FREQUENCY ANALYSIS OF TIME-VARYING GRAPH SIGNALS

Andreas Loukas *Damien Foucard*

Department of Telecommunication Systems, TU Berlin

ABSTRACT

This paper focuses on the harmonic analysis of graph signals that evolve with time. Our goal is to generalize and, in fact, unify the familiar concepts from time- and graph-frequency analysis. To this end, we study the properties of a *joint* time and graph Fourier transform (JFT) and the associated notion of variation. We build on our results to create filters which act on the joint (time and graph) frequency domain, and provide an example of how joint filters can improve signal recovery on time-varying graph signals with non-separable temporal and graph frequency structure.

1. INTRODUCTION

The availability of complex and high-dimensional datasets has spurred the need for new data analysis methods. One prominent research direction in signal processing has been the focus on data supported over graphs [17]. Graph signals, i.e., signals taking values on the vertices of combinatorial graphs, represent a convenient solution to model data exhibiting complex and non-uniform properties, such as those found in social, biological, and transportation networks, among others. Arguably, the most fundamental tool in the analysis of graph signals is the graph Fourier transform (GFT) [16, 17, 18]. In an analogous manner to the discrete Fourier transform (DFT), using GFT one may examine graph signals in the graph frequency domain, and, for instance, remove noise by attenuating high graph-frequencies. GFT has also lead to significant new insights in problems such as smoothing and denoising [8, 19, 23], segmentation [7], sampling and approximation [10, 25, 26], and classification [2, 20, 24] of graph data.

Yet, for many modern graph datasets, *time* is still of the essence. Whether we are interested in which candidate is more popular to whom in the political blogosphere [1], how an infection spreads over the global transportation network [21], or what the average daily traffic over the streets of a city is [9], the graph signals one encounters are not only a function of the underlying graph—they also evolve with time. Motivated by this need, this paper considers the frequency analysis of graph signals that change with time, referred to as *time-varying graph signals*. Our goal is to generalize and, in fact, unify the familiar concepts from time- and graph-frequency analysis so as to jointly consider graph and temporal aspects of data.

To this end, we advocate for a *joint (time and graph) Fourier transform* (JFT), constructed by taking the graph Fourier and discrete Fourier transforms jointly. The idea of JFT emerges naturally as a consequence of two recent works using the Z-transform for the analysis of the temporal dynamics of graph signals, that by Sandryhaila et al. [16] and our previous work on the behavior of graph filters under time-variations [6]. Building upon recent progress, this paper aims to expand our understanding of the joint time-graph transform. After examining the properties of JFT and its relations with standard Fourier transforms (see Section 2.1), we propose a generalization of the notion of smoothness (captured by the JFT basis) appropriate for time-varying graph signals (see Section 2.2). Our exposition abstracts from the matrix representation of a graph, such as the discrete and normalized Laplacian or the adjacency matrix. We also discuss the implications of choosing a directed graph in the joint transform.

Based on the JFT, one may design *joint filters* which selectively attenuate or amplify certain joint-frequencies of a time-varying graph signal [4, 6, 16]. We propose a filtering algorithm, which, improving previous work, (*i*) is distributed with linear complexity in the number of edges and time instances [16], and (*ii*) approximates a larger set of joint frequency responses [6]. In addition, we demonstrate how joint filters can be used to optimally solve a signal recovery problem, where given the statistical properties of a desired and interfering time-varying graph signal, one is asked to design the filter which recovers the original signal with the smallest mean-squared error. As an example, we consider the denoising of a wave equation on the graph and find that the joint approach achieves a significant increase in accuracy compared to disjoint graph/time denoising algorithms.

2. JOINT FOURIER TRANSFORM

Consider a weighted graph $\mathcal{G} = (V, E, W)$ of N vertices $u_1, \ldots, u_N$ and M edges, and suppose that we are given a periodic time-varying graph signal represented by matrix $\boldsymbol{X} \in \mathbb{C}^{N \times T}$, with X_{nt} being the value of vertex u_n at time instant t. The discrete Fourier transform on each row of $\boldsymbol{X}$ gives the

frequency representation of our signal

$$\text{DFT}\{\boldsymbol{X}\} = \boldsymbol{X}\boldsymbol{V}_T^\top, \quad (1)$$

with the unitary DFT matrix $\boldsymbol{V}_T$ having entries $[\boldsymbol{V}_T]_{t_1,t_2} = \exp(-\frac{2\pi j(t_1-1)(t_2-1)}{T})/\sqrt{T}$ for $t_1, t_2 = 1, \ldots, T$. However, since $\boldsymbol{V}_T^\top$ acts on each row of $\boldsymbol{X}$ independently, it overlooks the graph structure of our data. Similarly, applying the graph Fourier transform in parallel [17] for each time-instant as

$$\text{GFT}\{\boldsymbol{X}; \mathcal{G}\} = \boldsymbol{V}_G\boldsymbol{X}, \quad (2)$$

where $\boldsymbol{V}_G$ is the $N \times N$ left eigenvector matrix of a matrix representation of $\mathcal{G}$ (such as the Laplacian $\boldsymbol{L}_G$, normalized Laplacian $\boldsymbol{N}_G$, or the adjacency matrix $\boldsymbol{W}_G$) lets us take into account the variation of the signal with respect to the graph, but neglects the temporal aspect of data.

To capture the frequency content of $\boldsymbol{X}$ along both time and graph domains, one has to apply both transforms *jointly*, implying a joint graph and time Fourier transform defined as

$$\text{JFT}\{\boldsymbol{X}; \mathcal{G}\} := \boldsymbol{V}_G\boldsymbol{X}\boldsymbol{V}_T^\top. \quad (3)$$

It might be more convenient to express JFT as a matrix vector multiplication. Exploiting the properties of the Kronecker product ($\otimes$), we can write $\text{JFT}\{\boldsymbol{x}; \mathcal{G}\} = (\boldsymbol{V}_T \otimes \boldsymbol{V}_G)\,\boldsymbol{x} = \boldsymbol{V}_J\boldsymbol{x}$, where in the last step we set $\boldsymbol{V}_J = \boldsymbol{V}_T \otimes \boldsymbol{V}_G$.

We note that the definitions above are independent of the matrix representation of $\mathcal{G}$ and can be used in conjunction with each definition of GFT.

2.1. The properties of JFT

Before delving into the consistency of JFT with GFT and DFT, we examine two of its basic properties. To start with, let us simplify the notation by setting $\boldsymbol{U}_G = \boldsymbol{V}_G^{-1}$ and $\boldsymbol{U}_T = \boldsymbol{V}_T^*$. With this notation in place, we can express the inverse joint Fourier transform.

Property 1. JFT is an invertible transform. Suppose that $\boldsymbol{y} = vec(\boldsymbol{Y}) = JFT\{\boldsymbol{x}, \mathcal{G}\}$. The inverse transform in matrix and vector form is

$$JFT^{-1}\{\boldsymbol{Y}, \mathcal{G}\} = \boldsymbol{U}_G\boldsymbol{Y}\boldsymbol{U}_T^\top \;\; and \;\; JFT^{-1}\{\boldsymbol{y}, \mathcal{G}\} = \boldsymbol{U}_J\,\boldsymbol{y}, \quad (4)$$

respectively, where $\boldsymbol{U}_J = \boldsymbol{U}_T \otimes \boldsymbol{U}_G$.

As shown next, the unitarity of the joint transform depends on the type of matrix representation of a graph chosen.

Property 2. JFT is unitary if and only if GFT is unitary.

Proof. From definition, we have $\boldsymbol{V}_J\boldsymbol{V}_J^* = (\boldsymbol{V}_T \otimes \boldsymbol{V}_G)(\boldsymbol{V}_T \otimes \boldsymbol{V}_G)^* = (\boldsymbol{V}_T\boldsymbol{V}_T^*) \otimes (\boldsymbol{V}_G\boldsymbol{V}_G^*) = \boldsymbol{I}_T \otimes (\boldsymbol{V}_G\boldsymbol{V}_G^*)$. JFT is unitary if last statement is to $\boldsymbol{I}_{NT}$ (i.e., an identity matrix of dimension NT) it must be that $\boldsymbol{V}_G\boldsymbol{V}_G^* = \boldsymbol{I}_N$, which is equivalent to asserting that GFT is unitary. □

We deduce that JFT is a unitary transform (i.e., the columns of $\boldsymbol{U}_J$ form an orthonormal basis) for all symmetric matrix representations of a graph, such as the Laplacian or adjacency matrix, as long as the graph is undirected. When the graph is directed, unitarity is lost. For clarity, in the rest of this paper we opt to work with undirected graph $\mathcal{G}$ and only express our results w.r.t. the Laplacian matrix. Still, all results are directly applicable to alternative matrix representations.

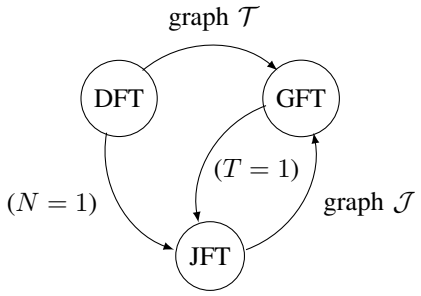

Fig. 1. Relations between Fourier transforms. Each directed arrow (say from A to B) in the figure should be interpreted as a generalization (A can be simulated by B). Edge annotations hint on the simulation method.

Consistency between transforms. Fig. 1 characterizes the relations between DFT, GFT, and JFT of $\boldsymbol{X}$. Each directed arrow (e.g., from A to B) in the figure should be interpreted as a transform-simulation (transform A can be simulated by B). The equivalence between GFT and JFT is illustrated as a bidirectional simulation. Let us begin from the obvious relations. By definition, both DFT and GFT are specific cases of JFT. In particular, $\text{DFT}\{\boldsymbol{X}\} = \text{JFT}\{\boldsymbol{X}; \mathcal{G}\}$ if the graph consists of a single vertex ($N = 1$) and $\text{GFT}\{\boldsymbol{X}; \mathcal{G}\} = \text{JFT}\{\boldsymbol{X}; \mathcal{G}\}$ if $\boldsymbol{X}$ does not change in the time domain ($T = 1$).

We proceed with the remaining two relations.

DFT → GFT. Though this is a generalization of known results [13, 15], it helps to illustrate the concept of transform equivalence. To establish the relation, we identify a graph $\mathcal{T} = (V_T, E_T)$, such that $\text{DFT}\{\boldsymbol{X}\} = \text{GFT}\{\boldsymbol{X}; \mathcal{T}\}$. To obtain $\mathcal{T}$, think of time as a ring graph consisting of T nodes. In other words, each vertex $u_t \in V_T$ is connected to vertex u_{t+1} for $t = 1, \ldots, T$, with index $T+1 = 1$. The adjacency matrix of $\mathcal{T}$ has $\boldsymbol{V}_T$ and $\boldsymbol{U}_T$ as left and right eigenvector matrices, and as eigenvalues $\lambda_T(t) = \exp((2\pi i(t-1)(T-1))/T)$. Furthermore, since $\boldsymbol{L}_T = \boldsymbol{I}_T - \boldsymbol{A}_T$, the Laplacian has the same spectrum (up to translation and reordering), rendering the choice of representation (i.e., $\boldsymbol{A}_T$ or $\boldsymbol{L}_T$ or $\boldsymbol{N}_T$) arbitrary.

JFT → GFT. We will simulate JFT by applying GFT on the joint graph $\mathcal{J} = (V_J, E_J)$, effectively showing that $\text{JFT}\{\boldsymbol{X}; \mathcal{G}\} = \text{GFT}\{\boldsymbol{X}; \mathcal{J}\}$. We construct $\mathcal{J}$ as the graph cartesian product of $\mathcal{G}$ and $\mathcal{T}$. The joint graph consists of T copies of $\mathcal{G}$, denoted by $\mathcal{G}_t = (V_t, E_t)$, one for each time-instant, with $V_J = V_1 \cup \ldots \cup V_T$. Name the corresponding vertices in each copy as $u_{n,t} \in V_t$. In addition to the $T \times M$ edges already introduced, the joint graph contains $T \times N$ extra edges joining consecutive copies: in particular, for each vertex $u_{n,t}$ in $\mathcal{G}_t$ the joint graph has a *directed* edge to vertex $u_{n,t+1}$ in $\mathcal{G}_{t+1}$ (modulo T). The Laplacian matrix[1] of $\mathcal{J}$ is

[1]The argument is identical for the adjacency and normalized Laplacian matrix representations.

expressed as

$$\boldsymbol{L}_J = \boldsymbol{I}_T \otimes \boldsymbol{L}_G + \boldsymbol{L}_T \otimes \boldsymbol{I}_N = \boldsymbol{L}_T \oplus \boldsymbol{L}_G, \quad (5)$$

where $(\oplus)$ is the knonecker sum operator. Even though $\boldsymbol{L}_J$ is not a symmetric matrix (due to $\mathcal{T}$ and $\mathcal{J}$ being directed) it follows from Theorem 13.16 in [5] that $\boldsymbol{L}_J$ has eigendecomposition $\boldsymbol{L}_J = (\boldsymbol{U}_T \otimes \boldsymbol{U}_G)(\boldsymbol{\Lambda}_T \oplus \boldsymbol{\Lambda}_G)(\boldsymbol{V}_T \otimes \boldsymbol{V}_G) = \boldsymbol{U}_J \boldsymbol{\Lambda}_J \boldsymbol{V}_J$ which fulfills our requirement.

2.2. Joint signal variation

The utility of a transform stems largely from its ability to provide insight about data. For instance, by observing the GFT of a graph signal one gains intuition about the *variation* of a signal over the graph, a notion which characterizes how aggressively a signal is changing on the graph. Therefore, GFT is useful because it allows us to distinguish smooth signals from non-smooth ones. In a similar manner, to render JFT a useful transform, we must give it insightful meaning.

We propose to use the relation $\text{JFT}\{\boldsymbol{X}; \mathcal{G}\} = \text{GFT}\{\boldsymbol{X}; \mathcal{J}\}$ to imbue JFT with an appropriate notion of smoothness. In this way the variation of a time-varying graph signal $\boldsymbol{X}$ on $\mathcal{G}$ is defined to be equal to the variation of the same signal (interpreted now as a graph signal) on the joint graph $\mathcal{J}$.

Consider a vertex $u_{n_i,t} \in V_t \subseteq V_J$ and denote by $n_j \overset{\mathcal{J}}{\sim} n_i$ its neighbors in $\mathcal{J}$. Using the definition of variation [17] on $\mathcal{J}$, we define the *local variation* of the time-varying graph signal $\boldsymbol{X}$ at the n_i-th vertex at time t to be

$$\|\nabla_{n_i,t} \boldsymbol{X}\|_2 := \left[\sum_{n_j \overset{\mathcal{J}}{\sim} n_i} \left(\frac{\partial \boldsymbol{X}}{\partial e_{n_i n_j}} \right)^2 \right]^{\frac{1}{2}}$$

$$= \left[\sum_{n_j \overset{\mathcal{G}}{\sim} n_i} \big(X_{n_j,t} - X_{n_i,t}\big)^2 + \big(X_{n_i,t-1} - X_{n_i,t}\big)^2 \right]^{\frac{1}{2}}, \quad (6)$$

where $\frac{\partial \boldsymbol{X}}{\partial e_{n_i n_j}}$ is the discrete edge derivative on the joint graph and, in the last equation, $n_j \overset{\mathcal{G}}{\sim} n_i$ are the neighbors of u_{n_i} in $\mathcal{G}$. We can also obtain a global notion of smoothness using the p-Dirichlet form

$$S_p(\boldsymbol{X}) := \frac{1}{p} \sum_{n=1}^{N} \sum_{t=1}^{p} \|\nabla_{n,t} \boldsymbol{X}\|_2^p \quad (7)$$

$$= \frac{1}{p} \sum_{n_i=1}^{N} \left[\sum_{n_j \overset{\mathcal{G}}{\sim} n_i} \big(X_{n_j,t} - X_{n_i,t}\big)^2 + \big(X_{n_i,t-1} - X_{n_i,t}\big)^2 \right]^{\frac{p}{2}}.$$

For $p = 2$ and after some manipulation, we find that

$$S_2(\boldsymbol{X}) = \text{vec}(\boldsymbol{X})^\top \boldsymbol{L}_J \text{vec}(\boldsymbol{X}) = \boldsymbol{x}^\top \boldsymbol{L}_J \boldsymbol{x}. \quad (8)$$

Similarly to the GFT, $S_2(\boldsymbol{X})$ is a quadratic form of the (joint) Laplacian, which implies that $S_2(\boldsymbol{X}) \geq 0$. Yet, here the variation of a signal is not only w.r.t. $\mathcal{G}$ but also w.r.t. time. For instance, $S_2(\boldsymbol{X}) = 0$ only if the signal is constant across all vertices and time-instances and, in general, the slower the values change along the graph and time domains, the smaller $S_2(\boldsymbol{X})$ becomes. Moreover, according to the Courant-Fischer theorem, when JFT is unitary, the signals which minimize $S_2(\boldsymbol{X})$ are exactly the eigenvectors of $\boldsymbol{L}_J$ (i.e., the rows of $\boldsymbol{V}_J$) with the corresponding minima being the associated eigenvalues λ_J of $\boldsymbol{L}_J$ [3]. JFT therefore characterizes a signal by how close its projections lie to the minimizers of the global variation $S_2(\boldsymbol{X})$; meaning that terms of low joint frequency λ_J (projections to eigenvectors associated with small eigenvalues) correspond to smoother signals and vice-versa.

3. JOINT FILTERING AND DENOISING

In the most general form, joint filtering is defined over a two-dimensional frequency domain, conveying the time- and graph- frequency of the signal. That is, one may now define joint frequency response $h(\lambda_T, \lambda_G)$ describing how the filter should change the frequency components independently w.r.t. λ_T and λ_G

$$h(\boldsymbol{L}_J)\, \boldsymbol{x} = \sum_{t=1,n=1}^{T,N} h(\lambda_T(t), \lambda_G(n))\, \boldsymbol{u}_J(t,n) \boldsymbol{v}_J^*(t,n)\, \boldsymbol{x}, \quad (9)$$

where $\boldsymbol{v}_J^*(t,n) = \boldsymbol{v}_T^*(t) \otimes \boldsymbol{v}_G^*(n)$ and $\boldsymbol{u}_J(t,n) = \boldsymbol{u}_T(t) \otimes \boldsymbol{u}_G(n)$ are eigenvectors of $\boldsymbol{L}_J$ with eigenvalue $\lambda_T(t) + \lambda_G(n)$.

To motivate the utility of the joint filters, Section 3.1 discusses a time-varying graph signal denoising problem that can be solved optimally by joint filtering. Section 3.2, then presents some preliminary results on the implementation of joint filtering.

3.1. Joint signal denoising

Suppose that we want to recover a graph signal $\boldsymbol{x} \in \mathbb{C}^{NT}$ from an interfering signal $\boldsymbol{w} \in \mathbb{C}^{NT}$, given the signals' statistical properties. In this case however, the two signals are zero mean and their covariances can be written as $\boldsymbol{\Sigma}_x = \boldsymbol{f}(\boldsymbol{L}_J)$ and $\boldsymbol{\Sigma}_w = \boldsymbol{g}(\boldsymbol{L}_J)$, respectively. Such processes are generalization of those presented in [11, 22] and can be used to model signals which are wide-sense stationary (WSS) in the time domain (covariance codiagonalizable with $\boldsymbol{L}_T$) or graph domain (GWSS) (covariance codiagonalizable with $\boldsymbol{L}_G$). For instance, one may consider a GWSS signal corrupted by a WSS signal, where both signals have arbitrary distribution (not necessarily Gaussian). Alternatively, the proposed model can capture the solution of differential equations of the Laplacian, such as of the wave equation—see example below.

The linear operator $\bar{\boldsymbol{F}}$ that recovers $\boldsymbol{x}$ from $\boldsymbol{y} = \boldsymbol{x} + \boldsymbol{w}$ with minimal mean-squared error is known to be

$$\bar{\boldsymbol{F}} = \underset{\boldsymbol{F}}{\text{argmin}}\, \mathbf{E}\left[\frac{\|\boldsymbol{F}\boldsymbol{y} - \boldsymbol{x}\|_2^2}{NT} \right] = \boldsymbol{\Sigma}_x (\boldsymbol{\Sigma}_x + \boldsymbol{\Sigma}_w)^\dagger \quad (10)$$

where $(\dagger)$ denotes the pseudo-inverse. Using the properties of JFT and of the Kronecker product (the derivation is not included due to space limitations), it can be shown that the

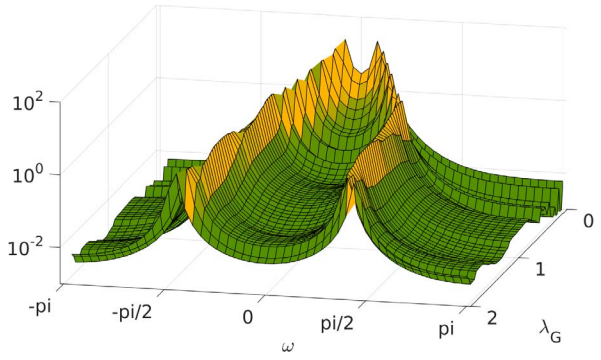

Fig. 2. The solution of a wave equation on a graph has a distinctive non-separable structure in the joint frequency domain.

optimal filter is equivalent to

$$\bar{\boldsymbol{F}} = \boldsymbol{U}_J \left(\boldsymbol{I}_T \otimes \boldsymbol{g}(\boldsymbol{\Lambda}_G)\right) \left(\boldsymbol{f}(\boldsymbol{\Lambda}_T) \oplus \boldsymbol{g}(\boldsymbol{\Lambda}_G)\right)^\dagger \boldsymbol{V}_J, \qquad (11)$$

which is a joint filter with frequency response

$$h(\lambda_T, \lambda_G) = \begin{cases} 0, & \text{if } f(\lambda_T, \lambda_G) + g(\lambda_T, \lambda_G) = 0 \\ \dfrac{f(\lambda_T, \lambda_G)}{f(\lambda_T, \lambda_G) + g(\lambda_T, \lambda_G)}, & \text{ow.} \end{cases}$$

Therefore, signals $\boldsymbol{x}$ and $\boldsymbol{w}$ cannot be well separated by acting disjointly on their respective domains; the best linear estimator $\bar{\boldsymbol{F}}\boldsymbol{y}$ of $\boldsymbol{x}$, is given by a filter acting on the joint Fourier domain.

Wave denoising. Let us consider, as an example, a time-varying graph signal (the wave) which is created by the standard wave equation on a graph [14]

$$\boldsymbol{x}_t = a\left(\boldsymbol{x}_{t-1} - \boldsymbol{x}_{t-2}\right) - \left(b\boldsymbol{I}_N + c\boldsymbol{L}_G\right)\boldsymbol{x}_{t-1}, \qquad (12)$$

where $\boldsymbol{x}_0 = 0$ and $\boldsymbol{x}_1$ is a gaussian random vector with zero mean and identity covariance. The wave output is corrupted with white noise $\boldsymbol{w}$ of progressively larger variance. The covariance of $\text{GFT}\{\boldsymbol{x}_t, \mathcal{G}\}$ is then diagonal with the n-th diagonal element equal to

$$\left[\boldsymbol{\Sigma}_{\text{GFT}\{\boldsymbol{x}_t,\mathcal{G}\}}\right]_{nn} = \left(\frac{r_1^{-t} - r_2^{-t}}{r_1 - r_2}\right)^2, \qquad (13)$$

where r_1, r_2 are the roots of polynomial $1 + (c\lambda_n - b + a)s + as^2 = 0$ in variable s (the derivation is not included due to space limitations). As illustrated by Fig. 2, the frequency response of $\boldsymbol{\Sigma}_{\boldsymbol{x}}$ (obtained by taking the DFT of the above expression) has a very distinctive structure in the joint frequency domain and is not separable (i.e., it cannot be written as the Kronecker product of a time and graph frequency response).

Fig. 3 depicts the denoising accuracy of the optimal joint filter with that obtained by Tikhonov and TV denoising in the graph and time domains disjointly. The results are averaged over five realizations with random geometric graphs of 100 nodes and average degree 6.76. Using a joint filter, we are able to take into account the covariance of $\boldsymbol{x}$ in the denoising, yielding a significant improvement over previous methods. We remark that, being non-separable, the covariance cannot be effectively modeled by solely time- or graph-based methods [11].

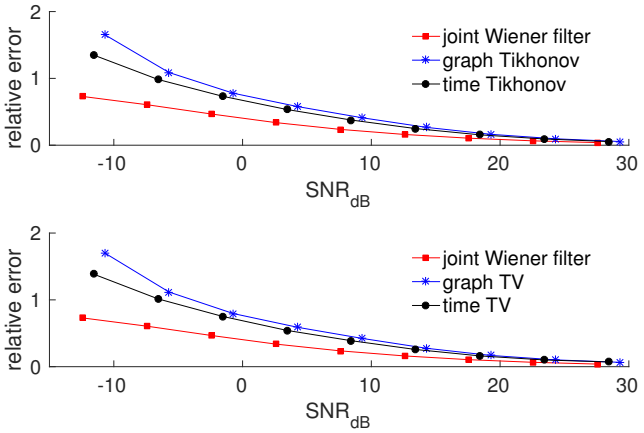

Fig. 3. Relative denoising error of joint approach (joint filter computed in closed form and approximated by a joint FIR) and denoising schemes over the time and graph domains. The horizontal axis measures the input signal-to-noise error. A small horizontal has been inserted to improve visibility.

3.2. Joint filter implementation

To avoid computing the GFT and DFT matrices, we approximate the desired joint frequency response $h^\star(\lambda_T, \lambda_G)$ by a bivariate polynomial of orders K, L in time and graph, respectively

$$h(\lambda_T, \lambda_G) = \sum_{k=0,\ell=0}^{K,L} c_{kl} \lambda_T^k \lambda_G^\ell, \qquad (14)$$

with coefficients c_{kl} chosen to minimize a certain norm (such as the max-norm or the euclidean norm) of the approximation error. The corresponding joint filter is $\boldsymbol{F}\boldsymbol{x} = \sum_{k=0,\ell=0}^{K,L} c_{kl} \left(\boldsymbol{L}_T^k \otimes \boldsymbol{L}_G^\ell\right) \boldsymbol{x}$ which, can be easily shown to possess the required response and is computable distributedly: each term $\boldsymbol{L}_T^k \otimes \boldsymbol{L}_G^\ell \boldsymbol{x}$ is computed by iteratively multiplying the signal by $\boldsymbol{I}_N \otimes \boldsymbol{L}_G$ (ℓ times) and $\boldsymbol{L}_T \otimes \boldsymbol{I}_N$ (k times), with each multiplication being a local operator on the joint graph and requiring the communication of $2MT$ and NT values, respectively. Since $2MT \geq NT$, the most efficient scheme, which involves computing first all powers $\boldsymbol{I}_T \otimes \boldsymbol{L}_G^\ell \boldsymbol{x}$ and then using them to compute remaining terms, has complexity $2MTK + (K+1)NTL = TK(2M + NL) + NTL = O(TKM + TKNL)$, which is linear on the number of samples, graph edges, and approximation orders. As shown in Figure 3, using joint FIR (the coefficients are here found by bivariate Chebychev approximation [12] of order 40), the above filters can approximate the closed-form solution of the joint wave denoising problem. Due to space limitations, we defer a more comprehensive discussion for the extended version of this paper.

References

[1] L. A. Adamic and N. Glance. The political blogosphere and the 2004 us election: divided they blog. In *LinkKDD*, pages 36–43. ACM, 2005.

[2] M. Belkin and P. Niyogi. Semi-supervised learning on riemannian manifolds. *Machine learning*, 56(1-3):209–239, 2004.

[3] R. A. Horn and C. R. Johnson. *Matrix analysis*. Cambridge university press, 2012.

[4] E. Isufi, G. Leus, and P. Banelli. 2-dimensional finite impulse response graph-temporal filters. *CoRR*, abs/1605.08964, 2016.

[5] A. J. Laub. *Matrix analysis for scientists and engineers*. Siam, 2005.

[6] A. Loukas, A. Simonetto, and G. Leus. Distributed Autoregressive Moving Average Graph Filters. *Signal Processing Letters*, 22(11):1931–1935, 2015.

[7] A. Loukas, M. A. Zúñiga, I. Protonotarios, and J. Gao. How to identify global trends from local decisions? Event Region Detection on Mobile Networks. In *INFOCOM*, 2014.

[8] A. Loukas, M. A. Zúñiga, M. Woehrle, M. Cattani, and K. Langendoen. Think Globally, Act Locally: On the Reshaping of Information Landscapes. In *IPSN*. ACM/IEEE, 2013.

[9] P. Mohan, V. N. Padmanabhan, and R. Ramjee. Nericell: rich monitoring of road and traffic conditions using mobile smartphones. In *SENSYS*, pages 323–336. ACM, 2008.

[10] S. K. Narang, A. Gadde, and A. Ortega. Signal processing techniques for interpolation in graph structured data. In *ICASSP*, pages 5445–5449. IEEE, 2013.

[11] N. Perraudin and P. Vandergheynst. Stationary signal processing on graphs. *arXiv preprint arXiv:1601.02522*, 2016.

[12] K. Poppe and R. Cools. *CHEBINT: Operations on multivariate Chebyshev approximations*. Department of Computer Science, KU Leuven, 2011.

[13] M. Püschel and J. M. Moura. Algebraic signal processing theory: 1-d space. *Signal Processing, IEEE Transactions on*, 56(8):3586–3599, 2008.

[14] T. Sahai, A. Speranzon, and A. Banaszuk. Wave equation based algorithm for distributed eigenvector computation. In *CDC*, pages 7308–7315. IEEE, 2010.

[15] A. Sandryhaila and J. M. Moura. Discrete signal processing on graphs. *Signal Processing, IEEE Transactions on*, 61(7):1644–1656, 2013.

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

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

[18] D. Shuman, B. Ricaud, P. Vandergheynst, et al. A windowed graph fourier transform. In *Statistical Signal Processing Workshop (SSP), 2012 IEEE*, pages 133–136. Ieee, 2012.

[19] D. I. Shuman, P. Vandergheynst, and P. Frossard. Chebyshev polynomial approximation for distributed signal processing. In *DCOSS*, pages 1–8. IEEE, 2011.

[20] A. J. Smola and R. Kondor. Kernels and regularization on graphs. In *Learning theory and kernel machines*, pages 144–158. Springer, 2003.

[21] A. J. Tatem, D. J. Rogers, and S. Hay. Global transport networks and infectious disease spread. *Advances in parasitology*, 62:293–343, 2006.

[22] C. Zhang, D. Florencio, and P. Chou. Graph signal processing - a probabilistic framework. Technical Report MSR-TR-2015-31, April 2015.

[23] F. Zhang and E. R. Hancock. Graph spectral image smoothing using the heat kernel. *Pattern Recognition*, 41(11):3328–3342, 2008.

[24] X. Zhu, Z. Ghahramani, and J. Lafferty. Semi-supervised learning using gaussian fields and harmonic functions. In *ICML*, volume 2, pages 912–919. AIAA Press, 2003.

[25] X. Zhu and M. Rabbat. Approximating signals supported on graphs. In *ICASSP*, pages 3921–3924. Citeseer, 2012.

[26] X. Zhu and M. Rabbat. Graph spectral compressed sensing for sensor networks. In *ICASSP*, pages 2865–2868. IEEE, 2012.