad (16)$$

where, as in the rest of this document, \mathbf{x} and \mathbf{y} are the vectorized version of \mathbf{X}, \mathbf{Y} . Further, suppose that the JPSD of \mathbf{x} is $h_{\mathbf{X}}$, whereas the noise \mathbf{w} is zero mean has JPSD $h_{\mathbf{W}}$ and may follow any distribution. Matrix \mathbf{A} is a general linear operator, not assumed to be jointly diagonalizable with \mathbf{L}_J .

Tikhonov-regularization. When the signal \mathbf{x} varies smoothly on the graph, i.e. is low frequency based, the classical approach of finding \mathbf{x} from \mathbf{y} , consists of solving the following optimization scheme, commonly referred to as Tikhonov-regularization

$$\arg \min_{\mathbf{x}} \|\mathbf{A}\mathbf{x} - \mathbf{y}\|_2^2 + \alpha \mathbf{x}^* \mathbf{L}_J \mathbf{x} \quad (17)$$

Notice that the prior above is separable into two terms

$$\mathbf{x}^* \mathbf{L}_J \mathbf{x} = \text{tr}(\mathbf{X}^* \mathbf{L}_G \mathbf{X}) + \text{tr}(\mathbf{X} \mathbf{L}_T \mathbf{X}^*). \quad (18)$$

As a result, optimization problem (17) can only encode a particular joint time-vertex structure. Additionally this scheme requires the parameter α to be tuned and does not take into account the statistical structure of the signals.

Wiener optimization framework. We instead propose to recover \mathbf{x} as the solution of the Wiener optimization problem

$$\hat{\mathbf{x}} = \arg \min_{\mathbf{x}} \|\mathbf{A}\mathbf{x} - \mathbf{y}\|_2^2 + \|f(\mathbf{L}_J)(\mathbf{x} - \mathbb{E}[\mathbf{x}])\|_2^2, \quad (19)$$

where $f(\lambda, \omega)$ are the joint Fourier penalization weights, defined as

$$f(\lambda, \omega) \triangleq \left| \sqrt{\frac{h_{\mathbf{W}}(\lambda, \omega)}{h_{\mathbf{X}}(\lambda, \omega)}} \right| = \frac{1}{\sqrt{\text{SNR}(\lambda, \omega)}}. \quad (20)$$

In the noise-less case, one alternatively solves the problem

$$\hat{\mathbf{x}} = \arg \min_{\mathbf{x}} \|h_{\mathbf{X}}^{-\frac{1}{2}}(\mathbf{L}_J) \mathbf{x}\|_2^2, \quad \text{subject to } \mathbf{A}\mathbf{x} = \mathbf{y}. \quad (21)$$

Intuitively, the weight $f(\lambda, \omega)$ heavily penalizes frequencies associated with low SNR and vice-versa. Formally, we can show that:

- If \mathbf{X} is a Gaussian process, then the solution of Problem (19) coincides with a MAP estimator.
- If \mathbf{A} is a masking operator, then the solution of Problem (19) coincides with the minimum mean square error linear estimator.
- If $\mathbf{A} = a(\mathbf{L}_J)$ is a joint filter, then the solution of Problem (19) is a joint Wiener filter [9].

The proofs are generalizations of Theorems 3,4 and 5 of [7].

Comparison to the MAP estimator. There are three main advantages of the Wiener optimization framework over a Gaussian MAP estimator based on an empirical covariance matrix estimate. Firstly, assuming stationarity allows for a

more robust estimate of the covariance matrix. This is crucial in this problem since we typically expect the number of variable $N \times T$ to be large and an empirical estimate of the covariance matrix to be expensive. Secondly, storing the covariance might not be possible as it consists of $O((NT)^2)$ elements. On the contrary, the JPSD $h_{\mathbf{X}}$ has only NT elements. Finally, thanks to proximal splitting methods, we can derive an algorithm for solving Problem (19) that requires only the application of \mathbf{A} and spectral graph filtering. On the contrary the classical Gaussian MAP estimator requires the inverse of a large part of the covariance matrix.

VI. EXPERIMENTS

We apply our methods to a weather dataset depicting the temperature of 32 weather stations, over a span of 31 days. Our experiment aims to show that 1) joint stationarity is a useful model, even in datasets which may violate the strict conditions of our definition, and 2) that time-vertex stationarity can yield a significant increase in denoising and recovery accuracy, as compared to time- or vertex-based methods, on a real dataset.

Experimental setup. The French national meteorological service has published in open access a dataset³ with hourly weather observations collected during the Month of January 2014 in the region of Brest (France). The graph is built from the coordinates of the weather stations by connecting all the neighbors in a given radius with a weight function $[\mathbf{W}_G]_{i_1, i_2} = \exp(-k d(i_1, i_2)^2)$, where $d(i_1, i_2)$ is the euclidean distance between the stations i_1 and i_2 . Parameter k is adjusted to as obtain an average degree around 3 (k , however, is not a sensitive parameter). As sole pre-processing, we remove the mean (over time and stations) of the temperature. This is equivalent to removing the first moment.

The dataset, which consisted of a total of $T = 744$ timesteps, was split into two parts of size ρT and $(1 - \rho)T$, respectively. We use the first part of the dataset to estimate the PSD and the second to quantify the joint filter performance. We compare our joint method to the state-of-the-art wiener filters for the disjoint time/vertex domains, which are known to outperform non-statistics based methods, such as graph/time Tikhonov and graph/time TV. To highlight the benefit of the joint approach, in the disjoint cases we use the entire dataset to estimate the PSD (for $\rho = 1$ the same data are used for both training and testing).

Denoising. For this experiment, we add Gaussian random noise to the data and remove the noise thanks to Wiener filter ($\mathbf{A} = \mathbf{I}$ in problem (19)). The result is displayed in Figure 1. Joint stationarity outperforms time or vertex stationarity especially when the noise level is high. Indeed, joint stationarity allows the estimator to average over more samples. In order to obtain a good denoising, we need a good JPSD estimation. The effect of the dataset size can be observed through the parameter ρ , with larger ρ resulting in higher accuracy. Especially for large input SNR, the joint approach becomes particularly meaningful as it outperforms

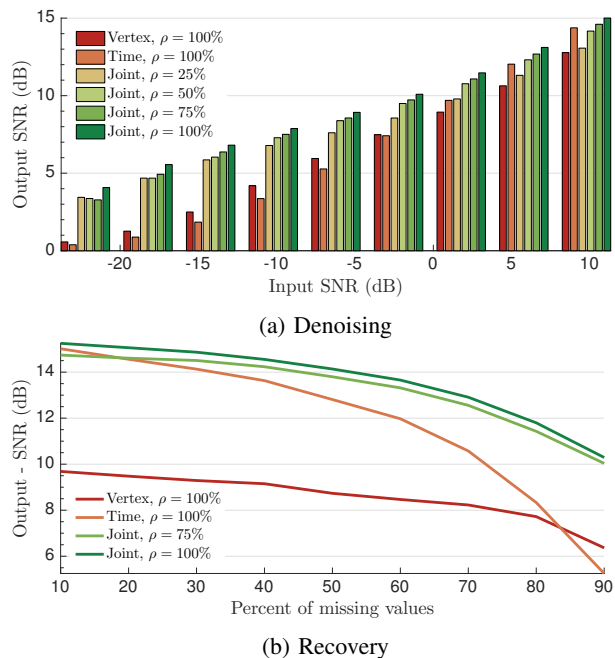


Fig. 1: Experiments on Molene temperatures. The joint approach becomes especially meaningful when the available data are very noisy or are few. The recovery performance is slightly improved when a larger percentage ρ of data are available for training.

other approaches, even when a very small portion of the data is used for JPSD estimation (whereas the time and vertex based methods $\rho = 1$, meaning that they use the entire dataset for PSD estimation).

Recovery. We also consider a recovery problem, where a given percentage of entries of matrix \mathbf{X} is missing. Figure 1 depicts the recovery error obtained using problem (21). Again, we observe a significant improvement over competing methods. This improvement is achieved because the joint approach leverages the correlation both in the time and in the vertex domain: each random variable in a TWSS or VWSS process is dependent on only $T - 1$ or $N - 1$ other random variables, respectively (rather than $NT - 1$ as in the joint case), implying a higher recovery variance.

VII. CONCLUSION

This paper proposed a novel definition of (wide-sense) stationarity appropriate for time-varying graph signals. We showed that joint stationarity possess a number of useful properties, that are familiar from the classical setting. Based on our definition, we proposed a Wiener optimization framework and the accompanying PSD estimation method, which together can be used to for solving the problem of inverting a rank-deficient linear system under a jointly stationary input and disturbance. The proposed optimization framework is optimal in the mean-squared error sense and scales well with the number of time samples. In our experiment with a weather dataset, the joint approach was shown to yield a significant

³Access to the raw data is possible directly from https://donneespubliques.meteofrance.fr/donnees_libres/Hackathon/RADOMEH.tar.gz

benefit over disjoint statistical methods for signal denoising and recovery.

The longer version of this paper will expand our analysis and evaluate our approach in a larger set of experiments. We will additionally make a detailed complexity analysis and propose solutions to avoid the computationally expensive diagonalization of the graph Laplacian L_G . We remark that our simulations were done using the GSPBOX [26], the UNLocBoX [27], and the LTFAT [23]. The code reproducing all figures will be made available soon.

REFERENCES

- [1] L. A. Adamic and N. Glance, "The political blogosphere and the 2004 us election: divided they blog," in *Proceedings of the 3rd international workshop on Link discovery*. ACM, 2005, pp. 36–43.
- [2] 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.
- [3] 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.
- [4] F. Zhang and E. R. Hancock, "Graph spectral image smoothing using the heat kernel," *Pattern Recognition*, vol. 41, no. 11, pp. 3328–3342, 2008.
- [5] A. J. Smola and R. Kondor, "Kernels and regularization on graphs," in *Learning theory and kernel machines*. Springer, 2003, pp. 144–158.
- [6] M. Belkin and P. Niyogi, "Semi-supervised learning on riemannian manifolds," *Machine learning*, vol. 56, no. 1-3, pp. 209–239, 2004.
- [7] N. Perraudin and P. Vandergheynst, "Stationary signal processing on graphs," *arXiv preprint arXiv:1601.02522*, 2016.
- [8] V. Kalofolias, "How to learn a graph from smooth signals," *arXiv preprint arXiv:1601.02513*, 2016.
- [9] A. Loukas and D. Foucard, "Frequency analysis of temporal graph signals," *arXiv preprint arXiv:1602.04434*, 2016.
- [10] B. Girault, "Stationary graph signals using an isometric graph translation," in *Signal Processing Conference (EUSIPCO), 2015 23rd European*. IEEE, 2015, pp. 1516–1520.
- [11] A. G. Marques, S. Segarra, G. Leus, and A. Ribeiro, "Stationary graph processes and spectral estimation," *arXiv preprint arXiv:1603.04667*, 2016.
- [12] A. Sandryhaila and J. M. Moura, "Discrete signal processing on graphs," *IEEE transactions on signal processing*, vol. 61, pp. 1644–1656, 2013.
- [13] D. K. Hammond, P. Vandergheynst, and R. Gribonval, "Wavelets on graphs via spectral graph theory," *Applied and Computational Harmonic Analysis*, vol. 30, no. 2, pp. 129–150, 2011.
- [14] 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.
- [15] D. I. Shuman, B. Ricaud, and P. Vandergheynst, "Vertex-frequency analysis on graphs," *arXiv preprint arXiv:1307.5708*, 2013.
- [16] G. Strang, "The discrete cosine transform," *SIAM review*, vol. 41, no. 1, pp. 135–147, 1999.
- [17] N. Perraudin, B. Ricaud, D. Shuman, and P. Vandergheynst, "Global and local uncertainty principles for signals on graphs," *arXiv preprint arXiv:1603.03030*, 2016.
- [18] N. Wiener, "Generalized harmonic analysis," *Acta mathematica*, vol. 55, no. 1, pp. 117–258, 1930.
- [19] M. Rudelson, "Random vectors in the isotropic position," *Journal of Functional Analysis*, vol. 164, no. 1, pp. 60–72, 1999.
- [20] 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.
- [21] M. S. Bartlett, "Periodogram analysis and continuous spectra," *Biometrika*, pp. 1–16, 1950.
- [22] 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.
- [23] 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, M. Aramaki, O. Derrien, R. Kronland-Martinet, and S. Ystad, Eds. Springer International Publishing, 2014, pp. 419–442. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-12976-1_25
- [24] K. Gröchenig, *Foundations of time-frequency analysis*. Springer Science & Business Media, 2013.
- [25] H. G. Feichtinger and T. Strohmer, *Gabor analysis and algorithms: Theory and applications*. Springer Science & Business Media, 2012.
- [26] 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.
- [27] N. Perraudin, D. Shuman, G. Puy, and P. Vandergheynst, "UNLocBoX A matlab convex optimization toolbox using proximal splitting methods," *ArXiv e-prints*, Feb. 2014.