WSS Processes and Wiener Filters on Digraphs

Mohammad Bagher Iraji, Mohammad Eini and Arash Amini, *Senior, IEEE*

***Abstract*—In this paper, we generalize the concepts of kernels, weak stationarity and white noise from undirected to directed graphs (digraphs) based on the Jordan decomposition of the shift operator. We characterize two types of kernels (type-I and type-II) and their corresponding localization operators for digraphs. We analytically study the interplay of these types of kernels with the concept of stationarity, specially the filtering properties. We also generalize graph Wiener filters and the related optimization framework to digraphs. For the special case of Gaussian processes, we show that the Wiener filtering again coincides with the MAP estimator. We further investigate the linear minimum mean-squared error (LMMSE) estimator for the non-Gaussian cases; the corresponding optimization problem simplifies to a Lyapunov matrix equation. We propose an algorithm to solve the Wiener optimization using proximal splitting methods. Finally, we provide simulation results to verify the provided theory.**

***Index Terms*—Graph signal, graph Fourier transform, power spectral density, directed graph, Wiener filter.**

I. Introduction

In classical signal processing, signals are often functions of time or space where the adjacency of temporal or spatial points can be considered as a regular grid. In general, however, the signal domain may be quite irregular; representing such signals over graphs has emerged as a successful approach in recent years leading to the field of graph signal processing [1]. Similar to conventional signals, graph signals could either be deterministic or stochastic. Therefore, any probability measure on the space of graph signals leads to a random graph process.

The stationary processes, or specifically, wide sense stationary scalar processes are a well-studied sub-class of random processes which are generalized to processes defined on undirected graphs [2]–[5]. The probability laws governing wide sense stationary processes allow for obtaining certain estimators such as the linear minimum mean-squared error (LMMSE) estimator employed in various applications including signal denoising (e.g., Wiener filtering), signal retrieval, inverse problems and etc. We should highlight that the stationarity property has potential applications in [6]:

- improving the covariance estimation (instead of estimating all the entries of the covariance matrix, we only need to estimate its eigenvalues or corresponding filter coefficients),
- denoising a set of observed graph signals,
- interpolating (prediction) a graph signal using observations over a subset of vertices, and
- extracting the graph topology.

Graph Wiener filters on undirected graphs have been studied in [5] and used for image and graph signal denoising in [7]. In particular, an algorithm was proposed in [5] for solving the Wiener optimization using proximal splitting methods [8], [9]. Besides, the stationarity level of some real data was evaluated in [5] for the undirected scenario. Distributed implementation of Wiener filtering in the undirected graph setting has been proposed in [10]. Also, new estimators for the graph signal recovery from nonlinear measurements were proposed in [11] and conditions under which the estimators coincide with MAP were stated; a special case is when the signal and noise are graph wide-sense stationary (GWSS) while the measurement is linear. This special case matches [5], [12] where the new estimators coincide with graph Wiener filter [5], LMMSE, and GSP-LMMSE estimator [12].

Although directed graphs (digraphs) appear in many applications, the tools for analyzing digraphs are not developed as much as those for undirected graphs. Unfortunately, many concepts on undirected graphs are not automatically applicable to digraphs. Among the few works on digraphs, we can name [13] in which the graph Fourier transform is defined based on the Jordan decomposition of the shift matrix. Devising a different strategy in [14], a symmetric Laplacian was introduced for digraphs based on the random walk matrix.

A. Contributions

In this paper, we focus on stationary processes on digraphs and define the concept from the base. We adopt the definition of Fourier transform on digraphs based on the Jordan decomposition and the related notations from [13], [15]. Next, we consider kernels on digraphs and divide them into two types; the first type stands for the conventional filters and the second type is used as the basis for the definition of the GWSS processes. After examining the kernel properties, we define GWSS processes on digraphs and their spectrum matrix using kernels of type-II. Then, we define the white noise on digraphs and verify that the output of a type-I kernel to a GWSS process is GWSS again. Based on our definition of stationarity, we answer the question whether every GWSS process is obtained by passing a white noise through a type-I kernel.

We further investigate the implications of the provided stationarity definition. As expected, we show that the Wiener filter coincides with the MAP estimator for Gaussian processes and with the LMMSE estimator for non-Gaussian processes. Since the optimization framework involves a Lyapunov matrix equation [16]–[23], we propose an algorithm similar to [5] for solving the Wiener optimization on digraphs in the spectral domain.

Manuscript received April 11, 2024; revised August 07, 2024, and accepted November 20, 2024.

The authors are with the department of electrical engineering at Sharif university of technology, Tehran, Iran. (emails: mbiraji@gmail.com, Mohammad.einy.afm@gmail.com and aamini@sharif.edu)

This paper has supplementary downloadable material available at http://ieeexplore.ieee.org., provided by the author. The material includes some explanations and proofs about the technical results. This material is 374KB in size.

As for numerical validations, we provide simulation results on real French meteorological data as in [5]. While we show that the data better fits into a digraph stationary model rather than the undirected model, we perform denoising and inpainting tasks (in the latter, we estimate the missing samples). In each application, we compare our results with those obtained by modeling the data using undirected graphs. We show that our Wiener filter outperforms the other competing methods, especially in lower SNRs, if the noise is digraph-stationary.

B. Paper organization

The rest of this paper is organized as follows: we first provide some basic preliminaries about graph signal processing in Section II. Next, we review the weak stationarity concept for undirected graphs in Section III. Our main contributions are provided in Section IV which consist of the definition of stationarity and its implications. In Section V, we report numerical results to support the theories and finally, we conclude the paper in Section VI. To facilitate the reading of the paper, we have postponed most of the proofs to the appendices and the supplementary material.

II. Preliminaries

A. Graph theory

A digraph G is an ordered pair $(\mathcal{V},\mathcal{E})$, where $\mathcal{V}\neq\emptyset$ is the finite set of vertices and $\mathcal{E}\subseteq\mathcal{V}\times\mathcal{V}$ is the set of edges connecting the vertices. Here, $\times$ denotes the Cartesian product of sets. There is an edge from vertex m to vertex n iff $(m,n)\in\mathcal{E}$. If

$$\forall m,n\in\mathcal{V},(m,n)\in\mathcal{E}\Rightarrow(n,m)\in\mathcal{E},$$

then, the digraph is called symmetric. For any two distinct and connected vertices m,n in a symmetric digraph, one can form the new set $\mathcal{E}_{\text{simple}}$ consisting of $\{m,n\}$ sets; the pair $(\mathcal{V},\mathcal{E}_{\text{simple}})$ is called a simple or undirected graph. Therefore, for every symmetric digraph, there is a corresponding simple graph. The adjacency matrix $\mathbf{A}$ of a graph/digraph G with $|\mathcal{V}|=N$ vertices is a binary-valued $N\times N$ matrix with entries:

$$a_{mn}\triangleq\begin{cases}1, & \text{if } (m,n)\in\mathcal{E},\\ 0, & \text{if } (m,n)\notin\mathcal{E}.\end{cases}\tag{1}$$

A weighted graph/digraph consists of positive real values w_{ij} assigned to the edges $(i,j)\in\mathcal{E}$ as their weight. We denote the $N\times N$ weight matrix by $\mathbf{W}$. The degree d_{mm} of a vertex m in a weighted (undirected) graph is defined as the weight sum of the edges connected to m:

$$d_{mm}=\sum_n w_{mn}.\tag{2}$$

The degree matrix $\mathbf{D}$ is a diagonal matrix whose m-th entry on the diagonal is d_{mm}. Subsequently, the Laplacian matrix of a weighted (undirected) graph denoted by $\mathbf{L}$ is defined as:

$$\mathbf{L}=\mathbf{D}-\mathbf{W}.\tag{3}$$

B. Spectral analysis of graph signals

The $N\times N$ *shift* matrix of the graph G denoted by $\mathbf{S}$ consists of s_{ij} entries that are nonzero only if $i=j$ or $(i,j)\in\mathcal{E}$ [24], [6]. The adjacency, weight and Laplacian matrices are all special cases of the shift matrix.

Definition 1. For a graph G with N vertices, a graph signal $\mathbf{x}$ is an $N\times 1$ vector whose n-th entry, i.e., $\mathbf{x}[n]$, is a real or complex number assigned to the n-th vertex of the graph:

$$\mathbf{x}=\Big[\mathbf{x}[1],\mathbf{x}[2],\ldots,\mathbf{x}[N]\Big]^T.\tag{4}$$

Obviously, if signal values on the vertices are random, we will have a random graph process.

Definition 2. (Undirected graph Fourier transform) If $\mathbf{S}$ is real and symmetric, then, it is unitarily diagonalizable as $\mathbf{S}=\mathbf{U}\mathbf{\Lambda}\mathbf{U}^T$, where $\mathbf{U}$ is a unitary matrix and $\mathbf{\Lambda}$ is diagonal. The graph Fourier transform (GFT) of any graph signal $\mathbf{x}$ on this graph is then, defined as:

$$\widehat{\mathbf{x}}=\mathrm{GFT}\{\mathbf{x}\}\triangleq\mathbf{U}^T\mathbf{x}.\tag{5}$$

Also, the inverse GFT can be expressed as:

$$\mathbf{x}=\mathrm{IGFT}\{\widehat{\mathbf{x}}\}=\mathbf{U}\widehat{\mathbf{x}}.\tag{6}$$

C. Scalar stationary processes

In general, the concept of stationarity asserts that a process and its shifted versions have the same probability laws. There are two well-known definitions for stationary processes (classical/scalar), which consist of strict-sense stationary (SSS) processes and wide-sense stationary (WSS) processes. The latter is more common in engineering contexts and therefore, we focus on WSS processes in this paper.

Definition 3. A scalar random process $x[n]$ is called WSS iff

$$\begin{cases}\forall n,\quad \mathbb{E}\big[x[n]\big]=\mu_x=const.,\\ \forall i,m,\quad \mathbb{E}\big[(x[i]-\mu_x)\overline{(x[m]-\mu_x)}\big]=r_x[i-m],\end{cases}\tag{7}$$

where r_x is called the autocorrelation function of the process.

Here, our goal is to define WSS processes on digraphs similar to that of the scalar case as much as possible. The main barrier is that the graph shift operator, unlike its scalar counterpart, is not energy-preserving; hence, we cannot expect a graph signal and its shifted versions to have the same law. Also, the asymmetry of the weight/adjacency matrix in digraphs prevents us from having an eigen-decomposition.

III. WSS Processes on Undirected Graphs

Before considering the digraphs, we review the existing stationarity concepts over undirected graphs.

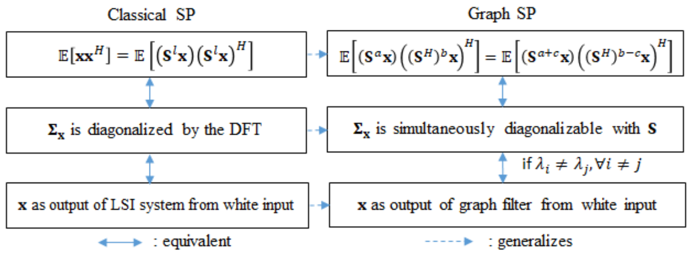

Fig. 1. Generalization of stationarity from the classical case to the graph case.

A. Main definition

Assume that $\mathbf{S}$ is unitarily diagonalizable; hence, we can write $\mathbf{S} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^H$, where $\mathbf{U}$ is a unitary matrix and $\mathbf{\Lambda}$ is diagonal [25]. Since the graph is undirected and the shift matrix, e.g. the adjacency or the Laplacian matrix, is likely to be Hermitian, this assumption is logical. By convention, we call a process white if its covariance matrix is a scaled version of the identity matrix. A graph filter is defined as a polynomial function of $\mathbf{S}$ which is linear and shift-invariant (LSI) [25, Equation (12.1)].

Definition 4. The graph process $\mathbf{x}$ could be called a GWSS process (based on the reference) iff it satisfies one of the following conditions:

1) $\mathbf{x}$ is the response of an LSI polynomial graph filter to a white input $\mathbf{n}$ [25, Definition 12.1].
2) We have [25, Definition 12.2.a]

$$\forall a, b, 0 \leq c \leq b,$$
$$\mathbb{E}\left[\mathbf{S}^a\mathbf{x}\big((\mathbf{S}^H)^b\mathbf{x}\big)^H\right] = \mathbb{E}\left[\mathbf{S}^{a+c}\mathbf{x}\big((\mathbf{S}^H)^{b-c}\mathbf{x}\big)^H\right]. \quad (8)$$

3) The covariance matrix of the process $\mathbf{x}$, i.e.,

$$\mathbf{\Sigma_x} = \mathbb{E}[\mathbf{x}\mathbf{x}^H],$$

and $\mathbf{S}$ are simultaneously diagonalizable [25, Definition 12.2.b].

It can be shown that Conditions 2 and 3 in the above definition are equivalent. Also, if all the eigenvalues of $\mathbf{S}$ are distinct, the first condition is equivalent to the other conditions.

As shown in Figure 1, Definition 4 generalizes the classical definitions and theorems for scalar signals to graph signals.

Definition 5. The power spectral density (PSD) vector of $\mathbf{x}$ or $\mathbf{p_x}$, is defined as the vector of eigenvalues of the matrix $\mathbf{\Sigma_x}$.

B. Equivalent definition

Here we explain an equivalent and more intuitive definition for stationary processes on undirected graphs. For this purpose, we need to define the localized version of a vector $\mathbf{x}$ with respect to a given vertex m [1, Equation (46)] as[1]

$$\mathbf{x}_m \triangleq \mathbf{x} *_G \boldsymbol{\delta}_m, \quad (9)$$

[1]The localization operator is inherently a shift operator which should not be mistaken with the graph shift operator $\mathbf{S}$; the localization operator is sometimes referred to as the vertex shift operator.

Fig. 2. Passing a GWSS process through a graph filter, results in a GWSS process.

where $\boldsymbol{\delta}_m$ is a signal whose value is 1 at vertex m and 0 elsewhere, and $*_G$ represents the graph convolution (to compute the graph convolution of two signals we first apply the GFT to both, multiply the results entry-wise, and take the inverse GFT). The expanded version of (9) is

$$\mathbf{x}_m[i] = \sum_{j=1}^{N} \big(\widehat{\mathbf{x}}[j] u_{ij} \overline{u_{mj}}\big). \quad (10)$$

This confirms that $\mathbf{x}_m$ is a linear transformation of $\mathbf{x}$.

Definition 6. The process $\mathbf{x}$ is a GWSS process iff [5]:

1) either $\mathbb{E}\big[\mathbf{x}[i]\big] = 0$ for all i, or $\mathbb{E}\big[\mathbf{x}[i]\big] = \mu_{\mathbf{x}} \neq 0$ is constant (for all i) and $\mathbf{j} = [1, 1, \ldots, 1]^T$ is an eigenvector of $\mathbf{S}$, and
2) there exists a vector $\mathbf{p_x}$ (PSD of $\mathbf{x}$) such that for all i, m:

$$\mathbb{E}\big[(\mathbf{x}[i] - \mu_{\mathbf{x}})\overline{(\mathbf{x}[m] - \mu_{\mathbf{x}})}\big] = \sum_{j=1}^{N} \big(\mathbf{p_x}[j] u_{ij} \overline{u_{mj}}\big).$$

By comparing the second condition in Definition 6 with $\mathbf{x}_m[i]$ in (10), we observe that the covariance matrix of $\mathbf{x}$ at location i, m needs to be equal to a hypothetical graph signal (which corresponds to $\mathbf{p_x}$ in the GFT domain) at vertex i, after being localized with respect to vertex m (vertex shift). This interpretation better matches the classical definition of WSS processes. Rewriting this second condition in matrix form, we arrive at

$$\mathbf{\Sigma_x} = \mathbf{U}\, \mathrm{diag}(\mathbf{p_x})\, \mathbf{U}^{-1}, \quad (11)$$

which is in turn equivalent to the simultaneous diagonalization of $\mathbf{\Sigma_x}$ and $\mathbf{S}$, or $\mathbf{\Sigma_x}\mathbf{S} = \mathbf{S}\mathbf{\Sigma_x}$. It is worth mentioning that since the covariance matrix of a white process $\mathbf{w}$ is a scaled identity matrix ($\mathbf{\Sigma_w} = \sigma^2\mathbf{I}$) and $\mathbf{I}$ is simultaneously diagonalizable with all diagonalizable matrices, the white process is GWSS in all undirected graphs irrespective of the choice of the shift operator.

Property 1. If $\mathbf{x}$ is a GWSS process, $\mathbf{H}$ is an LSI polynomial graph filter with frequency response $\widehat{\mathbf{h}}$ and $\mathbf{y} \triangleq \mathbf{H}\mathbf{x}$ (Figure 2), then [25, Property 12.1],

(a) $\mathbf{y}$ is also GWSS with covariance matrix

$$\mathbf{\Sigma_y} = \mathbf{H}\mathbf{\Sigma_x}\mathbf{H}^H,$$

and

(b) the PSD of $\mathbf{y}$, $\mathbf{p_y}$, is found by

$$\mathbf{p_y} = |\widehat{\mathbf{h}}|^2 \circ \mathbf{p_x},$$

where $\circ$ stands for the entry-wise multiplication of two vectors and $|\widehat{\mathbf{h}}|^2$ is again computed in an entry-wise fashion with respect to $\widehat{\mathbf{h}}$.

Property 2. If $\mathbf{x}$ is a GWSS process, then, its Fourier transform components, i.e., entries of $\widehat{\mathbf{x}} \triangleq \mathbf{U}^H\mathbf{x}$, shall be uncorrelated [25, Property 12.2].

IV. Main Results

As stated earlier, our goal in this paper is to extend the concepts of stationarity and Wiener filters to digraphs. In [26], the basis for such extension was built. In this paper, we continue the work with more details. We first propose a definition and then, investigate its consequences on Wiener filtering.

A. Definitions and properties

In digraphs, $\mathbf{S}$ is asymmetric in general and may not be diagonalizable. So it is necessary to rewrite the definition of GWSS processes for digraphs appropriately. When $\mathbf{S}$ is not diagonalizable, it is common to use its Jordan decomposition. In this section, we use the notations in appendix A of [13] and also [15] for Jordan decomposition. According to the Jordan normal form theorem, we know that every [shift] matrix can be written as $\mathbf{S} = \mathbf{V}\mathbf{J}\mathbf{V}^{-1}$ where $\mathbf{J}$ is a block diagonal matrix. If $\mathbf{S}$ is unitarily diagonalizable, then, $\mathbf{V}$ and $\mathbf{J}$ are the same as $\mathbf{U}$ and $\mathbf{\Lambda}$ in [5], respectively.

Suppose that $\mathbf{S}$ has the Jordan form $\mathbf{V}\mathbf{J}\mathbf{V}^{-1}$. If $\mathbf{S}$ has M distinct eigenvalues, then, $\mathbf{J}$ is a block diagonal matrix composed of M segments; in particular, the m-th segment consists of D_m Jordan blocks where D_m is the number of eigenvectors corresponding to λ_m. We denote the d-th Jordan block of the m-th segment by $\mathbf{J}_{(m,d)}(\lambda_m)$. This Jordan block is of size $R_{(m,d)} \times R_{(m,d)}$:

$$\mathbf{J}_{m,d}(\lambda_m) = \begin{bmatrix} \lambda_m & 1 & 0 & \dots & 0 \\ 0 & \lambda_m & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1 \\ 0 & 0 & 0 & \dots & \lambda_m \end{bmatrix}, \tag{12}$$

$$\mathbf{J} = \begin{bmatrix} \mathbf{J}_{1,1}(\lambda_1) & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \mathbf{J}_{1,2}(\lambda_1) & \dots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & \mathbf{J}_{M,D_M}(\lambda_M) \end{bmatrix}. \tag{13}$$

Let $\mathbf{V}' \triangleq (\mathbf{V}^H)^{-1} = \mathbf{V}^{-H}$. The columns of $\mathbf{V}$ and $\mathbf{V}'$ are called the Jordan basis vectors and their dual basis vectors, respectively. The columns of $\mathbf{V}'$ are related to the left generalized eigenspace of $\mathbf{S}$. Indeed, the columns of $\mathbf{V}$ and $\mathbf{V}'$ are right and left generalized eigenvectors of $\mathbf{S}$, respectively. So, we have: ${\mathbf{V}'}^H\mathbf{V} = \mathbf{V}^H\mathbf{V}' = \mathbf{I}$. Also, the graph Fourier transform of $\mathbf{x}$ has been defined in [13] as $\widehat{\mathbf{x}} \triangleq \mathbf{V}^{-1}\mathbf{x}$.

Definition 7 (type-I kernel)**.** Let $g : \mathbb{C} \to \mathbb{C}$ be a function that is analytic on a neighborhood of the spectrum of $\mathbf{S}$ [27]. By the *type-I kernel* or filter corresponding to g (which we denote by $g(\mathbf{S})$), we imply the matrix

$$g(\mathbf{S}) \triangleq \mathbf{V}g(\mathbf{J})\mathbf{V}^{-1}, \tag{14}$$

where

$$g(\mathbf{J}) \triangleq \begin{bmatrix} g(\mathbf{J}_{1,1}(\lambda_1)) & \dots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \dots & g(\mathbf{J}_{M,D_M}(\lambda_M)) \end{bmatrix}, \tag{15}$$

and

$$g\big(\mathbf{J}_{m,d}(\lambda_m)\big) \triangleq \begin{bmatrix} g(\lambda_m) & g'(\lambda_m) & \dots & \frac{g^{(k-1)}(\lambda_m)}{(k-1)!} \\ 0 & g(\lambda_m) & \dots & \frac{g^{(k-2)}(\lambda_m)}{(k-2)!} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & g(\lambda_m) \end{bmatrix}, \tag{16}$$

with k representing $R_{m,d}$. Note that $g\big(\mathbf{J}_{m,d}(\lambda_m)\big)$ is a toeplitz upper-triangular matrix.

Property 3. If the analytic function g used in Definition 7 is a polynomial, then, $g(\mathbf{S})$, $g(\mathbf{J})$ and $g(\mathbf{J}_{m,d}(\lambda_m))$ defined above are equal to the polynomial evaluated at matrices $\mathbf{S}$, $\mathbf{J}$ and $\mathbf{J}_{m,d}(\lambda_m)$, respectively [13], [27, Section 1.2.1].

Property 4. Every type-I kernel is an LSI filter. Also, type-I kernels commute [27, Theorem 1.13.a,e].

Since type-I kernels are not enough for the generalization of GWSS processes to digraphs, we introduce type-II kernels. Suppose h^{II} is a function on the space of square matrices that maps each Jordan block $\mathbf{J}_{m,d}$ to a square matrix of the same size. Unlike the output of type-I kernels in (16), the resulting square matrix may not be upper-triangular.

Definition 8 (type-II kernel)**.** A *type-II kernel* corresponding to h^{II}, is a matrix defined as below and denoted by $h_{\mathbf{S}}$:

$$h_{\mathbf{S}} \triangleq \mathbf{V}h^{\mathrm{II}}(\mathbf{J})\mathbf{V}^H, \tag{17}$$

$$h^{\mathrm{II}}(\mathbf{J}) \triangleq \begin{bmatrix} h^{\mathrm{II}}(\mathbf{J}_{1,1}(\lambda_1)) & \dots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \dots & h^{\mathrm{II}}(\mathbf{J}_{M,D_M}(\lambda_M)) \end{bmatrix}. \tag{18}$$

Note that the matrix $h^{\mathrm{II}}(\mathbf{J})$ is a block diagonal matrix.

Property 5. If the function h^{II} in Definition 8 is a polynomial, then, $h_{\mathbf{S}}$ defined above may not necessarily be equal to the polynomial evaluated at the matrix $\mathbf{S}$, i.e., $h^{\mathrm{II}}(\mathbf{S})$ (See Appendix A in the supplementary material).

Property 6. Every type-II kernel is linear, but not necessarily shift-invariant (See Appendix A in the supplementary material).

Next, we extend the notion of GWSS processes to digraphs based on the defined type-I and type-II kernels.

Definition 9. The localized version of a type-I kernel g at vertex i (shifted to vertex i) denoted by $\mathcal{T}_i g$ is a vector whose n-th entry is

$$\big(\mathcal{T}_i g\big)[n] \triangleq \big(g(\mathbf{S})\boldsymbol{\delta}_i\big)[n] = g(\mathbf{S})[n,i]. \tag{19}$$

Similarly, the localized version of a type-II kernel h^{II} at vertex i, represented by $\mathcal{T}_i h^{\mathrm{II}}$ is a vector whose n-th entry corresponds to

$$\big(\mathcal{T}_i h^{\mathrm{II}}\big)[n] \triangleq (h_{\mathbf{S}}\boldsymbol{\delta}_i)[n] = h_{\mathbf{S}}[n,i]. \tag{20}$$

Definition 10. We call a graph process $\mathbf{x}$ GWSS iff for some constant $m_{\mathbf{x}} \in \mathbb{C}$ and type-II kernel h^{II}, we have that

$$\begin{aligned} \forall i,\ \mathbf{m}_{\mathbf{x}}[i] &= \mathbb{E}\big[\mathbf{x}[i]\big] = m_{\mathbf{x}} \in \mathbb{C}, \\ \forall i, n,\ \boldsymbol{\Sigma}_{\mathbf{x}}[i,n] &= \mathbb{E}\big[(\mathbf{x}[i] - m_{\mathbf{x}})(\mathbf{x}[n] - m_{\mathbf{x}})^*\big] \\ &= \big(\mathcal{T}_n h^{\mathrm{II}}\big)[i] = h_{\mathbf{S}}[i,n], \end{aligned} \tag{21}$$

where either $m_{\mathbf{x}} = 0$, or $m_{\mathbf{x}} \neq 0$ and $\mathbf{j} = [1, 1, \ldots, 1]^T$ is an eigenvector of $\mathbf{S}$, and the second condition implies that the covariance matrix of the process is a localized type-II kernel.

Corollary 1. A graph process $\mathbf{x}$ is GWSS iff there is a type-II kernel $\gamma_{\mathbf{x}}$ such that

$$\boldsymbol{\Sigma}_{\mathbf{x}} = \mathbf{V}\gamma_{\mathbf{x}}(\mathbf{J})\mathbf{V}^H. \tag{22}$$

If the condition (22) holds, then, the block diagonal matrix $\boldsymbol{\Gamma}_{\mathbf{x}} = \gamma_{\mathbf{x}}(\mathbf{J})$ is called the spectrum of the process $\mathbf{x}$ and we have

$$\begin{aligned} \gamma_{\mathbf{x}}(\mathbf{J}_{m,d}) &= (\mathbf{V}^{-1}\boldsymbol{\Sigma}_{\mathbf{x}}\mathbf{V}^{-H})_{(m,d),(m,d)} \\ &= \mathbf{V}^{-1}_{(m,d),:}\boldsymbol{\Sigma}_{\mathbf{x}}\mathbf{V}^{-H}_{:,(m,d)}. \end{aligned} \tag{23}$$

In the above equation, (m,d) (e.g. in $(.)_{(m,d),(m,d)}$) means the indices of rows or columns that correspond to the Jordan block $\mathbf{J}_{m,d}(\lambda_m)$ and : refers to all the rows or columns. From now on, we denote the pair (m,d) with l for brevity. Note that Definition 6 is a special case of Definition 10.

Note 1. Since type-II kernels may not be LSI, the equation $\boldsymbol{\Sigma}_{\mathbf{x}}\mathbf{S} = \mathbf{S}\boldsymbol{\Sigma}_{\mathbf{x}}$ may not hold in general.

Property 7. The spectrum of a GWSS process $\mathbf{x}$, i.e., $\gamma_{\mathbf{x}}(\mathbf{J})$, and all of the blocks on its diagonal are Hermitian and positive semi-definite. (Proof is provided in Appendix B in the supplementary material).

Property 8. The spectrum matrix of a GWSS process and all the blocks on its diagonal can be written as $\mathbf{L}^H\mathbf{L}$ where $\mathbf{L}$ is a lower-triangular matrix. (Proof is provided in Appendix C in the supplementary material).

Example 1. The graph process $\mathbf{x}$ with mean $m_{\mathbf{x}}$ (constant over the vertices) and covariance $\boldsymbol{\Sigma}_{\mathbf{x}} = \sigma^2\mathbf{V}\mathbf{V}^H$ is GWSS with respect to $\mathbf{S} = \mathbf{V}\mathbf{J}\mathbf{V}^{-1}$. We call such a process white, as its spectrum is $\gamma_{\mathbf{x}}(\mathbf{J}) = \sigma^2\mathbf{I}$.

Property 9. If the graph process $\mathbf{y}$ is obtained by passing the GWSS process $\mathbf{x}$ through a type-I kernel $g(\mathbf{S})$ (Figure 3a), then, the process $\mathbf{y}$ is also GWSS.

Proof:

$$\begin{aligned} \mathbf{y} &= g(\mathbf{S})\mathbf{x} \Rightarrow \boldsymbol{\Sigma}_{\mathbf{y}} = \mathbb{E}[\mathbf{y}\mathbf{y}^H] \\ &= g(\mathbf{S})\mathbb{E}[\mathbf{x}\mathbf{x}^H]g(\mathbf{S})^H = g(\mathbf{S})\,\boldsymbol{\Sigma}_{\mathbf{x}}\,g(\mathbf{S})^H \\ &= \mathbf{V}\,g(\mathbf{J})\,\mathbf{V}^{-1}\,\mathbf{V}\,\gamma_{\mathbf{x}}(\mathbf{J})\,\mathbf{V}^H\,\mathbf{V}^{-H}\,g(\mathbf{J})^H\,\mathbf{V}^H \\ &= \mathbf{V}\,\big(g(\mathbf{J})\gamma_{\mathbf{x}}(\mathbf{J})g(\mathbf{J})^H\big)\,\mathbf{V}^H. \end{aligned} \tag{24}$$

Note that the product of the three block diagonal matrices $g(\mathbf{J})$, $\gamma_{\mathbf{x}}(\mathbf{J})$, and $g(\mathbf{J})^H$ is again block diagonal. Representing this product as $\gamma_{\mathbf{y}}(\mathbf{J})$, we have that

$$\boldsymbol{\Sigma}_{\mathbf{y}} = \mathbf{V}\gamma_{\mathbf{y}}(\mathbf{J})\mathbf{V}^H,$$

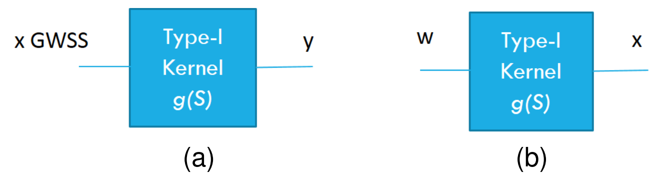

Fig. 3. Passing (a) a GWSS process and (b) a white noise through a type-I kernel

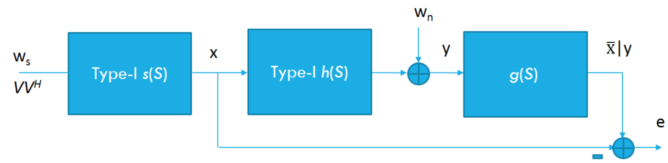

Fig. 4. Wiener estimation

where

$$\gamma_{\mathbf{y}}(\mathbf{J}) = g(\mathbf{J})\gamma_{\mathbf{x}}(\mathbf{J})g(\mathbf{J})^H, \tag{25}$$

is a type-II kernel and therefore, the process $\mathbf{y}$ is also GWSS. Equation (25) clarifies the relationship between the output spectrum and the input spectrum of a type-I kernel.

Example 2. (Filtered noise spectrum) If a white noise $\mathbf{w}$ with covariance $\boldsymbol{\Sigma}_{\mathbf{w}} = \mathbf{V}\mathbf{V}^H$ passes through a type-I kernel g (Figure 3b), then, the spectrum of the output process (which we denote by $\mathbf{x}$ in the figure) is given by $\gamma_{\mathbf{x}}(\mathbf{J}) = g(\mathbf{J})g(\mathbf{J})^H$; besides, its covariance matrix equals $\mathbf{V}g(\mathbf{J})g(\mathbf{J})^H\mathbf{V}^H$.

According to Property 8, each block in the spectrum of a GWSS process can be written as $\mathbf{R}\mathbf{R}^H$, where $\mathbf{R}$ is an upper-triangular matrix that may not necessarily be Toeplitz. Also, we saw in Example 2 that the spectrum of a GWSS process which is the output of a type-I kernel to a white noise has the form $g(\mathbf{J})g(\mathbf{J})^H$, where $g(\mathbf{J}_l)$ is upper-triangular and Toeplitz. A question that arises is whether every GWSS process is the output of a type-I kernel to a white noise.

Theorem 1. A necessary condition for the Hermitian and positive semi-definite matrix $\boldsymbol{\gamma} = [\gamma_{ij}]_{k\times k}$ to be equal to $\mathbf{O}\mathbf{O}^H$ where $\mathbf{O} = [o_{ij}]_{k\times k}$ is an upper-triangular and Toeplitz matrix, i.e., $\mathbf{O} = [o_{j-i}]_{k\times k}$ with $o_m = 0$ for $m < 0$, is that $\gamma_i \triangleq \gamma_{ii}$ shall be non-negative and non-increasing with respect to i, and for all $1 \le i \le j \le k-1$

$$\sqrt{(\gamma_i - \gamma_{i+1})(\gamma_j - \gamma_{j+1})} = |\gamma_{ij} - \gamma_{(i+1)(j+1)}|, \tag{26}$$

$$\sqrt{(\gamma_i - \gamma_{i+1})\gamma_k} = |\gamma_{ik}|. \tag{27}$$

(Proof is provided in Appendix D in the supplementary material).

Considering the matrix $\gamma_{\mathbf{x}}(\mathbf{J}_l)$ as the $\boldsymbol{\gamma}$ matrix in Theorem 1, we conclude that one of the necessary conditions for a GWSS process to be the output of a filter to a white noise is that the diagonal entries of $\gamma_{\mathbf{x}}(\mathbf{J}_l)$ must be non-increasing and furthermore, $\gamma_{\mathbf{x}}(\mathbf{J}_l)$ must satisfy conditions (26) and (27). Therefore, not all GWSS processes are obtained by filtering a white noise.

B. Graph Wiener filters

In [5], graph Wiener filters have been studied for undirected graphs. Also in [7] graph Wiener filters for undirected graphs have been employed to denoise images and graph signals. Figure 4 shows the block diagram of Wiener estimation; here, $\mathbf{w_s}$ is a white noise with covariance $\mathbf{V}\mathbf{V}^H$ and zero mean, and $\mathbf{w_n}$ is a GWSS noise with spectrum $\gamma_\mathbf{n}(\mathbf{J})$. As $\mathbf{x}$ is the output of the filter $s(\mathbf{S})$ to the input $\mathbf{w_s}$, it is also zero-mean. Thus we have:

$$\gamma_\mathbf{x}(\mathbf{J}) = s(\mathbf{J})s^H(\mathbf{J}). \tag{28}$$

The graph process $\mathbf{y}$ is obtained by adding the measurement noise $\mathbf{w_n}$ to the output of the measurement filter $h^{\mathrm{I}}(\mathbf{S})$ to the input $\mathbf{x}$. Hence, we can write:

$$\mathbf{y} = h^{\mathrm{I}}(\mathbf{S})\mathbf{x} + \mathbf{w_n}. \tag{29}$$

The noise $\mathbf{w_n}$ and the process $\mathbf{x}$ are assumed to be uncorrelated. We shall show in Theorems 3 and 4 that under certain conditions, the estimator of $\mathbf{x}$ given $\mathbf{y}$, i.e., $\bar{\mathbf{x}}|\mathbf{y}$, is obtained by using a graph filter $g(\mathbf{S})$ called the Wiener filter. Therefore,

$$\bar{\mathbf{x}} = g(\mathbf{S})\mathbf{y}. \tag{30}$$

By representing the estimation error via $\mathbf{e}$, the error corresponding to the l-th block can be written as:

$$\mathbf{e}_l = \mathbb{E}\{\|\widehat{\mathbf{x}}_l - \widehat{\widehat{\mathbf{x}}}_l\|^2\}, \quad \widehat{\widehat{\mathbf{x}}}_l = g(\mathbf{J}_l)\widehat{\mathbf{y}}_l. \tag{31}$$

1) Wiener optimization: Let us assume that we have a general linear filter $\mathbf{H}$ which may not necessarily be a type-I kernel, and

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{w_n}. \tag{32}$$

Theorem 2. If $\mathbf{x}$ is a GWSS and Gaussian process with the spectrum given by (28), i.e., $\mathbf{x} \sim \mathcal{N}(\mathbf{m_x}, \mathbf{V}s(\mathbf{J})s^H(\mathbf{J})\mathbf{V}^H)$, and $\mathbf{w_n}$ is white Gaussian noise with $\mathbf{w_n} \sim \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{V}\mathbf{V}^H)$, then, the following estimator is the MAP estimator for $\mathbf{x}|\mathbf{y}$ based on (32):

$$\bar{\mathbf{x}}|\mathbf{y} = \underset{\mathbf{x}}{\operatorname{argmin}} \|\mathbf{V}^{-1}(\mathbf{H}\mathbf{x} - \mathbf{y})\|_2^2 + \|\mathcal{W}(\mathbf{J})\mathbf{V}^{-1}(\mathbf{x} - \mathbf{m_x})\|_2^2, \tag{33}$$

where

$$\mathcal{W}(\mathbf{J}) = \sigma\, s(\mathbf{J})^{-1}, \tag{34}$$

if $s(\mathbf{J})$ is invertible. (Proof is provided in Appendix E in the supplementary material).

In the optimization problem (33), the matrix blocks $\mathcal{W}(\mathbf{J}_l)$ are the Fourier penalization coefficient matrices. Intuitively, the matrix blocks $\mathcal{W}(\mathbf{J}_l)$ penalize the low-SNR frequency components. Also, note that the optimization in (33) is convex.

Theorem 3. If $\mathbf{x}$ is GWSS with the spectrum given by (28) and $\mathbf{w_n}$ is a zero-mean white noise, i.e., $\gamma_\mathbf{n}(\mathbf{J}) = \sigma^2\mathbf{I}$, then, the problem (33) leads to the following linear MMSE estimator:

$$\bar{\mathbf{x}}|\mathbf{y} = \mathbf{\Sigma_{xy}}\mathbf{\Sigma_y}^{-1}\mathbf{y} + (\mathbf{I} - \mathbf{\Sigma_{xy}}\mathbf{\Sigma_y}^{-1}\mathbf{H})\mathbf{m_x}, \tag{35}$$

where

$$\begin{aligned}\mathbf{\Sigma_{xy}} &= \mathbf{V}s(\mathbf{J})s^H(\mathbf{J})\mathbf{V}^H\mathbf{H}^H = \mathbf{\Sigma_x}\mathbf{H}^H,\\ \mathbf{\Sigma_y} &= \mathbf{H}\mathbf{V}s(\mathbf{J})s^H(\mathbf{J})\mathbf{V}^H\mathbf{H}^H + \sigma^2\mathbf{V}\mathbf{V}^H\\ &= \mathbf{H}\mathbf{\Sigma_x}\mathbf{H}^H + \sigma^2\mathbf{V}\mathbf{V}^H.\end{aligned}$$

(Proof is provided in Appendix F in the supplementary material).

Suppose that $m_\mathbf{x} = 0$, the matrix $\mathbf{H}$ in (32) is a type-I kernel, i.e., $\mathbf{H} = h^{\mathrm{I}}(\mathbf{S}) = \mathbf{V}h^{\mathrm{I}}(\mathbf{J})\mathbf{V}^{-1}$, and $\mathbf{w_n}$ is the output of a type-I kernel denoted by $g_\mathbf{n}$ to a white noise. We are now interested in a different choice of $\mathcal{W}(\mathbf{J})$ which is block-diagonal with diagonal blocks of the same size as the diagonal blocks of $\mathbf{J}$. More precisely, we are looking for the optimal $\mathcal{W}(\mathbf{J})$ which minimizes the estimation error of $\mathbf{x}$. To introduce this $\mathcal{W}(\mathbf{J})$, we first define

$$\begin{aligned}&\mathbf{H}_l \triangleq h^{\mathrm{I}}(\mathbf{J}_l), \mathbf{S_x} \triangleq s(\mathbf{J}_l), \mathbf{G_n} \triangleq g_\mathbf{n}(\mathbf{J}_l),\\ &\mathbf{P} \triangleq \mathbf{H}_l^H\mathbf{H}_l, \mathbf{B} \triangleq \mathbf{S_x}\mathbf{S_x}^H = \boldsymbol{\gamma}_\mathbf{x},\\ &\mathbf{C} \triangleq \mathbf{H}_l^H\mathbf{G_n}\mathbf{G_n}^H\mathbf{H}_l = \mathbf{H}_l^H\boldsymbol{\gamma}_\mathbf{n}\mathbf{H}_l.\end{aligned} \tag{36}$$

We further let

$$\mathbf{T} \triangleq \mathbf{PBP} + \mathbf{C} = \mathbf{H}_l^H\mathbf{H}_l\mathbf{S_x}\mathbf{S_x}^H\mathbf{H}_l^H\mathbf{H}_l + \mathbf{H}_l^H\mathbf{G_n}\mathbf{G_n}^H\mathbf{H}_l, \tag{37}$$

$$\mathbf{E} \triangleq \mathbf{PB} + \mathbf{BP} = \mathbf{H}_l^H\mathbf{H}_l\mathbf{S_x}\mathbf{S_x}^H + \mathbf{S_x}\mathbf{S_x}^H\mathbf{H}_l^H\mathbf{H}_l.$$

Now, we define $\mathbf{Z}''$ as the solution to the following continuous Lyapunov equation [16]–[23]:

$$\mathbf{Z}''\mathbf{T} + \mathbf{T}\mathbf{Z}'' = \mathbf{E}, \tag{38}$$

which is given by[2]

$$vec(\mathbf{Z}'') = \mathbf{F}^{-1}\mathbf{g}, \tag{40}$$

$$\begin{aligned}\mathbf{F} &\triangleq \mathbf{T}^T \oplus \mathbf{T} = (\mathbf{PBP} + \mathbf{C})^T \oplus (\mathbf{PBP} + \mathbf{C})\\ &= \mathbf{I} \otimes (\mathbf{PBP} + \mathbf{C}) + (\mathbf{PBP} + \mathbf{C})^T \otimes \mathbf{I},\\ \mathbf{g} &\triangleq vec(\mathbf{E}) = vec(\mathbf{PB} + \mathbf{BP}),\end{aligned} \tag{41}$$

where $\oplus$ and $\otimes$ are Kronecker (Cartesian) sum and Kronecker (tensor) product operation symbols, respectively. We should highlight that $\mathbf{F}$ is invertible iff $\mathbf{T}$ is invertible. Now that we have $\mathbf{Z}''$, we define $\mathbf{Z}$ as

$$\mathbf{Z} \triangleq \mathbf{Z}''^{-1} - \mathbf{P}, \tag{42a}$$

and introduce $\mathbf{W}_l \triangleq \mathcal{W}(\mathbf{J}_l)$ via the solution to the following equation

$$\mathbf{W}_l^H\mathbf{W}_l = \mathbf{Z}. \tag{42b}$$

Note that if $\mathbf{Z}''^{-1} - \mathbf{P}$ is Hermitian and positive semi definite, the existence of $\mathbf{W}_l$ is guaranteed. Having obtained the desired $\mathbf{W}_l$, we are ready to use it in Theorem 4. In the proof of this theorem, we show that $\mathbf{W}_l$ obtained through the above procedure is optimal.

Corollary 2. In the special case where $\gamma_\mathbf{n}(\mathbf{J}_l) = \sigma^2\mathbf{I}$, $\mathcal{W}(\mathbf{J}_l) = \sigma(s(\mathbf{J}_l))^{-1}$ satisfies the Lyapunov equation (38).

[2]The solution to the Lyapunov equation (38) can also be obtained as follows: since $\mathbf{T}$ is Hermitian, it is diagonalizable and can be written as $\mathbf{Q}\mathbf{T}'\mathbf{Q}^H$ where $\mathbf{Q}$ is unitary and $\mathbf{T}' = diag(\delta_i)$ is diagonal with real entries. By letting $\mathbf{Y} \triangleq \mathbf{Q}^H\mathbf{Z}''\mathbf{Q}, \mathbf{K} \triangleq \mathbf{Q}^H\mathbf{E}\mathbf{Q}$, we can express the entries of $\mathbf{Y}$ as

$$y_{ij} = \frac{k_{ij}}{\delta_i + \delta_j}, \tag{39}$$

and $\mathbf{Z}'' = \mathbf{Q}\mathbf{Y}\mathbf{Q}^H$ will be obtained. More explanations are provided in Appendix J in the supplementary material.

(Proof is provided in Appendix H in the supplementary material).

Corollary 3. In a more general case, if $\boldsymbol{\gamma}_{\mathbf{x}}^{-1}\mathbf{H}_l^{-1}\boldsymbol{\gamma}_{\mathbf{n}}\mathbf{H}_l$ is Hermitian and positive semi definite, then, $\mathcal{W}(\mathbf{J}_l) = (\boldsymbol{\gamma}_{\mathbf{x}}^{-1}\mathbf{H}_l^{-1}\boldsymbol{\gamma}_{\mathbf{n}}\mathbf{H}_l)^{\frac{1}{2}}$ satisfies the Lyapunov equation (38). Intuitively, $\mathcal{W}$ is proportional to $\frac{1}{\sqrt{SNR}}$ here. (Proof is provided in Appendix I in the supplementary material).

Theorem 4. If $m_{\mathbf{x}} = 0$, $\mathbf{H}$ is a kernel of type-I, that is, $\mathbf{H} = h^{\mathrm{I}}(\mathbf{S}) = \mathbf{V}h^{\mathrm{I}}(\mathbf{J})\mathbf{V}^{-1}$, and $\mathbf{w_n}$ is the output of the type-I kernel $g_{\mathbf{n}}$ to some white input, then, the solution to (33) for the $\mathcal{W}$ introduced in (42b) minimizes the MSE error below

$$\mathbb{E}\big\{\|\widehat{\mathbf{e}}\|_2^2\big\} = \mathbb{E}\big\{\|\widehat{\bar{\mathbf{x}}} - \widehat{\mathbf{x}}\|_2^2\big\} = \mathbb{E}\big\{\sum_{i=1}^{N}|\widehat{\bar{\mathbf{x}}}[i] - \widehat{\mathbf{x}}[i]|^2\big\}. \quad (43)$$

In addition, the resulting $\bar{\mathbf{x}}$ is the output of a Wiener filter applied to $\mathbf{y}$:

$$\widehat{\bar{\mathbf{x}}}_l = \big(\mathcal{W}^H(\mathbf{J}_l)\mathcal{W}(\mathbf{J}_l) + h^{\mathrm{I}}(\mathbf{J}_l)^H h^{\mathrm{I}}(\mathbf{J}_l)\big)^{-1} h^{\mathrm{I}}(\mathbf{J}_l)^H\widehat{\mathbf{y}}_l. \quad (44)$$

(Proof of Theorem 4 is provided in Appendix G in the supplementary material).

Note that Wiener filters for undirected graphs in [5] are special cases of our Wiener filters for digraphs.

2) Solving the Wiener optimization: Below, we propose an algorithm for solving the Wiener optimization (33) using proximal splitting and FISTA methods [8], [9] similar to the technique proposed in [5]. The reason behind using the proximal gradient method is that if $s(\mathbf{J}_l)$ is close to singular for some l, i.e., $s(\lambda_m) \approx 0$, $\mathcal{W}(\mathbf{J})$ becomes ill-conditioned. To overcome this problem, (33) can be solved effectively using forward-backward splitting [9] and fast forward backward [10] methods leading to Algorithm 1 which is a fast and scalable implementation of our proposed method. We design our algorithm in the spectral domain; we first define $\mathbf{z} \triangleq \widehat{\bar{\mathbf{x}}} = \widehat{\mathbf{x}}$ ($m_{\mathbf{x}} = 0$ is assumed), and rewrite (33) as

$$\bar{\mathbf{z}}|\mathbf{y} = \underset{\mathbf{z}}{\operatorname{argmin}}\ \tfrac{1}{2}\|\mathbf{V}^{-1}(\mathbf{H}\mathbf{V}\mathbf{z} - \mathbf{y})\|_2^2 + \tfrac{1}{2}\|\mathcal{W}(\mathbf{J})\mathbf{z}\|_2^2. \quad (45)$$

To further proceed, we consider the first and second terms in the above objective function separately:

$$\begin{aligned}
f_1(\mathbf{z}) =& \tfrac{1}{2}\|\mathcal{W}(\mathbf{J})\mathbf{z}\|_2^2,\\
f_2(\mathbf{z}) =& \tfrac{1}{2}\|\mathbf{V}^{-1}(\mathbf{H}\mathbf{V}\mathbf{z} - \mathbf{y})\|_2^2\\
=& \tfrac{1}{2}(\mathbf{z}^H\mathbf{V}^H\mathbf{H}^H - \mathbf{y}^H)\mathbf{V}^{-H}\mathbf{V}^{-1}(\mathbf{H}\mathbf{V}\mathbf{z} - \mathbf{y})\\
=& \tfrac{1}{2}(\mathbf{z}^H\mathbf{V}^H\mathbf{H}^H\mathbf{V}^{-H}\mathbf{V}^{-1}\mathbf{H}\mathbf{V}\mathbf{z} - \mathbf{z}^H\mathbf{V}^H\mathbf{H}^H\mathbf{V}^{-H}\mathbf{V}^{-1}\mathbf{y}\\
&- \mathbf{y}^H\mathbf{V}^{-H}\mathbf{V}^{-1}\mathbf{H}\mathbf{V}\mathbf{z} + \mathbf{y}^H\mathbf{V}^{-H}\mathbf{V}^{-1}\mathbf{y}). \quad (46)
\end{aligned}$$

Now, if we define

$$\begin{aligned}
\mathbf{v}_j \triangleq& \mathbf{z}_j - \beta\nabla f_2(\mathbf{z}_j)\\
=& \mathbf{z}_j - \beta(\mathbf{V}^H\mathbf{H}^H\mathbf{V}^{-H}\mathbf{V}^{-1}\mathbf{H}\mathbf{V}\mathbf{z}_j - \mathbf{V}^H\mathbf{H}^H\mathbf{V}^{-H}\mathbf{V}^{-1}\mathbf{y})\\
=& \mathbf{z}_j - \beta\mathbf{V}^H\mathbf{H}^H\mathbf{V}^{-H}\mathbf{V}^{-1}(\mathbf{H}\mathbf{V}\mathbf{z}_j - \mathbf{y}), \quad (47)
\end{aligned}$$

Algorithm 1 Wiener optimization algorithm in vertex domain

Initialize:
  $\mathbf{x} = \mathbf{x}_1 = \mathbf{u}' = \mathbf{u}'_1,\ t_1 = 1$
loop
  Gradient step:
    $\mathbf{v}'_j = \mathbf{x}_j - \beta\mathbf{V}\mathbf{V}^H\mathbf{H}^H\mathbf{V}^{-H}\mathbf{V}^{-1}(\mathbf{H}\mathbf{x}_j - \mathbf{y})$,
  Proximal step:
    $\mathbf{u}'_{j+1} = g(\mathbf{S})\mathbf{v}'_j, g(\mathbf{J}_l) = (\beta\mathcal{W}(\mathbf{J}_l)^H\mathcal{W}(\mathbf{J}_l) + \mathbf{I})^{-1}$,
  FISTA scheme: $t_{j+1} = \frac{1+\sqrt{1+4t_j^2}}{2}$,
  Update step: $\mathbf{x}_{j+1} = \mathbf{u}'_{j+1} + \frac{t_j-1}{t_{j+1}}(\mathbf{u}'_{j+1} - \mathbf{u}'_j)$,
end loop
return $\mathbf{x}_{final}$

we can express the proximal operator as

$$\begin{aligned}
\mathbf{u}_{j+1} =& \operatorname{prox}_{\beta f_1}(\mathbf{v}_j) = \operatorname{prox}_{\frac{\beta}{2}\|\mathcal{W}(\mathbf{J})\mathbf{z}\|_2^2}(\mathbf{v}_j)\\
=& \underset{\mathbf{z}}{\operatorname{argmin}}\ \beta\|\mathcal{W}(\mathbf{J})\mathbf{z}\|_2^2 + \|\mathbf{z} - \mathbf{v}_j\|_2^2\\
=& \underset{\mathbf{z}}{\operatorname{argmin}}\ \left(\sum_l \beta\|\mathcal{W}(\mathbf{J}_l)\mathbf{z}_l\|_2^2 + \|\mathbf{z}_l - (\mathbf{v}_j)_l\|_2^2\right),
\end{aligned} \quad (48)$$

where j is the iteration index and l is the Jordan block index. Note that $\mathcal{W}(\mathbf{J})$ is block diagonal. The l-th term in the last equation can be rewritten as

$$\beta\mathbf{z}_l^H\mathcal{W}(\mathbf{J}_l)^H\mathcal{W}(\mathbf{J}_l)\mathbf{z}_l + \mathbf{z}_l^H\mathbf{z}_l - \mathbf{z}_l^H(\mathbf{v}_j)_l - (\mathbf{v}_j)_l^H\mathbf{z}_l + (\mathbf{v}_j)_l^H(\mathbf{v}_j)_l. \quad (49)$$

We conclude that the solution to the optimization (48) satisfies

$$\begin{aligned}
&\beta\mathcal{W}(\mathbf{J}_l)^H\mathcal{W}(\mathbf{J}_l)\mathbf{z}_l + \mathbf{z}_l - (\mathbf{v}_j)_l = \mathbf{0},\\
\Rightarrow\ &(\mathbf{u}_{j+1})_l = \mathbf{z}_l = (\beta\mathcal{W}(\mathbf{J}_l)^H\mathcal{W}(\mathbf{J}_l) + \mathbf{I})^{-1}(\mathbf{v}_j)_l, \quad (50)
\end{aligned}$$

which introduces a filtering equation in the spectral domain. Note that $\mathbf{z}_j, \mathbf{v}_j, \mathbf{u}_j$ are all in the spectral domain. Finally, by defining $\mathbf{v}'_j \triangleq \mathbf{V}\mathbf{v}_j$ and $\mathbf{u}'_{j+1} \triangleq \mathbf{V}\mathbf{u}_{j+1}$, the overall vertex domain optimization can be summarized as in Algorithm 1.

We should mention that for having a stopping condition in the loop for $\mathbf{x}_j$, we can set a threshold such as $0 < \epsilon$ for checking the relative variations, i.e., $\frac{\|\mathbf{x}_{j+1} - \mathbf{x}_j\|^2}{\|\mathbf{x}_j\|^2 + \delta} < \epsilon$.

Unlike (44), $\mathbf{H}$ may not be of type-I in Algorithm 1, but it solves (33) in the general case. In this case, (33) cannot be decomposed into smaller optimization problems. Thus, the final solution involves inverting an $N \times N$ matrix, which is computationally expensive. However, in Algorithm 1, the inverse in the proximal step is applied only to a matrix block which confirms its superiority.

V. Simulations and Experiments

To validate the previous theories, we provide experimental results on real data in this section. For this purpose, we first delineate how we create a digraph associated with the data. Next, we form an undirected graph based on the directed adjacency matrix, which shall be used in undirected Wiener optimization as a competing method [5, Algorithm 1]. We further build a symmetric Laplacian for the digraph using the random walk technique in [14]. This symmetric Laplacian matrix shall be used in both the Tikhonov minimization and the undirected Wiener optimization (due to its symmetry). Then, we elaborate how to define the stationarity level for the directed and undirected cases.

We use the meteorological dataset (specifically, the temperature and the humidity) released by the French national meteorological service, which is an hourly observation of the weather, collected during January 2014 in the region of Brest (France). As we shall show, the tools developed for digraphs demonstrate a better fit to the data compared to the undirected graph model. All experiments were performed using MATLAB, and the codes are made publicly available[3]. As a quality metric, we mainly rely on the signal to noise ratio (SNR) measure

$$\mathrm{SNR}(\mathbf{x}\,,\,\widehat{\mathbf{x}}) = -10\log\Big(\frac{\mathrm{var}(\mathbf{x}-\widehat{\mathbf{x}})}{\mathrm{var}(\mathbf{x})}\Big). \tag{51}$$

Below, we explain how the digraphs are built for each data type. To compare the performance of the developed techniques over digraph with the conventional undirected graph techniques, we use the nonsymmetric adjacency weighted matrix of the digraph ($\mathbf{W}_d$) to build a symmetric adjacency weighted matrix $\mathbf{W}_u = \frac{\mathbf{W}_d+\mathbf{W}_d^T}{2}$ corresponding to an undirected graph. As yet another method, we build a symmetric Laplacian matrix according to the suggested way in [14], and use it in both Tikhonov minimization and undirected Wiener optimization. To build such a Laplacian matrix, it is necessary for our drawn digraph to satisfy two conditions. First, the random walk matrix defined as

$$\mathbf{P} = \mathbf{D}_{\mathrm{out}}^{-1}\mathbf{W}_d, \tag{52}$$

where $\mathbf{D}_{\mathrm{out}}$ is the diagonal output degree matrix, needs to be irreducible, which means that each node is reachable from all other nodes. Our digraphs already have this property, but in general, we can add some edges to enforce this. The second condition is that $\mathbf{P}$ needs to be aperiodic. This condition is not difficult to meet: if $\mathbf{P}$ is not aperiodic, we can build an alternative matrix ($\tilde{\mathbf{P}}_a$) as:

$$\tilde{\mathbf{P}}_a = (1-a)\mathbf{P} + a\mathbf{I}, \quad a\in(0,1), \tag{53}$$

which is equivalent to adding self-loops with weight a.

When $\tilde{\mathbf{P}}_a$ is both irreducible and aperiodic, it is assured to have a unique stationary distribution $\boldsymbol{\pi}$. In particular, $\boldsymbol{\pi}\tilde{\mathbf{P}}_a = \boldsymbol{\pi}$, with $\boldsymbol{\pi}$ consisting of non-negative elements summing up to 1. By setting $\mathbf{\Pi} \triangleq diag(\boldsymbol{\pi})$, we can form the symmetric Laplacian matrix for a digraph as

$$\mathbf{L}_d = \mathbf{\Pi} - \frac{\mathbf{\Pi}\tilde{\mathbf{P}}_a+\tilde{\mathbf{P}}_a^T\mathbf{\Pi}}{2}. \tag{54}$$

A. Stationarity level

Since we model some real data with stationary graph processes (either directed graphs or undirected), we need to measure how well the data fits into our model. More specifically, we evaluate to what extent the data can be assumed stationary (leading to the concept of stationarity level).

We explained in Example 2 that the signal $\mathbf{x}$ obtained by passing a white noise with spectrum $\gamma(\mathbf{J}) = \mathbf{I}$ through a type-I kernel s with $s(\mathbf{S}) \triangleq \mathbf{V}s(\mathbf{J})\mathbf{V}^{-1}$, is GWSS. Here, $s(\mathbf{J})$ is a block diagonal matrix, and the covariance of the output signal is $\mathbf{V}s(\mathbf{J})s(\mathbf{J})^H\mathbf{V}^H$.

When dealing with real data, we have access to the (approximate) covariance matrix. If the data were the outcome of passing a white noise through a type-I kernel $s(\mathbf{S})$, the covariance of the data would have been $\mathbf{\Sigma}_{\mathbf{x}} = \mathbf{V}s(\mathbf{J})s(\mathbf{J})^H\mathbf{V}^H$. A measure of stationarity is to check whether $\mathbf{V}^{-1}\mathbf{\Sigma}_{\mathbf{x}}\mathbf{V}^{-H}$ (which would be equal to $s(\mathbf{J})s(\mathbf{J})^H$) is block diagonal with diagonal blocks matching those of $\mathbf{J}$. Let $\tilde{s}(\mathbf{J})\tilde{s}(\mathbf{J})^H$ be the block diagonal projection of $\mathbf{V}^{-1}\mathbf{\Sigma}_{\mathbf{x}}\mathbf{V}^{-H}$, which is formed by eliminating the elements which are not on the diagonal blocks according to the diagonal blocks of $\mathbf{J}$. Now, we define

$$\mathrm{StationarityLevel} = \frac{\|\tilde{s}(\mathbf{J})\tilde{s}(\mathbf{J})^H\|}{\|\mathbf{V}^{-1}\mathbf{\Sigma}_{\mathbf{x}}\mathbf{V}^{-H}\|}. \tag{55}$$

The stationarity level for undirected graphs defined in [5] is a special case of (55) when the blocks are all of size 1 (only the main diagonal is kept in $\tilde{s}$). In Table I, we have reported the stationarity levels of both temperature and humidity data treated as processes over graphs/digraphs by considering the three main graph shift operators.

B. Graph signal denoising on temperature data

In our first experiment, we consider the temperature of the stations; in particular, we focus on a subset of 16 stations to form a graph with 16 nodes. To form the edges, we recall that the temperature data shall vary smoothly over the graph. In other words, graph neighbors shall have similar temperature values. Thus, we consider the 744 recorded values at each station, as a feature vector to that node (station). Then, we remove the temperature mean from each feature vector, which is equivalent to removing the first moment. Next, we define the distance between two nodes as the norm of the difference between the two feature vectors. For each node (station) in the graph, we consider the 10 nearest nodes and form a directed edge between them. The direction of the edge is from the node with smaller latitude to the node with higher latitude as in [6], [28], [29]. The weight of the edges is also calculated as $w_{ij} = \mathrm{e}^{-\tau d^2}$, where d is the defined distance, and τ is a tuning parameter adjusted in a way that the average of the input and output degrees of all nodes becomes 0.5. The acquired graph and the corresponding adjacency matrix can be seen in Figure 5. To apply a denoising task, we add noise to the recordings with the covariance $\sigma^2\mathbf{V}\mathbf{V}^H$. Here, $\mathbf{V}$ is the Jordan basis of the acquired digraph and σ is adjusted to achieve a desired SNR value. For every input SNR, we run the simulation 20000 times by choosing a random signal from the dataset at each time and then, we average the output SNR for each method of denoising. The results of denoising are shown in Figure 5. While the Tikhonov method consistently exhibits the worst performance among the considered methods,

[3]https://github.com/mohammadeinyafm/WSS_Processes_On_Directed_Graphs.

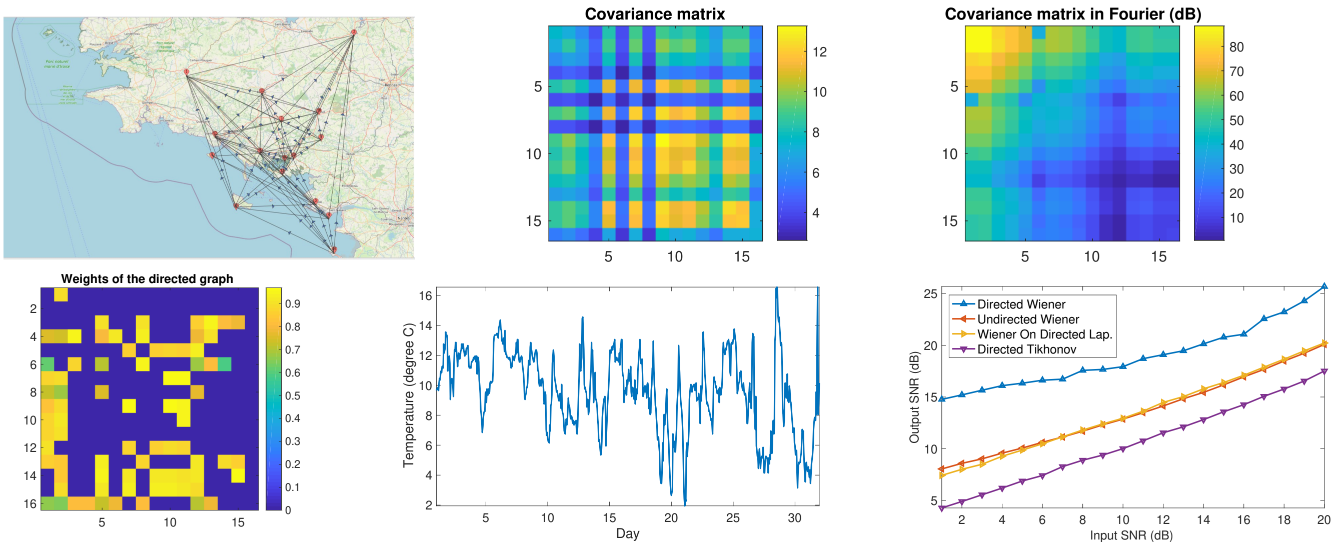

Fig. 5. (Top left) Digraph of France meteorological stations for temperature data. (Top middle and right) The covariance matrices of the data in vertex and Fourier domains. (Bottom left) The weight matrix of the digraph. (Bottom middle) A realization of the signal (first measure). (Bottom right) Output SNRs of different methods in dB.

TABLE I
STATIONARITY LEVELS OF TEMPERATURE AND HUMIDITY DATA (FRANCE METEOROLOGICAL STATIONS) OVER THE GRAPHS/DIGRAPHS BY CONSIDERING VARIOUS GRAPH SHIFT MATRICES.

Used method	Graph shift operator matrix	Stationarity percentage	
		Temperature	Humidity
Proposed digraph	$\mathbf{W}_d$	0.99	1
Undirected graph [5]	$\mathbf{W}_u$	0.97	0.9848
Symmetric Laplacian [14]	$\mathbf{L}_d$	0.94	0.9846

the two denoising methods based on the undirected graph and the symmetric Laplacian case have nuance differences in all SNRs. It can also be observed that the proposed method outperforms the rest, confirming that our stationarity definition is appropriate and the digraph provides a better model for this particular data, especially in lower SNRs, assuming digraph-based stationary noise. Another observation is that the output SNR for the undirected and symmetric Laplacian methods tend to the the input SNR at large SNR values, which implies that the denoising gain is vanishing. In contrast, the proposed method maintains roughly a 4dB denoising gain.

C. Inpainting on humidity data

Our next experiment will be the estimation of missing samples on the humidity data. To further explore the robustness of the methods relative to the graph topology, this time we choose 10 stations and connect each node (station) to 4 nearest nodes using the same strategy as explained for the temperature data. However, in contrast to the graph for the temperature data, the direction of each edge is determined so that the humidity data can be considered stationary on the digraph with a stationarity measure greater than 95%. The obtained digraph, its weight matrix and the covariance matrix of the humidity data are shown in Figure 6. For the purpose of comparison, we again form the undirected graph and the symmetric Laplacian matrix as explained earlier. In this scenario, our goal is to predict the humidity of 40% of the stations (which are masked) based on the noisy measurements at the remaining stations. For each input SNR, we ran the simulation 20000 times by choosing a random signal at each time, then, we averaged the output SNRs. For the third experiment, the input data and graphs are the same as the second experiment; but this time, the input SNR is always 14dB and the percentage of the data that mask covers, varies from 10% to 90%. The results of output SNRs for different methods for both experiments are illustrated in Figure 6. While the Tikhonov method consistently marks the worst performance, at lower input SNRs, the undirected and symmetric Laplacian methods perform almost equally. At higher SNRs, however, the undirected method takes the lead with around 1dB gain. Interestingly, our proposed method provides the best performance at all SNR values, particularly, with a considerable gap at the lower SNR regime. At fixed input SNR, we again observe that our method outperforms the rest at all fractions of kept data. This confirms that the proposed digraph model (accompanied with Wiener filtering) is a better fit for these types of data.

D. Further experiments

We have additionally considered two more scenarios using the same meteorological dataset. Firstly, in denoising the temperature, we have built a 5 nearest neighbours graph with 16 nodes that has a rather good stationarity level for both undirected and directed cases. The results are shown in Figure 7 showing that our method is effective even on sparse graphs. Secondly, using the same graph and by changing the noise to be white in the classical (undirected) sense, as can be seen in Figure 8, the techniques based on both undirected and directed Wiener optimizations do not provide any gain in the denoising task which is a limitation.

VI. CONCLUSION

In this work, after reviewing the notion of GWSS processes on undirected graphs, we generalized this concept to digraphs. We classified and studied the kernels on digraphs leading to GWSS processes into two categories, namely type-I and type-II. We further provided the generalization of graph Wiener

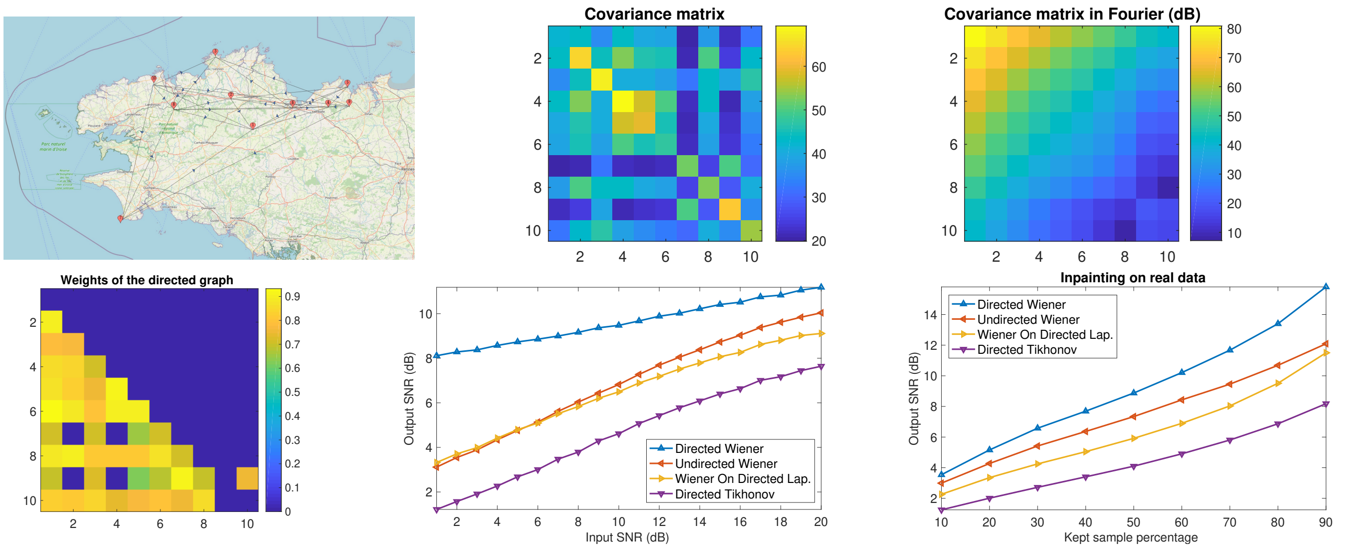

Fig. 6. (Top left) The digraph for humidity data. (Top middle and right) The covariance matrices of the data in vertex and Fourier domains. (Bottom left) The graph weight matrix. (Bottom middle and right) Output SNRs for different methods for (middle) different input SNRs, saved percentage = 60% and (right) $\mathrm{SNR}_{\mathrm{in}} = 14$dB.

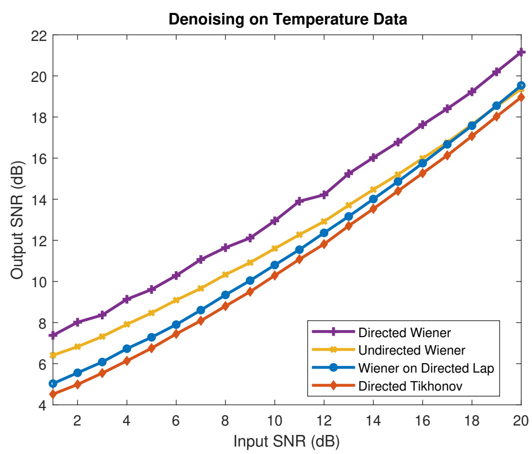

Fig. 7. Output SNRs of different methods in dB for the sparse graph scenario.

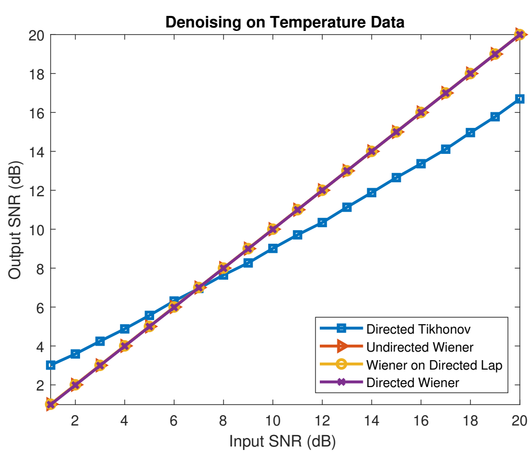

Fig. 8. Output SNRs of different methods in dB for the classical white noise scenario.

filters and Wiener optimization framework for digraphs and showed that for Gaussian processes, the framework simplifies to a MAP estimation; for non-Gaussian processes, instead, it leads to the LMMSE estimator. We showed that for type-I kernels, the framework is associated with a Lyapunov matrix equation and its solution minimizes the MSE. For digraphs, we solved the Wiener optimization framework using the proximal splitting method. We performed denoising and inpainting experiments on France temperature and humidity data and showed that these real data better fit into the framework of stationary processes on digraphs (in contrast to undirected graphs) specially for the considered noise type.

References

[1] L. Stanković, M. Daković, and E. Sejdić, "Introduction to graph signal processing," in *Vertex-Frequency Analysis of Graph Signals*. Springer, 2019, pp. 3–108.

[2] B. Girault, "Stationary graph signals using an isometric graph translation," in *23rd European Signal Processing Conference (EUSIPCO)*. IEEE, 31 August–4 September, 2015, pp. 1516–1520.

[3] B. Girault, P. Gonçalves, E. Fleury, and A. S. Mor, "Semi-supervised learning for graph to signal mapping: A graph signal wiener filter interpretation," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, May 4–9, 2014, pp. 1115–1119.

[4] B. Girault, P. Gonçalves, and E. Fleury, "Signaux stationnaires sur graphe: étude d'un cas réel," in *Gretsi*, September 8–11, 2015, p. 4.

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

[6] A. G. Marques, S. Segarra, and G. Mateos, "Signal processing on directed graphs: The role of edge directionality when processing and learning from network data," *IEEE Signal Processing Magazine*, vol. 37, no. 6, pp. 99–116, 2020.

[7] A. C. Yağan and M. T. Özgen, "Spectral graph based vertex-frequency wiener filtering for image and graph signal denoising," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 6, pp. 226–240, 2020.

[8] 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.

[9] 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.

[10] C. Zheng, C. Cheng, and Q. Sun, "Wiener filters on graphs and distributed polynomial approximation algorithms," *arXiv preprint arXiv:2205.04019*, 2022.

[11] G. Sagi and T. Routtenberg, "MAP estimation of graph signals," *IEEE Transactions on Signal Processing*, vol. 72, pp. 463–479, 2024.

[12] A. Kroizer, T. Routtenberg, and Y. C. Eldar, "Bayesian estimation of graph signals," *IEEE Transactions on Signal Processing*, vol. 70, pp. 2207–2223, 2022.

[13] A. Sandryhaila and J. M. Moura, "Discrete signal processing on graphs," *IEEE Transactions on Signal Processing*, vol. 61, no. 7, pp. 1644–1656, 2013.

[14] H. Sevi, G. Rilling, and P. Borgnat, "Harmonic analysis on directed graphs and applications: From Fourier analysis to wavelets," *Applied and Computational Harmonic Analysis*, vol. 62, pp. 390–440, 2023.

[15] J. A. Deri and J. M. Moura, "Spectral projector-based graph Fourier transforms," *IEEE Journal of Selected Topics in Signal Processing*, vol. 11, no. 6, pp. 785–795, 2017.

[16] P. C. Parks, "A. M. Lyapunov's stability theory—100 years on," *IMA Journal of Mathematical Control and Information*, vol. 9, no. 4, pp. 275–303, 1992.

[17] V. Simoncini, "Computational methods for linear matrix equations," *SIAM Review*, vol. 58, no. 3, pp. 377–441, 2016.

[18] R. H. Bartels and G. W. Stewart, "Solution of the matrix equation AX+XB=C [F4]," *Communications of the ACM*, vol. 15, no. 9, pp. 820–826, 1972.

[19] J. J. Sylvester, "Sur l'équation en matrices px=xq," *C.R. Acad. Sci. Paris*, vol. 99, no. 2, pp. 67–71, 1884.

[20] R. Bhatia and P. Rosenthal, "How and why to solve the operator equation AX-XB=Y," *Bulletin of the London Mathematical Society*, vol. 29, no. 1, pp. 1–21, 1997.

[21] A. Dmytryshyn and B. Kågström, "Coupled Sylvester-type matrix equations and block diagonalization," *SIAM Journal on Matrix Analysis and Applications*, vol. 36, no. 2, pp. 580–593, 2015.

[22] S.-G. Lee and Q.-P. Vu, "Simultaneous solutions of Sylvester equations and idempotent matrices separating the joint spectrum," *Linear Algebra and its Applications*, vol. 435, no. 9, pp. 2097–2109, 2011.

[23] G. Birkhoff and S. MacLane, *A Survey of Modern Algebra*. Macmillan, New York, 1965.

[24] A. G. Marques, S. Segarra, G. Leus, and A. Ribeiro, "Stationary graph processes and spectral estimation," *IEEE Transactions on Signal Processing*, vol. 65, no. 22, pp. 5911–5926, 2017.

[25] S. Segarra, S. P. Chepuri, A. G. Marques, and G. Leus, "Statistical graph signal processing: Stationarity and spectral estimation," in *Cooperative and Graph Signal Processing*. Elsevier, 2018, pp. 325–347.

[26] M. B. Iraji, M. Eini, A. Amini, and S. Rini, "Stationary processes on directed graphs," in *12th Iran Workshop on Communication and Information Theory (IWCIT)*. IEEE, May 1–2, 2024, pp. 1–5.

[27] N. J. Higham, *Functions of matrices: Theory and computation*. Philadelphia, PA, USA: SIAM, 2008.

[28] S. Furutani, T. Shibahara, M. Akiyama, K. Hato, and M. Aida, "Graph signal processing for directed graphs based on the Hermitian Laplacian," in *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML*. Springer, September 16–20, 2019, pp. 447–463.

[29] R. Shafipour, A. Khodabakhsh, G. Mateos, and E. Nikolova, "A directed graph Fourier transform with spread frequency components," *IEEE Transactions on Signal Processing*, vol. 67, no. 4, pp. 946–960, 2018.

[30] D. Dereniowski and M. Kubale, "Cholesky factorization of matrices in parallel and ranking of graphs," in *Parallel Processing and Applied Mathematics: 5th International Conference*. Springer, September 7–10, 2003, pp. 985–992.

[31] G. H. Golub and C. F. Van Loan, *Matrix computations*, 3rd ed. Johns Hopkins University Press, 1996.

[32] R. A. Horn and C. R. Johnson, *Matrix analysis*. Cambridge University Press, 2012.

[33] L. N. Trefethen and D. Bau, *Numerical linear algebra*. SIAM, 2022.

[34] J. Stoer and R. Bulirsch, *Introduction to numerical analysis*. Springer, 2002.

[35] G. Strang, *Linear algebra and learning from data*. Wellesley-Cambridge Press, 2019.

[36] M. A. Woodbury, *Inverting modified matrices*. Department of Statistics, Princeton University, 1950.

[37] L. Guttman, "Enlargement methods for computing the inverse matrix," *The Annals of Mathematical Statistics*, pp. 336–343, 1946.

[38] W. W. Hager, "Updating the inverse of a matrix," *SIAM Review*, vol. 31, no. 2, pp. 221–239, 1989.

[39] N. J. Higham, *Accuracy and stability of numerical algorithms*. Philadelphia, PA, USA: SIAM, 2002.

[40] K. B. Petersen and M. S. Pedersen, "The matrix cookbook," Technical University of Denmark, November 2008.

[41] M. Brookes, "The matrix reference manual," Imperial College London, 2022.

Biographies

Mohammad Bagher Iraji is a Ph. D. student at Sharif University of Technology. He received his B. Sc. and M. Sc. degrees in Electrical Engineering (Communication Systems) both from Sharif University of Technology in 2007 and 2009. He also received another M. Sc. degree in Electrical Engineering from California Institute of Technology in 2011. In 2017, he joined I. R. of Iran Meteorological Organization as a telecommunications expert. He is interested in graph signal processing.

Mohammad Eini is a Ph. D. student at Michigan State University. He received his B. Sc. from Iran University of Science and Technology in Electrical Engineering (Communication Systems) and his M. Sc. in Signal Processing from Sharif University of Technology. He is interested in machine learning algorithms and graph signal processing.

Arash Amini received the B.Sc., M.Sc., and Ph.D. degrees in electrical engineering (communications and signal processing) and the B.Sc. degree in petroleum engineering (reservoir) from the Sharif University of Technology, Tehran, Iran, in 2005, 2007, 2011, and 2005, respectively. He was a Researcher with the École Polytechnique fédérale de Lausanne, Lausanne, Switzerland, from 2011 to 2013, working on statistical approaches toward modeling sparsity in continuous-domain. He joined Sharif University of Technology as an assistant professor in 2013, where he is now an associate professor since 2018. He has served as an associate editor of IEEE Signal Processing Letters from 2014 to 2018. His research interests include various topics in the field of signal processing.