Optimal Graph-Filter Design and Applications to Distributed Linear Network Operators

Santiago Segarra, *Member, IEEE*, Antonio G. Marques, *Senior Member, IEEE*, and Alejandro Ribeiro, *Member, IEEE*

***Abstract*—We study the optimal design of graph filters (GFs) to implement arbitrary linear transformations between graph signals. GFs can be represented by matrix polynomials of the graph-shift operator (GSO). Since this operator captures the local structure of the graph, GFs naturally give rise to distributed operators. In most setups the GSO is given, so that GF design consists fundamentally in choosing the (filter) coefficients of the matrix polynomial to resemble desired linear transformations. We determine spectral conditions under which a specific linear transformation can be implemented perfectly using GFs. For the cases where perfect implementation is infeasible, we address the optimization of the filter coefficients to approximate the desired transformation. Additionally, for settings where the GSO itself can be modified, we study its optimal design too. After this, we introduce the notion of a *node-variant* GF, which allows the simultaneous implementation of multiple (regular) GFs in different nodes of the graph. This additional flexibility enables the design of more general operators without undermining the locality in implementation. Perfect and approximate designs are also studied for this new type of GFs. To showcase the relevance of the results in the context of distributed linear network operators, the paper closes with the application of our framework to two particular distributed problems: finite-time consensus and analog network coding.**

***Index Terms*—Graph signal processing, Graph filter design, Distributed linear transformation, Network operator, Finite-time consensus, Analog network coding.**

I. Introduction

Networks have often intrinsic value and are themselves the object of study. In other occasions, the network defines an underlying notion of proximity or dependence, but the object of interest is a signal defined on top of the graph, i.e., data associated with the nodes of the network. This is the matter addressed in the field of Graph Signal Processing (GSP), where notions such as frequency analysis, sampling, and stationarity are extended to signals supported on graphs [3]–[6]. Graph-supported signals exist in a number of fields, with examples encompassing gene-expression patterns defined on top of gene networks, the spread of epidemics over a social network, or the congestion level at the nodes of a telecommunication network. Transversal to the particular application, one must address the question of how to redesign traditional tools originally conceived to study and *process* signals defined on regular domains and extend them to the more complex graph domain. Motivated by this, the present manuscript looks at the problem of optimal design of graph filters (GFs), and its applications in the context of distributed linear network operators.

Design of GFs to approximate a given *impulse or frequency response* has been recently investigated in [4], [7], [8], considering both moving average and autoregressive configurations. This paper goes one step further and addresses the optimization of GFs to approximate a pre-specified *generic* linear transformation. Mathematically, GFs are graph-signal operators that can be expressed as polynomials of the graph-shift [9], which is a sparse matrix accounting for the local structure of the graph. Hence, GF design amounts to selecting the filter coefficients and, in certain applications, the entries of the graph-shift operator itself [10]. An attractive feature of GFs is that they naturally give rise to (local) *distributed implementations*. Therefore, our results are also relevant from the point of view of *network* linear transformations. During the last decade, development of distributed linear schemes has attracted a lot of attention. The field has evolved quickly, shifting from the estimation of simple network parameters such as the mean using consensus algorithms [11]–[15], to advanced inference problems such as sparse linear regression [16]; see also [17] for a review. Indeed, in the context of GSP but without leveraging the particular structure of a GF, several works have addressed the problem of *distributed* implementation of different linear operators. Examples include the projection of graph signals onto low-dimensional spaces [18]; the approximation of a specific class of linear transformations (the so-called graph Fourier multipliers) using Chebyshev polynomials [19]; or graph-signal inpainting [20], [21]. GFs have been used to implement distributedly *specific* linear transformations such as fast average consensus [22], projections onto the low-rank space of graph-bandlimited signals [23], or interpolation of graph-bandlimited signals [24], [25], but not for general linear transformations. Design of distributed linear network operators under a GF framework is useful not only because it identifies conditions and rules for perfect and approximate implementation, but also because it allows us to leverage many of the insights and results existing for classical linear time-invariant filters.

To further motivate the practical interest of the investigated problem, one can exploit the fact that the output of a GF can be viewed as the outcome of a diffusion or spreading process, with the filter coefficients corresponding to the rate of diffusion. Therefore, our results can be applied to net-

Work in this paper is supported by the USA NSF CCF-1217963 and the Spanish MINECO grants No TEC2013-41604-R and TEC2016-75361-R. S. Segarra is with the Inst. for Data, Systems and Society, Massachusetts Inst. of Technology. A. G. Marques is with the Dept. of Signal Theory and Comms., King Juan Carlos Univ. A. Ribeiro is with the Dept. of Electrical and Systems Eng., Univ. of Pennsylvania. Emails: segarra@mit.edu, antonio.garcia.marques@urjc.es, and aribeiro@seas.upenn.edu. Part of the results in this paper were presented at the *Allerton Conf.* [1] and ICASSP [2].

work processes evolving according to a dynamics given by a (weighted) connectivity graph. The input of the GF is the initial network state, the output is the final network state, and the shift operator is given by the connectivity graph. The objective is then to design the filter coefficients, which amounts to tuning the parameters that control the dynamics, to approximate a given linear transformation. Another relevant example is distributed state estimation in power or communication networks, such as wireless sensor networks, where lack of infrastructure or privacy issues prevents the deployment of a central controller and the processing and exchange of information must be implemented locally. In this case, the entries of the shift correspond to the weights that each node gives to the information received from its neighbors, and the filter coefficients correspond to the weights that nodes give to the signals exchanged at different time instants.

The contributions and organization of the paper are as follows. In Section II, we first introduce the concepts of graph signals and (node-invariant) GFs, and then propose two new types of GF, whose coefficients are allowed to vary across nodes. Upon casting the problem of approximating linear *network transformations* as a *GF design*, Sections III and IV investigate perfect and approximate GF implementation. In Section III, the focus is on node-invariant GFs. First, conditions under which a specific linear transformation can be implemented perfectly are identified. These conditions depend on the spectrum of the linear transformation, the spectrum of the graph-shift operator, and the order of the filter. For the cases where the previous conditions are not met, the design of optimal filter coefficients to approximate the desired transformation under different error metrics is addressed. Moreover, for setups where the entries of the shift can be modified, the spectral conditions are also leveraged to design shifts that facilitate the implementation of the given transformation. Section IV mimics the structure of Section III, but focusing on node-variant filters. To illustrate the relevance of our results for the *design of distributed network operators*, Section V applies the developed framework to the problems of finite-time consensus [22], [26] and analog network coding (ANC) [27], [28]. Numerical experiments and concluding remarks in Sections VI and VII wrap-up the manuscript. Relative to our conference papers [1] and [2], here we present a new definition for node-variant GF, provide proofs for our theoretical claims, investigate the problem of optimal shift design, discuss additional insights and generalizations, and report new numerical results.

Notation: In general, the entries of a matrix $\mathbf{X}$ and a vector $\mathbf{x}$ will be denoted as X_{ij} and x_i; however, when contributing to avoid confusion, the alternative notation $[\mathbf{X}]_{ij}$ and $[\mathbf{x}]_i$ will be used. The notation ^{T}, ^{H}, and † stands for transpose, transpose conjugate, and Moore-Penrose pseudoinverse, respectively; $\lambda_{\max}(\mathbf{X})$ is the largest eigenvalue of the symmetric matrix $\mathbf{X}$; $\mathbf{e}_i$ is the i-th $N \times 1$ canonical basis vector (all entries of $\mathbf{e}_i$ are zero except for the i-th one, which is one); $\mathbf{0}$ and $\mathbf{1}$ are the all-zero and all-one vectors, respectively; $\odot$ denotes the Khatri-Rao (column-wise Kronecker) product and $\circ$ the Hadamard (element-wise) product.

II. Graph signals and filters

Let $\mathcal{G}$ denote a directed graph with a set of N nodes or vertices $\mathcal{N}$ and a set of links $\mathcal{E}$, such that if node i is connected to j, then $(i,j) \in \mathcal{E}$. The (incoming) neighborhood of i is defined as the set of nodes $\mathcal{N}_i = \{j \,|\, (j,i) \in \mathcal{E}\}$ connected to i. For any given graph we define the adjacency matrix $\mathbf{A}$ as a sparse $N \times N$ matrix with non-zero elements A_{ji} if and only if $(i,j) \in \mathcal{E}$. The value of A_{ji} captures the strength of the connection from i to j. The focus of this paper is not on analyzing $\mathcal{G}$, but graph signals defined on the set of nodes $\mathcal{N}$. Formally, each of these signals can be represented as a vector $\mathbf{x} = [x_1, ..., x_N]^T \in \mathbb{R}^N$ where the i-th element represents the value of the signal at node i or, alternatively, as a function $f : \mathcal{N} \to \mathbb{R}$, defined on the vertices of the graph.

The graph $\mathcal{G}$ is endowed with a *graph-shift operator* $\mathbf{S}$ [4], [9]. The operator $\mathbf{S}$ is an $N \times N$ matrix whose entry S_{ji} can be non-zero only if $i = j$ or if $(i,j) \in \mathcal{E}$. The sparsity pattern of $\mathbf{S}$ captures the local structure of $\mathcal{G}$, but we make no specific assumptions on the values of the non-zero entries of $\mathbf{S}$. Choices for $\mathbf{S}$ are $\mathbf{A}$ [4], [9], the graph Laplacian $\mathbf{L}$ [3], and their respective generalizations [29]. The intuition behind $\mathbf{S}$ is to represent a linear transformation that can be computed *locally* at the nodes of the graph. More rigorously, if $\mathbf{y}$ is defined as $\mathbf{y} = \mathbf{S}\mathbf{x}$, then node i can compute y_i provided that it has access to the value of x_j at $j \in \mathcal{N}_i$. We assume henceforth that $\mathbf{S}$ is diagonalizable, so that there exists an $N \times N$ matrix $\mathbf{V}$ and an $N \times N$ diagonal matrix $\mathbf{\Lambda}$ such that $\mathbf{S} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^{-1}$. When $\mathbf{S}$ is normal, meaning that $\mathbf{S}\mathbf{S}^H = \mathbf{S}^H\mathbf{S}$, not only $\mathbf{S}$ is diagonalizable but $\mathbf{V}^{-1} = \mathbf{V}^H$ is unitary. Given a graph signal $\mathbf{x}$, we refer to $\widehat{\mathbf{x}} := \mathbf{V}^{-1}\mathbf{x}$ as the frequency representation of $\mathbf{x}$ [9].

A linear graph signal operator is a transformation $\mathbf{B} : \mathbb{R}^N \to \mathbb{R}^N$ between graph signals. Since the transformation is linear and the input and output spaces are the same, $\mathbf{B}$ can be represented by a square $N \times N$ matrix.

A. GFs and distributed implementation

Here we define (node-invariant) GFs, provide some intuition on their behavior, and discuss their distributed implementation. Mathematically, with $\mathbf{c} := [c_0, \ldots, c_{L-1}]^T$ representing the vector of coefficients, GFs are linear graph-signal operators $\mathbf{H} : \mathbb{R}^N \to \mathbb{R}^N$ of the form

$$\mathbf{H} := \sum_{l=0}^{L-1} c_l \mathbf{S}^l. \tag{1}$$

That is, GFs are *polynomials* of the graph-shift operator [4]. To emphasize its dependence on $\mathbf{c}$ and $\mathbf{S}$, the GF in (1) will be occasionally written as $\mathbf{H}(\mathbf{c}, \mathbf{S})$. Leveraging the spectral decomposition of the shift, the GF $\mathbf{H}$ can also be written as $\mathbf{H} = \mathbf{V}\big(\sum_{l=0}^{L-1} c_l \mathbf{\Lambda}^l\big)\mathbf{V}^{-1}$. The diagonal matrix $\widehat{\mathbf{H}} := \sum_{l=0}^{L-1} c_l \mathbf{\Lambda}^l$ can then be viewed as the frequency response of $\mathbf{H}$ and it can be alternatively written as $\widehat{\mathbf{H}} = \text{diag}(\widehat{\mathbf{c}})$, where vector $\widehat{\mathbf{c}}$ is a vector that contains the N frequency responses of the filter. Let λ_k denote the k-th eigenvalue of $\mathbf{S}$ and define

the $N \times L$ Vandermonde matrix

$$\mathbf{\Psi} := \begin{pmatrix} 1 & \lambda_1 & \dots & \lambda_1^{L-1} \\ \vdots & \vdots & & \vdots \\ 1 & \lambda_N & \dots & \lambda_N^{L-1} \end{pmatrix}. \tag{2}$$

For future reference, D denotes the number of distinct eigenvalues in $\{\lambda_k\}_{k=1}^N$. Using (2), it holds that $\widehat{\mathbf{c}} = \mathbf{\Psi}\mathbf{c}$ and thus

$$\mathbf{H} = \sum_{l=0}^{L-1} c_l \mathbf{S}^l = \mathbf{V}\text{diag}\big(\mathbf{\Psi}\mathbf{c}\big)\mathbf{V}^{-1} = \mathbf{V}\text{diag}(\widehat{\mathbf{c}})\mathbf{V}^{-1}. \tag{3}$$

This implies that if $\mathbf{y}$ is defined as $\mathbf{y} = \mathbf{H}\mathbf{x}$, its frequency representation $\widehat{\mathbf{y}} = \mathbf{V}^{-1}\mathbf{y}$ satisfies

$$\widehat{\mathbf{y}} = \text{diag}\big(\mathbf{\Psi}\mathbf{c}\big)\mathbf{V}^{-1}\mathbf{x} = \text{diag}\big(\widehat{\mathbf{c}}\big)\widehat{\mathbf{x}} = \widehat{\mathbf{c}} \circ \widehat{\mathbf{x}}, \tag{4}$$

which shows that the output at a given frequency depends only on the value of the input and the filter response at that given frequency. In contrast with traditional discrete signal processing, where the operator that transforms the signals and the filter coefficients into the frequency domain is the same – the Discrete Fourier Transform (DFT) –, when a generic $\mathbf{S}$ is considered, matrices $\mathbf{V}^{-1}$ and $\mathbf{\Psi}$ are different.

A particularity of GFs is that they can be implemented locally, e.g., with $L-1$ exchanges of information among neighbors. This is not surprising, since the shift $\mathbf{S}$ is a local operator. To formalize this, let us define the l-th shifted input signal as $\mathbf{z}^{(l)} := \mathbf{S}^l\mathbf{x}$ and note that two properties of $\mathbf{z}^{(l)}$ are: i) it can be computed recursively (sequentially) as $\mathbf{z}^{(l)} = \mathbf{S}\mathbf{z}^{(l-1)}$, with $\mathbf{z}^{(0)} = \mathbf{x}$; and ii) node i can obtain $[\mathbf{z}^{(l)}]_i$ locally based on the values of $[\mathbf{z}^{(l-1)}]_j$ at $j \in \mathcal{N}_i$. To emphasize this locality, we define $\mathbf{z}_i$ as an $L \times 1$ vector collecting the entries of $\{\mathbf{z}^{(l)}\}_{l=0}^{L-1}$ that are known by node i, so that $[\mathbf{z}_i]_l := [\mathbf{z}^{(l)}]_i$. With $\mathbf{y}$ denoting the output of a GF for the input signal $\mathbf{x}$, it follows from (1) that

$$\mathbf{y} = \mathbf{H}\,\mathbf{x} = \sum_{l=0}^{L-1} c_l \mathbf{S}^l\mathbf{x} = \sum_{l=0}^{L-1} c_l \mathbf{z}^{(l)}. \tag{5}$$

Hence, the i-th entry of vector $\mathbf{y}$ can be computed as $y_i = \sum_{l=0}^{L-1} c_l [\mathbf{z}^{(l)}]_i = \mathbf{c}^T\mathbf{z}_i$, showing that if the nodes know the value of the filter coefficients, y_i can be computed using solely information available at node i. This also suggests a two-step distributed implementation of GFs. First, $L-1$ sequential shifts are applied to the input signal $\mathbf{x}$ using only local information. At each shift, every node stores its own value of the shifted input. In the second step, nodes perform a linear combination of the values obtained in the first step. The weights, that are the same across nodes, are given by the coefficients $\mathbf{c}$. Alternatively, these two steps can be implemented simultaneously via a "multiply and accumulate" operation. A different expression to define a GF is

$$\mathbf{H} = a_0 \prod_{l=1}^{L-1} (\mathbf{S} - a_l\mathbf{I}), \tag{6}$$

which also gives rise to a polynomial on $\mathbf{S}$ of degree $L-1$. The previous expression can be leveraged to obtain an alternative distributed implementation of GFs. To show this, we define the intermediate graph signal $\mathbf{w}^{(l)} = \mathbf{S}\mathbf{w}^{(l-1)} - a_l\mathbf{w}^{(l-1)}$, with $\mathbf{w}^{(0)} = \mathbf{x}$. Notice that: i) the computation of $\mathbf{w}^{(l)}$ based on $\mathbf{w}^{(l-1)}$ can be performed by each node locally; and ii) the output of applying $\mathbf{H}$ to input $\mathbf{x}$ can be obtained after $L-1$ shifts just by setting $\mathbf{y} = \mathbf{H}\mathbf{x} = a_0\mathbf{w}^{(L-1)}$.

A convenient property of the representation in (6) is that it provides a straightforward way to design band-pass (frequency annihilating) filters [23]. In particular, upon setting $a_l = \lambda_k$ for the k-th eigenvalue of $\mathbf{S}$, filter $\mathbf{H}$ will eliminate the frequency basis $\mathbf{v}_k$, i.e., the eigenvector associated with λ_k.

B. Node-variant GFs

This paper proposes a generalization of GFs, called *node-variant* GFs, as operators $\mathbf{H}_{\text{nv}} : \mathbb{R}^N \to \mathbb{R}^N$ of the form [cf. (1)]

$$\mathbf{H}_{\text{nv}} := \sum_{l=0}^{L-1} \text{diag}(\mathbf{c}^{(l)})\mathbf{S}^l. \tag{7}$$

Whenever the $N \times 1$ vectors $\mathbf{c}^{(l)}$ are constant, i.e. $\mathbf{c}^{(l)} = c_l\mathbf{1}$ for all l, the node-variant GF reduces to the regular (node-invariant) GF. However, for general $\mathbf{c}^{(l)}$, when $\mathbf{H}_{\text{nv}}$ is applied to a signal $\mathbf{x}$, each node applies different weights to the shifted signals $\mathbf{z}^{(l)} = \mathbf{S}^l\mathbf{x}$. This additional flexibility enables the design of more general operators without undermining the local implementation. For notational convenience, the filter coefficients associated with node i are collected in the $L \times 1$ vector $\mathbf{c}_i$, such that $[\mathbf{c}_i]_l = [\mathbf{c}^{(l)}]_i$, and we define the $L \times N$ matrix $\mathbf{C} := [\mathbf{c}_1, \dots, \mathbf{c}_N]$. The GFs in (7) can be viewed as a generalization of linear time-varying filters whose impulse response changes with time. Whenever convenient to emphasize their dependence on the filter coefficients or the shift, the GF $\mathbf{H}_{\text{nv}}$ in (7) will be written as $\mathbf{H}_{\text{nv}}\big(\{\mathbf{c}_i\}_{i=1}^N, \mathbf{S}\big)$.

Since $\mathbf{S}^l$ and $\text{diag}(\mathbf{c}^{(l)})$ are not simultaneously diagonalizable (i.e., their eigenvectors are not the same), the neat frequency interpretation succeeding (1) does not hold true for the filters in (7). However, the spectral decomposition of $\mathbf{S}$ can still be used to understand how the output of the filter at a given node i depends on the frequency components of the input. To be specific, let us write the filter in (7) as

$$\mathbf{H}_{\text{nv}} = \sum_{l=0}^{L-1} \text{diag}(\mathbf{c}^{(l)})\mathbf{V}\mathbf{\Lambda}^l\mathbf{V}^{-1}. \tag{8}$$

Next, to analyze the effect of $\mathbf{H}_{\text{nv}}$ on the value of the output signal at node i, consider the i-th row of $\mathbf{H}_{\text{nv}}$, given by

$$\mathbf{h}_i^T := \mathbf{e}_i^T\mathbf{H}_{\text{nv}} = \sum_{l=0}^{L-1} [\mathbf{c}_i]_l \mathbf{e}_i^T\mathbf{V}\mathbf{\Lambda}^l\mathbf{V}^{-1}, \tag{9}$$

with $\mathbf{e}_i$ being the i-th $N \times 1$ canonical vector. Defining the vectors $\mathbf{u}_i := \mathbf{V}^T\mathbf{e}_i$ and $\widehat{\mathbf{c}}_i := \mathbf{\Psi}\mathbf{c}_i$, we can rewrite (9) as

$$\begin{aligned} \mathbf{h}_i^T &= \sum_{l=0}^{L-1} [\mathbf{c}_i]_l \mathbf{u}_i^T\mathbf{\Lambda}^l\mathbf{V}^{-1} = \mathbf{u}_i^T\Big(\sum_{l=0}^{L-1} [\mathbf{c}_i]_l\mathbf{\Lambda}^l\Big)\mathbf{V}^{-1} \\ &= \mathbf{u}_i^T\text{diag}(\mathbf{\Psi}\mathbf{c}_i)\mathbf{V}^{-1} = \mathbf{u}_i^T\text{diag}(\widehat{\mathbf{c}}_i)\mathbf{V}^{-1}. \end{aligned} \tag{10}$$

The expression in (10) reveals that the output of the filter at node i, which is given by $\mathbf{h}_i^T\mathbf{x}$, can be viewed as an inner product of $\mathbf{V}^{-1}\mathbf{x}$ (the frequency representation of the input) and $\mathbf{u}_i$ (how strongly node i senses each of the frequencies),

modulated by $\widehat{\mathbf{c}}_i$ (the frequency response associated with the coefficients used by node i). The expressions in (10) will be leveraged in Section IV to identify the type of linear transformations that node-variant filters are able to implement.

As a final remark, note that an alternative definition for node-variant GFs – different from the one in (7) – is to consider

$$\mathbf{H}'_{\text{nv}} := \sum_{l=0}^{L-1} \mathbf{S}^l \text{diag}(\mathbf{c}^{(l)}). \tag{11}$$

As was the case for (7), if $\mathbf{c}^{(l)} = c_l \mathbf{1}$, then (11) is equivalent to the node-invariant GF in (1). For each of the terms in (11) the filter first modulates the signal $\mathbf{x}$ with $\mathbf{c}^{(l)}$ and then applies the shift $\mathbf{S}^l$. This is in contrast with (7), which first shifts $\mathbf{x}$ and then modulates the shifted input with the corresponding $\mathbf{c}^{(l)}$. As was the case for (7), it is possible to implement (11) in a distributed and sequential fashion. To be more specific, to obtain the output signal $\mathbf{y} = \mathbf{H}'_{\text{nv}}\mathbf{x}$ one starts with the intermediate graph signal $\mathbf{t}^{(0)} := \mathbf{0}$, computes $\mathbf{t}^{(l)} \in \mathbb{R}^N$ for $l > 0$ as $\mathbf{t}^{(l)} := \mathbf{S}\mathbf{t}^{(l-1)} + \text{diag}(\mathbf{c}^{(L-l)})\mathbf{x}$, and sets $\mathbf{y} = \mathbf{t}^{(L)}$. Although the rest of the paper focuses on node-variant GFs of the form in (7), the frequency interpretations in (9) and (10) as well as the optimal designs derived in the ensuing sections can be generalized for the GFs in (11).

Remark 1 Node-variant GFs come with some caveats, one of them being that their spectral interpretation, which is a key component in most graph signal processing works, is more involved than that of node-invariant GFs. On the other hand, they exhibit a number of advantages related to their flexibility. As already pointed out, from a practical point of view they offer a way to implement a very large class of linear operators in a distributed manner (see the discussions after (7) and (11), and the results in Sections IV and V). Moreover, just like classical time-varying filters, node-varying GFs are expected to play an important role in problems related to adaptive graph filtering or modeling of non-stationary graph processes, as well as in establishing links with control theory and modeling of local diffusion dynamics.

III. Optimal design of node-invariant GFs

The first goal is to use the node-invariant GFs in (1) to implement a pre-specified *linear* graph signal transformation $\mathbf{B}$. To be more concrete, we want to find the filter coefficients $\mathbf{c}$ such that the following equality

$$\mathbf{B} = \mathbf{H}\big(\mathbf{c}, \mathbf{S}\big) = \sum_{l=0}^{L-1} c_l \mathbf{S}^l \tag{12}$$

holds. This is addressed in Section III-A, while approximate solutions are analyzed in Section III-B.

A. Conditions for perfect implementation

With $\boldsymbol{\beta} := [\beta_1, ..., \beta_N]^T$ standing for the vector containing the eigenvalues of matrix $\mathbf{B}$ (including multiplicities), the conditions for perfect implementation are given next.

Proposition 1 *The linear transformation* $\mathbf{B}$ *can be implemented using a node-invariant GF* $\mathbf{H} = \mathbf{H}(\mathbf{c}, \mathbf{S})$ *of the form given in* (1) *if the three following conditions hold true:*
a) Matrices $\mathbf{B}$ *and* $\mathbf{S}$ *are simultaneously diagonalizable; i.e., all the eigenvectors of* $\mathbf{B}$ *and* $\mathbf{S}$ *coincide.*
b) For all (k_1,k_2) *such that* $\lambda_{k_1} = \lambda_{k_2}$*, it holds that* $\beta_{k_1} = \beta_{k_2}$*.*
c) The degree of $\mathbf{H}$ *is such that* $L \geq D$*.*

Proof: Using condition *a)*, we may reformulate the equivalence between $\mathbf{B}$ and $\mathbf{H}$ in the frequency domain and require their eigenvalues to be the same, i.e. $\boldsymbol{\beta} = \boldsymbol{\Psi}\mathbf{c}$. Denote by $\mathcal{K}$ the set of indices of the D unique eigenvalues of $\mathbf{S}$ and define the $N \times D$ matrix $\mathbf{E}_{\mathcal{K}} = [\mathbf{e}_{k_1}, \ldots, \mathbf{e}_{k_D}]$ for all $k_i \in \mathcal{K}$ and the $N \times N-D$ matrix $\mathbf{E}_{\bar{\mathcal{K}}} = [\mathbf{e}_{k_1}, \ldots, \mathbf{e}_{k_{N-D}}]$ for all $k_i \notin \mathcal{K}$. Then, we split the system of equations $\boldsymbol{\beta} = \boldsymbol{\Psi}\mathbf{c}$ into two groups

$$\mathbf{E}_{\mathcal{K}}^T \boldsymbol{\beta} = \mathbf{E}_{\mathcal{K}}^T \boldsymbol{\Psi}\mathbf{c}, \tag{13}$$

$$\mathbf{E}_{\bar{\mathcal{K}}}^T \boldsymbol{\beta} = \mathbf{E}_{\bar{\mathcal{K}}}^T \boldsymbol{\Psi}\mathbf{c}. \tag{14}$$

Condition *b)* guarantees that if $\mathbf{c}$ solves (13) then it also solves (14). Finally, combining the fact that $\mathbf{E}_{\mathcal{K}}^T \boldsymbol{\Psi}$ has full row rank (Vandermonde matrix with non-repeated basis) and condition *c)*, there exists at least one $\mathbf{c}$ that solves (13), and the proof concludes. ■

Based on the proposition, the expression for the optimal coefficients follows readily, as stated in the following corollary.

Corollary 1 *If the conditions in Proposition 1 hold true, a vector of filter coefficients* $\mathbf{c}^*$ *satisfying* (12) *is given by*

$$\mathbf{c}^* = \boldsymbol{\Psi}^\dagger \boldsymbol{\beta}. \tag{15}$$

Proof: If the conditions in Proposition 1 hold, then there exists a vector $\mathbf{c}$ such that the cost $\|\boldsymbol{\beta} - \boldsymbol{\Psi}\mathbf{c}\|_2^2$ is zero. Since the pseudo-inverse solution $\mathbf{c}^*$ in (15) is one of the solutions to the minimization of $\|\boldsymbol{\beta} - \boldsymbol{\Psi}\mathbf{c}\|_2^2$, we have that $\|\boldsymbol{\beta} - \boldsymbol{\Psi}\mathbf{c}^*\|_2^2 = 0$ and the corollary follows. When $L = D$, the solution is unique and, using the set $\mathcal{K}$ defined before (13), the expression in (15) can be alternatively written as $\mathbf{c}^* = (\mathbf{E}_{\mathcal{K}}^T \boldsymbol{\Psi})^{-1} (\mathbf{E}_{\mathcal{K}}^T \boldsymbol{\beta})$. ■

The corollary leverages the spectral interpretation of GFs, advocating for a design where the filter coefficients are found in the frequency domain. To be precise, if the conditions in Proposition 1 hold then: i) $\boldsymbol{\beta}$ represents the desired frequency response of the filter $\mathbf{H} = \sum_{l=0}^{L-1} c_l \mathbf{S}^l$; and ii) the pseudo-inverse $\boldsymbol{\Psi}^\dagger$ in (15) is the generalization of the *inverse* graph Fourier transform when matrix $\boldsymbol{\Psi}$ is not square.

To gain some intuition on the meaning of the proposition, let $\mathbf{A}_{dc}$ be the adjacency matrix of the directed cycle, which is the support of classical time-varying signals [3]. When Proposition 1 is particularized to $\mathbf{S} = \mathbf{A}_{dc}$, condition a) implies that the transformations $\mathbf{B}$ that can be implemented perfectly are those diagonalized by the DFT matrix, i.e., circulant transformations. In other words, if the output of the filter is a linear combination of successive applications of the shift $\mathbf{A}_{dc}$ to the input, only transformations that are time invariant (so that each row of $\mathbf{B}$ is the shifted version of the previous row) can be implemented perfectly using $\mathbf{H}(\mathbf{c}, \mathbf{A}_{dc})$.

Proposition 1 will also be exploited in Section III-C to design shifts able to implement desired linear transformations.

B. Approximate implementation

In general, if the conditions in Proposition 1 are not satisfied, perfect implementation of $\mathbf{B}$ is not feasible. In such cases, the filter coefficients $\mathbf{c}$ can be designed to minimize a pre-specified error metric. We suppose first that prior knowledge of the input signal $\mathbf{x}$ is available. This knowledge can be incorporated into the design of $\mathbf{c}$ and the goal is then to minimize an error metric of the difference *vector* $\mathbf{d} := \mathbf{H}\mathbf{x} - \mathbf{B}\mathbf{x}$. A case of particular interest is when $\mathbf{x}$ is drawn from a zero-mean distribution with known covariance $\mathbf{R}_\mathbf{x} := \mathbb{E}[\mathbf{x}\mathbf{x}^T]$. In this case, the error covariance is given by $\mathbf{R}_\mathbf{d} := \mathbb{E}[\mathbf{d}\mathbf{d}^T] = (\mathbf{H}-\mathbf{B})\mathbf{R}_\mathbf{x}(\mathbf{H}-\mathbf{B})^T$. Our objective is to pick $\mathbf{c}$ to minimize some metric of the error covariance matrix $\mathbf{R}_\mathbf{d}$. Two commonly used approaches are the minimization of $\text{Trace}(\mathbf{R}_\mathbf{d})$ and $\lambda_{\max}(\mathbf{R}_\mathbf{d})$, where the former is equivalent to minimizing the mean squared error (MSE) of $\mathbf{d}$ and the latter minimizes the worst-case error (WCE) by minimizing the maximum variance of $\mathbf{d}$ [30].

Define the $N^2 \times L$ matrix $\mathbf{\Theta}_{\mathbf{R}_\mathbf{x}} := [\text{vec}(\mathbf{I}\mathbf{R}_\mathbf{x}^{1/2}), \text{vec}(\mathbf{S}\mathbf{R}_\mathbf{x}^{1/2}), \ldots, \text{vec}(\mathbf{S}^{L-1}\mathbf{R}_\mathbf{x}^{1/2})]$ and the $N^2 \times 1$ vector $\mathbf{b}_{\mathbf{R}_\mathbf{x}} := \text{vec}(\mathbf{B}\mathbf{R}_\mathbf{x}^{1/2})$. The optimal filter coefficients are provided in the following proposition.

Proposition 2 *The optimal MSE filter coefficients* $\mathbf{c}^*_{\text{Tr}} := \text{argmin}_\mathbf{c}\, \text{Trace}(\mathbf{R}_\mathbf{d})$ *are given by*

$$\mathbf{c}^*_{\text{Tr}} = \mathbf{\Theta}^\dagger_{\mathbf{R}_\mathbf{x}}\mathbf{b}_{\mathbf{R}_\mathbf{x}} \overset{\star}{=} (\mathbf{\Theta}^T_{\mathbf{R}_\mathbf{x}}\mathbf{\Theta}_{\mathbf{R}_\mathbf{x}})^{-1}\mathbf{\Theta}^T_{\mathbf{R}_\mathbf{x}}\mathbf{b}_{\mathbf{R}_\mathbf{x}}, \tag{16}$$

where $\overset{\star}{=}$ *holds if* $\mathbf{\Theta}_{\mathbf{R}_\mathbf{x}}$ *has full column rank; while the optimal WCE coefficients* $\mathbf{c}^*_\lambda := \text{argmin}_\mathbf{c}\, \lambda_{\max}(\mathbf{R}_\mathbf{d})$ *are obtained as*

$$\{\mathbf{c}^*_\lambda, s^*\} = \underset{\{\mathbf{c},s\}}{\text{argmin}} \quad s \tag{17}$$

$$\text{s. to} \begin{bmatrix} s\mathbf{I} & \mathbf{V}\text{diag}(\mathbf{\Psi}\mathbf{c})\mathbf{V}^{-1} - \mathbf{B} \\ (\mathbf{V}\text{diag}(\mathbf{\Psi}\mathbf{c})\mathbf{V}^{-1} - \mathbf{B})^T & s\mathbf{R}_\mathbf{x}^{-1} \end{bmatrix} \succeq 0.$$

Proof : By expressing $\mathbf{R}_\mathbf{x} = \mathbf{R}_\mathbf{x}^{1/2}\mathbf{R}_\mathbf{x}^{1/2}$, we may rewrite $\text{Trace}(\mathbf{R}_\mathbf{d})$ as $\text{Trace}((\mathbf{H}\mathbf{R}_\mathbf{x}^{1/2} - \mathbf{B}\mathbf{R}_\mathbf{x}^{1/2})(\mathbf{H}\mathbf{R}_\mathbf{x}^{1/2} - \mathbf{B}\mathbf{R}_\mathbf{x}^{1/2})^T)$. Thus, given that for any real matrix $\mathbf{A}$ we have that $\|\mathbf{A}\|_\text{F}^2 = \text{Trace}(\mathbf{A}\mathbf{A}^T)$, it follows that

$$\text{Trace}(\mathbf{R}_\mathbf{d}) = \|\mathbf{H}\mathbf{R}_\mathbf{x}^{1/2} - \mathbf{B}\mathbf{R}_\mathbf{x}^{1/2}\|_\text{F}^2. \tag{18}$$

This implies that $\mathbf{c}^*_{\text{Tr}}$ minimizes the Frobenius norm in (18). From the definition of Frobenius norm, we have that $\|\mathbf{H}\mathbf{R}_\mathbf{x}^{1/2} - \mathbf{B}\mathbf{R}_\mathbf{x}^{1/2}\|_\text{F} = \|\text{vec}(\mathbf{H}\mathbf{R}_\mathbf{x}^{1/2}) - \text{vec}(\mathbf{B}\mathbf{R}_\mathbf{x}^{1/2})\|_2$. The result in (16) follows from noting that $\text{vec}(\mathbf{H}\mathbf{R}_\mathbf{x}^{1/2}) = \mathbf{\Theta}_{\mathbf{R}_\mathbf{x}}\mathbf{c}$ and using the pseudoinverse to solve the system of linear equations [31].

To show the result for $\mathbf{c}^*_\lambda$, note that the constraint in (17) can be rewritten in terms of its Schur complement as $s^2\mathbf{I} \succeq (\mathbf{H}-\mathbf{B})\mathbf{R}_\mathbf{x}(\mathbf{H}-\mathbf{B})^T$. In terms of eigenvalues, this is equivalent to $s^2 \geq \lambda_{\max}(\mathbf{R}_\mathbf{d})$, from where it follows that s^2 is an upper bound on $\lambda_{\max}(\mathbf{R}_\mathbf{d})$. Since the constraint in (17) is tight, minimizing the upper bound is equivalent to minimizing $\lambda_{\max}(\mathbf{R}_\mathbf{d})$, so that the obtained coefficients $\mathbf{c}^*_\lambda$ are optimal, as stated in the proposition; see [32], [33]. ■

Proposition 2 provides a closed-form expression for the MSE coefficients $\mathbf{c}^*_{\text{Tr}}$ and obtains the WCE coefficients $\mathbf{c}^*_\lambda$ as the solution of a semi-definite convex program [12]. Whenever the spectrum of $\mathbf{B}$ is incompatible with that of $\mathbf{S}$ – violations on conditions *a)* or *b)* in Proposition 1 – or when the filter degree is lower than that needed for perfect reconstruction, expressions (16) and (17) become relevant. Alternative error metrics and additional assumptions could be incorporated into the designs including, but not limited to, supplementary statistical knowledge of $\mathbf{x}$ and structural properties such as sparsity or bandlimitedness [9]. For example, if $\mathbf{S}$ is normal and $\mathbf{x}$ is graph stationary [6], then the designs in (16) and (17) can be addressed in the frequency domain, with the benefit of all involved matrices being diagonal. Finally, if no prior information on $\mathbf{x}$ is available, we may reformulate the optimal criterion to minimize the discrepancy between the *matrices* $\mathbf{H}$ and $\mathbf{B}$. The optimal solutions are still given by (16) and (17) by setting $\mathbf{R}_\mathbf{x} = \mathbf{I}$. More precisely, the resulting coefficients $\mathbf{c}^*_{\text{Tr}}$ in (16) minimize the error metric $\|\mathbf{H}-\mathbf{B}\|_\text{F}$, while $\mathbf{c}^*_\lambda$ in (17) is the minimizer of $\|\mathbf{H}-\mathbf{B}\|_2$.

C. Designing the shift

In this section we go beyond the design of the *filter coefficients* and discuss how to use the results in Proposition 1 to design a *shift* $\mathbf{S}$ that enables the implementation of a given linear transformation $\mathbf{B}$.

Since Proposition 1 established that $\mathbf{B}$ and $\mathbf{S}$ need to share the entire set of eigenvectors, we first look at the case where $\mathbf{B}$ is a rank-one matrix that can be written as $\mathbf{B}_{\text{rk1}} := \mathbf{a}\mathbf{b}^T$. Finding a shift able to implement $\mathbf{B} = \mathbf{B}_{\text{rk1}}$ is a favorable setup because $\mathbf{B}_{\text{rk1}}$ specifies only one of the eigenvectors of $\mathbf{S}$, while the others can be chosen to endow $\mathbf{S}$ with desired properties. Rank-one operators are common in, for example, distributed estimation setups such as a wireless sensor network, where the goal is for each of the sensors to obtain the same estimate. To be specific, suppose that the estimate $\hat{\theta}$ can be written as a linear combination of the observations available at the nodes, i.e., $\hat{\theta} = \sum_{i=1}^N b_i x_i$ where $\mathbf{x} = [x_1, ..., x_N]^T$ are the observations and $\mathbf{b} = [b_1, ..., b_N]^T$ the weighting coefficients. Then, the sought network transformation is $\mathbf{B}_{\text{rk1}} = \mathbf{1}\mathbf{b}^T$, so that when $\mathbf{B}_{\text{rk1}}$ is applied to $\mathbf{x}$ gives rise to $\hat{\theta}\mathbf{1}$.

The fact that $N-1$ eigenvalues of $\mathbf{B}_{\text{rk1}} = \mathbf{a}\mathbf{b}^T$ are equal to zero can be leveraged to show the following proposition.

Proposition 3 *Consider the graph* $\mathcal{G}$ *and the rank-one transformation* $\mathbf{B}_{\text{rk1}} = \mathbf{a}\mathbf{b}^T$. *If* $\mathcal{G}$ *is connected and* $a_i \neq 0$ *and* $b_i \neq 0$ *for all* i, *then there exists a shift* $\mathbf{S}$ *associated with* $\mathcal{G}$ *such that the transformation* $\mathbf{B}_{\text{rk1}}$ *can be written as a node-invariant GF* $\mathbf{H}(\mathbf{c}, \mathbf{S})$ *of the form given in* (1).

Proof : The proof is constructive. Suppose without loss of generality that $\mathbf{a}$ and $\mathbf{b}$ have unit norm and let $\mathcal{T}$ denote a spanning tree of $\mathcal{G}$ with edge set $\mathcal{E}_\mathcal{T}$. With $\mu \in \mathbb{R}$ being an arbitrary constant, set the shift operator $\mathbf{S}$ to

$$S_{ij} = \begin{cases} \mu + \sum_{n\in\mathcal{N}_i} a_n b_n & \text{if } i = j, \\ S_{ij} = -a_i b_j & \text{if } (i,j) \in \mathcal{E}_\mathcal{T}, \\ S_{ij} = 0 & \text{otherwise,} \end{cases} \tag{19}$$

where the neighborhood $\mathcal{N}_i$ refers to the spanning tree $\mathcal{T}$. We will show that the three conditions in Proposition 1 are

satisfied for $\mathbf{H} = \sum_{l=0}^{N-1} c_l \mathbf{S}^l$. First notice that since $D \leq N$, condition *c)* is fulfilled. To see that condition *a)* is satisfied, we must show that $\mathbf{B}_{\text{rk1}}$ and $\mathbf{S}$ share the eigenvectors. Since $\mathbf{B}_{\text{rk1}}$ is rank one, this requires showing that $\mathbf{a}$ is an eigenvector of $\mathbf{S}$. To this end, define $\mathbf{q} := \mathbf{S}\mathbf{a}$. Given the construction of $\mathbf{S}$ in (19), it holds that $q_i = S_{ii}a_i + \sum_{j\in\mathcal{N}_i} S_{ij}a_j = \mu a_i$, so that $\mathbf{a}$ is an eigenvector of $\mathbf{S}$ with eigenvalue μ. Since the shift is not symmetric, we also need to show that the row of $\mathbf{V}^{-1}$ associated with eigenvector $\mathbf{a}$ corresponds to $\mathbf{b}$. To do so, define $\mathbf{p} := \mathbf{S}^T\mathbf{b}$ and notice that $p_j = S_{jj}b_j + \sum_{i\in\mathcal{N}_j} S_{ij}b_i = \mu b_j$ so that $\mathbf{b}$ is an eigenvector of $\mathbf{S}^T$ with eigenvalue μ. Since $\mathbf{S}^T = (\mathbf{V}^{-1})^T\mathbf{\Lambda}\mathbf{V}^T$, $\mathbf{b}$ is the column of $(\mathbf{V}^{-1})^T$ (alternative the row of $\mathbf{V}^{-1}$) associated with eigenvalue μ. To conclude the proof, we need to show that the multiplicity of μ is one [cf. condition *b)* in Proposition 1].

To show that the eigenvalue μ is indeed simple, consider the incidence matrix $\mathbf{M_a} \in \mathbb{R}^{N\times N-1}$ of the spanning tree. The n-th row of $\mathbf{M_a}$ corresponds to node n, while each column corresponds to one link of $\mathcal{T}$. Suppose that the l-th link is given by (i,j) where we follow the convention that $i < j$, then the incidence matrix $\mathbf{M_a}$ is defined as $[\mathbf{M_a}]_{il} = a_j$, $[\mathbf{M_a}]_{jl} = -a_i$ and $[\mathbf{M_a}]_{ul} = 0$ for all $u \neq i$ and $u \neq j$.

Lemma 1 *If $a_i \neq 0$ for all i, then* $\ker(\mathbf{M_a}^T) = \text{span}\{\mathbf{a}\}$ *and* $\ker(\mathbf{M_a}) = \{\mathbf{0}\}$.

Proof: To prove that $\ker(\mathbf{M_a}^T) = \text{span}\{\mathbf{a}\}$, we first show that $\mathbf{M_a}^T\mathbf{a} = \mathbf{0}$. Using the definition of $\mathbf{M_a}$, it holds that for any $\mathbf{x} \in \mathbb{R}^N$ we have $[\mathbf{M_a}^T\mathbf{x}]_l = a_jx_i - a_ix_j$, where we recall that (i,j) is the l-th link in $\mathcal{E}_\mathcal{T}$. Upon setting $\mathbf{x} = \mathbf{a}$ it readily follows that $\mathbf{a} \in \ker(\mathbf{M_a}^T)$. To show that $\mathbf{a}$ is the only eigenvector with zero eigenvalue, consider matrix $\mathbf{L_a} = \mathbf{M_a}\mathbf{M_a}^T$. Since $[\mathbf{M_a}^T\mathbf{x}]_l = a_jx_i - a_ix_j$, it readily follows that $\mathbf{x}^T\mathbf{L_a}\mathbf{x} = \sum_{(i,j)\in\mathcal{E}_\mathcal{T}}(a_jx_i - a_ix_j)^2$. In order for $\mathbf{x}^T\mathbf{L_a}\mathbf{x}$ to be zero, all the terms $(a_jx_i - a_ix_j)^2$ must be zero. Since the graph is connected this is only possible if $\mathbf{x} = \alpha\mathbf{a}$. Hence, $\text{rank}(\mathbf{L_a}) = N-1$. Since $\mathbf{L_a} = \mathbf{M_a}\mathbf{M_a}^T$, the previous finding implies that $\text{rank}(\mathbf{M_a}^T) = N-1$; hence $\ker(\mathbf{M_a}^T)$ has dimension one. Note that the fact that $\text{rank}(\mathbf{M_a}^T) = N-1$ also implies that $\text{rank}(\mathbf{M_a}) = N-1$, which implies that the dimension of $\ker(\mathbf{M_a})$ is zero, concluding the proof. ■

Lemma 2 *Consider the shift* $\mathbf{S}$ *given in* (19) *and suppose that $a_i \neq 0$ and $b_i \neq 0$ for all i, then it holds that* $\ker(\mathbf{S} - \mu\mathbf{I}) = \text{span}\{\mathbf{a}\}$.

Proof: The key step to show this result is to notice that, given (19), the shift can be written as $\mathbf{S} = \mu\mathbf{I} + \mathbf{M_b}\mathbf{M_a}^T$. Using now the facts that $\ker(\mathbf{M_a}^T) = \text{span}\{\mathbf{a}\}$ (Lemma 1) and $\ker(\mathbf{M_b}) = \{\mathbf{0}\}$ (Lemma 1 with $\mathbf{a} = \mathbf{b}$), Lemma 2 follows. ■

From Lemma 2, it follows that $\mathbf{a}$ is the only eigenvector of $\mathbf{S} - \mu\mathbf{I}$ with null eigenvalue and, thus, the only eigenvector of $\mathbf{S}$ with eigenvalue equal to μ, as wanted. Finally, if we combine the facts that the eigenvalue in $\mathbf{S}$ associated with $\mathbf{a}$ is simple (non-repeated) and that every other eigenvalue is repeated in $\mathbf{B}_{\text{rk1}}$ (since they are equal to 0), fulfillment of condition *b)* follows, completing the proof. Note that the fact of $\mathcal{G}$ being connected is critical to guarantee that the multiplicity of μ is one. In fact it is straightforward to show that if the graph is not connected then the multiplicity of μ is equal to the number of components, so that the linear transformation cannot be implemented as a filter. ■

The proof of Proposition 3 not only guarantees the existence of an $\mathbf{S}$ for which $\mathbf{B}_{\text{rk1}} = \mathbf{a}\mathbf{b}^T$ can be written as a GF, but also provides a particular shift that achieves so [cf. (19)]. Clearly, the definition in (19), which can be understood as a "generalized" Laplacian using the entries of $\mathbf{a}$ and $\mathbf{b}$ as weights, is not unique. Note first that different spanning trees will give rise to different valid $\mathbf{S}$. Moreover, if $\mathbf{B}_{\text{rk1}}$ is symmetric, our proof can be adapted to hold for a shift $\mathbf{S}$ where the edge set in (19) corresponds to that of any connected subgraph of $\mathcal{G}$, including $\mathcal{G}$ itself. This is particularly relevant in the context of distributed implementation, since the interplay between the number of edges of the subgraph and the order of the filter can be exploited to reduce the number of local exchanges required to compute the rank-one transformation distributedly.

When the linear transformation $\mathbf{B}$ is of high rank, the design of a shift $\mathbf{S}$ that preserves the sparsity of $\mathcal{G}$ and shares the eigenvectors of $\mathbf{B}$ is more involved and not always feasible. One possibility to address the design is to leverage the results in [10] and obtain the shift by solving a problem of the form

$$\min_{\{\mathbf{S},\boldsymbol{\lambda}\}} \quad f(\mathbf{S}) \quad \text{s. to} \quad \mathbf{S} = \textstyle\sum_{k=1}^N \lambda_k \mathbf{W}_k, \ \mathbf{S} \in \mathcal{S}, \tag{20}$$

where $\mathbf{W}_k := \mathbf{V}\text{diag}(\mathbf{e}_k)\mathbf{V}^{-1}$. Note that in the above problem the optimization variables are effectively the eigenvalues $\boldsymbol{\lambda} = [\lambda_1, ..., \lambda_N]^T$, since the constraint $\mathbf{S} = \sum_{k=1}^N \lambda_k \mathbf{V}\text{diag}(\mathbf{e}_k)\mathbf{V}^{-1}$ forces the columns of $\mathbf{V}$ to be the eigenvectors of $\mathbf{S}$. The objective f promotes desirable network structural properties, such as sparsity or minimum-energy edge weights. The constraint set $\mathcal{S}$ imposes requirements on the sought shift, including the elements of $\mathbf{S}$ being zero for entries (i,j) corresponding to edges not present in $\mathcal{G}$. For setups where only a few eigenvectors are known or when the equality constraint $\mathbf{S} = \sum_{k=1}^N \lambda_k \mathbf{V}\text{diag}(\mathbf{e}_k)\mathbf{V}^{-1}$ is relaxed, which can be useful if the optimization in (20) is not feasible, see [10].

IV. Optimal design of node-variant GFs

The objective in this section is to implement pre-specified linear transformations using the node-variant GFs introduced in (7). Since node-invariant filters are a specific instance of (7), the class of linear network transformations that can be implemented perfectly using node-variant GFs is larger and the approximation error (when perfect implementation is not feasible) is smaller.

More specifically, given a desired linear transformation $\mathbf{B}$ we want to design the coefficient vectors $\mathbf{c}^{(l)}$ for $l = 0, \ldots, L-1$ so that the equality

$$\mathbf{B} = \mathbf{H}_{\text{nv}}\big(\{\mathbf{c}_i\}_{i=1}^N, \mathbf{S}\big) = \sum_{l=0}^{L-1} \text{diag}(\mathbf{c}^{(l)})\mathbf{S}^l \tag{21}$$

holds. Using the same structure than that in Section III, we first identify the conditions under which (21) can be solved exactly, and then analyze approximate solutions.

A. Conditions for perfect implementation

Defining the vectors $\mathbf{b}_i := \mathbf{B}^T\mathbf{e}_i$ and $\tilde{\mathbf{b}}_i := \mathbf{V}^T\mathbf{b}_i$, the conditions under which the equivalence in (21) can be achieved are given in Proposition 4 and the subsequent corollary.

Proposition 4 *The linear transformation* $\mathbf{B}$ *can be implemented using a node-variant GF* $\mathbf{H}_{\text{nv}} = \mathbf{H}_{\text{nv}}(\{\mathbf{c}_i\}_{i=1}^N, \mathbf{S})$ *of the form given in* (7) *if the three following conditions hold for all* i*:*
a) $[\tilde{\mathbf{b}}_i]_k = 0$ *for those* k *such that* $[\mathbf{u}_i]_k = 0$.
b) For all (k_1,k_2) *such that* $\lambda_{k_1} = \lambda_{k_2}$*, it holds that* $[\tilde{\mathbf{b}}_i]_{k_1}/[\mathbf{u}_i]_{k_1} = [\tilde{\mathbf{b}}_i]_{k_2}/[\mathbf{u}_i]_{k_2}$.
c) The degree of $\mathbf{H}_{\text{nv}}$ *is such that* $L \geq D$.

Proof: We use (10) to write the row-wise equality between $\mathbf{H}_{\text{nv}}$ and $\mathbf{B}$

$$\mathbf{b}_i^T = \mathbf{h}_i^T = \mathbf{u}_i^T \text{diag}(\mathbf{\Psi}\mathbf{c}_i)\mathbf{V}^{-1}, \tag{22}$$

for all i. Multiplying (22) from the right by $\mathbf{V}$, transposing the equality, and using the definition of $\tilde{\mathbf{b}}_i$, we get

$$\tilde{\mathbf{b}}_i = \text{diag}(\mathbf{\Psi}\mathbf{c}_i)\mathbf{u}_i = \text{diag}(\mathbf{u}_i)\mathbf{\Psi}\mathbf{c}_i, \tag{23}$$

for all i. Thus, for $\mathbf{B}$ to be implementable, for every node i we need to find the vector $\mathbf{c}_i$ that satisfies (23).

For each i, partition the set $\{1,\dots,N\}$ into three subsets of indices $\mathcal{K}_1^i$, $\mathcal{K}_2^i$, and $\mathcal{K}_3^i$ where i) $\mathcal{K}_1^i$ contains the indices k such that $[\mathbf{u}_i]_k = 0$; ii) $\mathcal{K}_2^i$ contains the indices of all the eigenvalues in $\{1,\dots,N\}\setminus\mathcal{K}_1^i$ that are distinct as well as one index per repeated eigenvalue λ_k, and; iii) $\mathcal{K}_3^i = \{1,\dots,N\}\setminus(\mathcal{K}_1^i \cup \mathcal{K}_2^i)$. Thus, separate the N linear equations in (23) in three groups

$$\mathbf{E}_{\mathcal{K}_1^i}^T\tilde{\mathbf{b}}_i = \text{diag}(\mathbf{E}_{\mathcal{K}_1^i}^T\mathbf{u}_i)\mathbf{E}_{\mathcal{K}_1^i}^T\mathbf{\Psi}\mathbf{c}_i, \tag{24}$$

$$\mathbf{E}_{\mathcal{K}_2^i}^T\tilde{\mathbf{b}}_i = \text{diag}(\mathbf{E}_{\mathcal{K}_2^i}^T\mathbf{u}_i)\mathbf{E}_{\mathcal{K}_2^i}^T\mathbf{\Psi}\mathbf{c}_i, \tag{25}$$

$$\mathbf{E}_{\mathcal{K}_3^i}^T\tilde{\mathbf{b}}_i = \text{diag}(\mathbf{E}_{\mathcal{K}_3^i}^T\mathbf{u}_i)\mathbf{E}_{\mathcal{K}_3^i}^T\mathbf{\Psi}\mathbf{c}_i. \tag{26}$$

If condition *a)* holds, then (24) is true for any $\mathbf{c}_i$ because both sides of the equality are 0. Since $\mathcal{K}_2^i$ contains no repeated eigenvalues, the matrix $\text{diag}(\mathbf{E}_{\mathcal{K}_2^i}^T\mathbf{u}_i)\mathbf{E}_{\mathcal{K}_2^i}^T\mathbf{\Psi}$ in (25) has full row-rank. Given that $|\mathcal{K}_2^i| \leq D$, condition *c)* ensures that $\mathbf{c}_i$ has at least as many elements as equations in (25), guaranteeing the existence of a solution. Denoting by $\mathbf{c}_i^*$ a solution of (25), condition *b)* implies that this same vector also solves (26). To see why this is true, notice that for every equation in (26) (of index k_3) there is one in (25) (of index k_2) such that $[\mathbf{\Psi}\mathbf{c}_i^*]_{k_3} = [\mathbf{\Psi}\mathbf{c}_i^*]_{k_2} = [\tilde{\mathbf{b}}_i]_{k_2}/[\mathbf{u}_i]_{k_2}$, where the last equality follows from the fact that $\mathbf{c}_i^*$ solves (25). Imposing condition *b)*, we arrive to the conclusion that $[\mathbf{\Psi}\mathbf{c}_i^*]_{k_3} = [\mathbf{b}_i]_{k_3}/[\mathbf{u}_i]_{k_3}$ and, thus, (26) is satisfied. The simultaneous fulfillment of (24), (25), and (26), implies (23), concluding the proof. ■

The conditions in Proposition 4 detail how the spectral properties of $\mathbf{S}$ impact the set of linear transformations that can be implemented. Condition *a)* states that if *node* i is unable to sense a specific frequency k, only linear operators whose i-th row is orthogonal to the k-th frequency basis vector can be implemented. Condition *b)* states that if two frequencies k_1 and k_2 are indistinguishable for the *graph-shift* operator, then the projection of the i-th row of $\mathbf{B}$ onto these two frequency basis vectors must be proportional to how strongly node i senses frequencies k_1 and k_2 for every node i. Finally, condition *c)* requires that the order of the *filter* has to be high enough to have enough degrees of freedom to design the linear operator and to allow the original signal $\mathbf{x}$ to percolate through the network. As was the case for Proposition 1, *a)* and *b)* are necessary conditions while *c)* details a sufficient filter degree for general implementation. However, filters with lower degree may be enough to implement particular linear transformations.

The following result follows as a corollary of Proposition 4.

Corollary 2 *Suppose that spectrum of the graph-shift operator* $\mathbf{S} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^{-1}$ *satisfies the following two properties:*
a) all the entries of $\mathbf{V}$ *are non-zero.*
b) all the eigenvalues $\{\lambda_k\}_{k=1}^N$ *are distinct.*
Then, it holds that any linear transformation $\mathbf{B}$ *can be implemented using a node-variant GF* $\mathbf{H}_{\text{nv}} = \mathbf{H}_{\text{nv}}(\{\mathbf{c}_i\}_{i=1}^N, \mathbf{S})$ *of the form given in* (7) *for* $L = N$*, with the set of filter coefficients* $\{\mathbf{c}_i\}_{i=1}^N$ *being unique and given by*

$$\mathbf{c}_i = \mathbf{\Psi}^{-1}\text{diag}(\mathbf{u}_i)^{-1}\mathbf{V}^T\mathbf{b}_i, \quad \text{for all } i. \tag{27}$$

Proof: Conditions *a)* and *b)* immediately guarantee the corresponding conditions in Proposition 4 for any $\mathbf{B}$. Condition *c)* in Proposition 4 is satisfied from the fact that the corollary sets $L = N$ and, by definition, $N \geq D$. Equation (27) follows directly after substituting $\tilde{\mathbf{b}}_i = \mathbf{V}^T\mathbf{b}_i$ into (23). Since conditions a) and b) guarantee that the full rank matrices $\text{diag}(\mathbf{u}_i)^{-1}$ and $\mathbf{\Psi}^{-1}$ exist, respectively, uniqueness also follows. ■

To gain some intuition on the design in (27), recall first that, as explained after (10), the output of a node-variant GF at node i (here $\mathbf{b}_i^T\mathbf{x}$) can be viewed as the inner product between $\mathbf{u}_i$ and $\widehat{\mathbf{x}}$ modulated by the frequency response of the GF $\widehat{\mathbf{c}}_i = \mathbf{\Psi}\mathbf{c}_i$. Then, comparing the expression in (27) to its counterpart for node-invariant GFs in (15), we observe that the design still tries to match the frequency response of $\mathbf{B}$ and $\mathbf{H}_{\text{nv}}$, but here in a per-node fashion; so that (27) replaces $\boldsymbol{\beta}$ (the spectrum of $\mathbf{B}$) with $\mathbf{V}^T\mathbf{b}_i$ and accounts for $\mathbf{u}_i$ (the effect of the particular node in the frequency domain, which was not present in (15)). More importantly, Proposition 4 and Corollary 2 confirm that the class of linear transformations that can be implemented using (7) is quite broad and significantly larger than the one that can be implemented by using (1). This is consistent with the fact that, as explained in Section II-B, node-variant GFs are more flexible operators since their number of free parameters (coefficients) is N times larger than those for node-invariant filters.

B. Approximate implementation

When the conditions in Proposition 4 are not satisfied, we resort again to approximate designs aimed at minimizing a pre-specified error metric. As done in Section III-B we minimize the MSE and the WCE under the assumption that the signal $\mathbf{x}$ is zero-mean and that its covariance $\mathbf{R}_{\mathbf{x}}$ is known. To be specific, recall that the difference (error) vector is defined as $\mathbf{d} = \mathbf{H}_{\text{nv}}\mathbf{x} - \mathbf{B}\mathbf{x}$ and its covariance matrix is denoted by $\mathbf{R}_{\mathbf{d}}$. Recall also that $\mathbf{C} = [\mathbf{c}_1,\dots,\mathbf{c}_N]$ and

define the matrices $\tilde{\mathbf{U}} := [\text{diag}(\mathbf{u}_1), \text{diag}(\mathbf{u}_2), \ldots, \text{diag}(\mathbf{u}_N)]^T$ and $\mathbf{\Phi}_i := (\mathbf{V}^{-1})^T \text{diag}(\mathbf{u}_i)\mathbf{\Psi}$. Then, the filter coefficients minimizing the MSE given by $\text{Trace}(\mathbf{R_d})$ and the WCE given by $\|\mathbf{R_d}\|_2$ can be found as specified in the following proposition, which reveals that the optimal MSE coefficients can be designed separately across nodes.

Proposition 5 *The optimal MSE coefficients* $\{\mathbf{c}^*_{i,\text{Tr}}\}_{i=1}^N := \text{argmin}_{\{\mathbf{c}_i\}_{i=1}^N} \text{Trace}(\mathbf{R_d})$ *are given by*

$$\mathbf{c}^*_{i,\text{Tr}} = (\mathbf{R}_\mathbf{x}^{1/2}\mathbf{\Phi}_i)^\dagger \mathbf{R}_\mathbf{x}^{1/2}\mathbf{b}_i \overset{\star}{=} (\mathbf{\Phi}_i^T \mathbf{R}_\mathbf{x}\mathbf{\Phi}_i)^{-1}\mathbf{\Phi}_i^T\mathbf{R}_\mathbf{x}\mathbf{b}_i \quad (28)$$

for all i, where $\overset{\star}{=}$ holds if $\mathbf{R}_\mathbf{x}^{1/2}\mathbf{\Phi}_i$ has full column rank; while the optimal WCE filter coefficients $\{\mathbf{c}^*_{i,\lambda}\}_{i=1}^N := \text{argmin}_{\{\mathbf{c}_i\}_{i=1}^N} \lambda_{\max}(\mathbf{R_d})$ *are found as*

$$\{\mathbf{C}^*_\lambda, s^*\} = \underset{\{\mathbf{C}, s\}}{\text{argmin}} \quad s \quad (29)$$

$$\text{s.to} \begin{bmatrix} s\mathbf{I} & (\mathbf{I}\odot\mathbf{\Psi C})^T\tilde{\mathbf{U}}\mathbf{V}^{-1} - \mathbf{B} \\ ((\mathbf{I}\odot\mathbf{\Psi C})^T\tilde{\mathbf{U}}\mathbf{V}^{-1} - \mathbf{B})^T & s\mathbf{R}_\mathbf{x}^{-1} \end{bmatrix} \succeq 0,$$

where $\odot$ denotes the Khatri-Rao product.

Proof: Leveraging equivalence (18), we may rewrite

$$\text{Trace}(\mathbf{R_d}) = \sum_{i=1}^N \|\mathbf{h}_i^T\mathbf{R}_\mathbf{x}^{1/2} - \mathbf{b}_i^T\mathbf{R}_\mathbf{x}^{1/2}\|_2^2. \quad (30)$$

From (10), we write $\mathbf{h}_i$ in terms of $\mathbf{c}_i$ as $\mathbf{h}_i^T = \mathbf{u}_i^T\text{diag}(\mathbf{\Psi c}_i)\mathbf{V}^{-1}$. By transposing both sides of the equation and recalling that for general vectors $\text{diag}(\mathbf{a})\mathbf{b} = \text{diag}(\mathbf{b})\mathbf{a}$, it follows that $\mathbf{h}_i = (\mathbf{V}^{-1})^T\text{diag}(\mathbf{u}_i)\mathbf{\Psi c}_i$. Using the definition of $\mathbf{\Phi}_i$ we obtain

$$\text{Trace}(\mathbf{R_d}) = \sum_{i=1}^N \|\mathbf{R}_\mathbf{x}^{1/2}\mathbf{\Phi}_i\mathbf{c}_i - \mathbf{R}_\mathbf{x}^{1/2}\mathbf{b}_i\|_2^2. \quad (31)$$

Since the i-th term in the summation depends on $\mathbf{c}_i$ and not on $\mathbf{c}_j$ for $i \neq j$, we may minimize the summands separately. The result in (28) follows from minimizing $\|\mathbf{R}_\mathbf{x}^{1/2}\mathbf{\Phi}_i\mathbf{c}_i - \mathbf{R}_\mathbf{x}^{1/2}\mathbf{b}_i\|_2^2$ via the Moore-Penrose pseudoinverse [31].

The proof to show (29) is analogous to that of Proposition 2, thus, we only need to show that $(\mathbf{I}\odot\mathbf{\Psi C})^T\tilde{\mathbf{U}}\mathbf{V}^{-1} = \mathbf{H}_\text{nv}$. To see why this is true, transpose (10) to obtain $\mathbf{h}_i = (\mathbf{V}^{-1})^T\text{diag}(\mathbf{u}_i)\mathbf{\Psi c}_i$, from where it follows that

$$[\mathbf{h}_1, ..., \mathbf{h}_N] = (\mathbf{V}^{-1})^T[\text{diag}(\mathbf{u}_1), ..., \text{diag}(\mathbf{u}_2)] \begin{bmatrix} \mathbf{\Psi c}_1 & \mathbf{0} & \cdots \\ \mathbf{0} & \mathbf{\Psi c}_2 & \\ \vdots & \vdots & \ddots \end{bmatrix}.$$

Noting that the rightmost matrix can be written as $\mathbf{I}\odot\mathbf{\Psi C}$ and that $\mathbf{H}_\text{nv} = [\mathbf{h}_1, ..., \mathbf{h}_N]^T$, the result follows. ■

The reason that allows optimizing the MSE coefficients of a given node separately from those of the other nodes is that the MSE is given by $\text{Trace}(\mathbf{R_d}) = \|\mathbf{H}_\text{nv}\mathbf{R}_\mathbf{x}^{1/2} - \mathbf{B}\mathbf{R}_\mathbf{x}^{1/2}\|_\text{F}^2$ [cf. (18)], which is an element-wise norm that can be decoupled across nodes. By contrast, if the objective is to optimize the WCE given by $\|\mathbf{R_d}\|_2$, then the WCE coefficients must be optimized jointly.

Naturally, there is a resemblance between the optimal designs in Proposition 5 and those for node-invariant GFs in Proposition 2. When comparing (17) with (29), we see that the expression $\mathbf{V}\text{diag}(\mathbf{\Psi c})\mathbf{V}^{-1}$ that explicitly states the dependence of a node-invariant filter on the filter coefficients $\mathbf{c}$, is replaced by the more involved expression $(\mathbf{I}\odot\mathbf{\Psi C})^T\tilde{\mathbf{U}}\mathbf{V}^{-1}$ stating the dependence of node-variant filters on the matrix $\mathbf{C}$, which collects the filter coefficients at every node [cf. (10)]. Regarding expressions (16) and (28), their main difference stems from the additional design flexibility of node-variant GFs. To be more precise, if (31) is augmented with $\mathbf{c}_i = \mathbf{c}_j$ for all nodes i and j, then the optimal coefficients boil down to those in (16). The fact that every node i can select different coefficients $\mathbf{c}_i$ decouples the cost in (31) ultimately leading to the optimal expression in (28).

As in Section III-B, when no prior information about $\mathbf{x}$ is available, the covariance matrix of the input in (28) and (29) is set to $\mathbf{R}_\mathbf{x} = \mathbf{I}$. The resultant coefficients $\{\mathbf{c}^*_{i,\text{Tr}}\}_{i=1}^N$ minimize the error metric $\|\mathbf{H}_\text{nv} - \mathbf{B}\|_\text{F}$, while $\{\mathbf{c}^*_{i,\lambda}\}_{i=1}^N$ are the minimizers of $\|\mathbf{H}_\text{nv} - \mathbf{B}\|_2$.

Although counterparts to the results in Section III-C could be derived here too, optimization of $\mathbf{S}$ in this case is not as relevant, since the role of $\mathbf{S}$ in the class of transformations that node-variant GFs can implement is marginal compared to that for node-invariant GFs (see Propositions 1 and 4).

V. Implementation of distributed operators

While the findings in Sections III and IV hold for any linear operator $\mathbf{B}$, our results, especially those related to node-variant GFs, are particularly relevant in the context of *distributed signal processing*. To demonstrate this, we specialize $\mathbf{B}$ to two operators commonly studied in *networked* setups: finite-time average consensus and analog network coding (ANC). Since the leniency of the consensus operator facilitates its implementation, the focus there is primarily on node-invariant GFs. As shown next, the more challenging ANC operators call for the implementation of node-variant GFs.

A. Finite-time average consensus

Define the $N \times N$ matrix $\mathbf{B}_\text{con} := \mathbf{1}\mathbf{1}^T/N$ and consider the case where the goal is to implement $\mathbf{B} = \mathbf{B}_\text{con}$. Clearly, the application of this transformation to a signal $\mathbf{x}$ yields the *consensus* signal $\mathbf{B}_\text{con}\mathbf{x} = \mathbf{1}(\mathbf{1}^T\mathbf{x})/N = \mathbf{1}\bar{x}$, i.e., a constant signal whose value is equal to $\bar{x} = N^{-1}\sum_{i=1}^N x_i$. The fact of $\mathbf{B}_\text{con}$ being a rank-one matrix can be leveraged to show the following corollary of Proposition 3.

Corollary 3 *The consensus transformation $\mathbf{B}_\text{con}$ can be written as a (node-invariant) filter $\sum_{l=0}^{N-1} c_l\mathbf{S}^l$ for some $\mathbf{S}$ associated with an undirected graph $\mathcal{G}$ if and only if $\mathcal{G}$ is connected.*

Although $\mathbf{B}_\text{con}$ is a particular case of the rank-one operators considered in Proposition 3, the result in Corollary 3 is stronger. The reason for this is that $\mathbf{B}_\text{con}$ is symmetric. As explained after Proposition 3, for symmetric transformations the proof can be modified to hold not only for spanning trees but for any connected subgraph. When the design for the shift in (19) is particularized to $\mathbf{B}_\text{con}$, the obtained $\mathbf{S}$ is a biased and scaled version of the Laplacian. To be more precise, if

we set $\mu = 1$ and chose as subgraph $\mathcal{T}$ the graph $\mathcal{G}$ itself, then the obtained shift is $\mathbf{S} = (\mathbf{I} + \mathbf{L})/N$, but any other selection for μ and $\mathcal{T}$ could also be used. Notice that an immediate consequence of Corollary 3 is that *if the shift can be selected*, consensus can be achieved in finite time for every connected undirected graph [22], [26], with $N-1$ being an upper bound on the number of local interactions needed for convergence. Compared to classical consensus algorithms that require infinite number of iterations, the price to pay here is that the values of $\{\lambda_k\}_{k=1}^N$ need to be known in order to compute $\mathbf{\Psi}$ in (15). If the selected $\mathbf{S}$ has *repeated eigenvalues*, then the required filter degree (number of exchanges) will be lower than $N-1$ (cf. Proposition 1.*c*). For setups where the upper bound on the number of allowed exchanges is less than $D-1$, the designs presented in Sections III-B and IV-B can be leveraged to minimize the approximation error. Another alternative is to design an $\mathbf{S}$ that, while respecting the support of $\mathcal{G}$, minimizes the number of required exchanges. Algorithms to carry out such an optimization can be obtained along the lines of (20), but their development is left as future work.

In setups where $\mathbf{S}$ cannot be selected, node-variant filters are a better alternative to approximate the consensus operator. Indeed, Corollary 2 guarantees error-free implementation for a broad class of shifts. Equally interesting, when $\mathbf{S} = \mathbf{I} + a\mathbf{L}$ with $a \neq 0$, it is not difficult to show that: i) for $L \leq D$ node-variant GFs outperform their node-invariant counterparts (numerical simulations show that the gain is moderate); and ii) for $L > D$ both types of filters perform equally (zero error) and, in fact, yield the same set of coefficients.

B. Analog network coding

In the context of multi-hop communication networks, network coding is a scheme where routing nodes, instead of simply relaying the received information, combine the packets (symbols) received from different sources to perform a single transmission. The general goal is to optimize the network flow while providing error-free communication [34]. Even though network coding was originally conceived for transmission of digital data in the form of packets [35], [36], extensions to the transmission of analog signals have been developed under the name of ANC [27], [28]. In this section, we show how GFs can be leveraged to design ANC protocols. Apart from traditional communication networks, the results presented here are also relevant for setups where there exists an inherent diffusion dynamics that percolates the information across the network as, for example, in the case of molecular and nano communication networks [37], [38].

To account for the particularities of ANC, the optimal GF-design framework presented in the previous sections has to be slightly modified. Up to this point, we have considered $\mathbf{B} \in \mathbb{R}^{N\times N}$ to be a desired transformation encompassing the *whole* set of nodes. However, in ANC we are interested in the transmission of information from sources to sinks that are, in general, a *subset* of the nodes in a graph. To be precise, denote by $\mathcal{S} := \{s_1, s_2, \ldots, s_S\}$ the set of S sources and by $\mathcal{R} := \{r_1, r_2, \ldots, r_R\}$ the set of R sinks or receivers. Since every source can have one or more receivers, we also define the

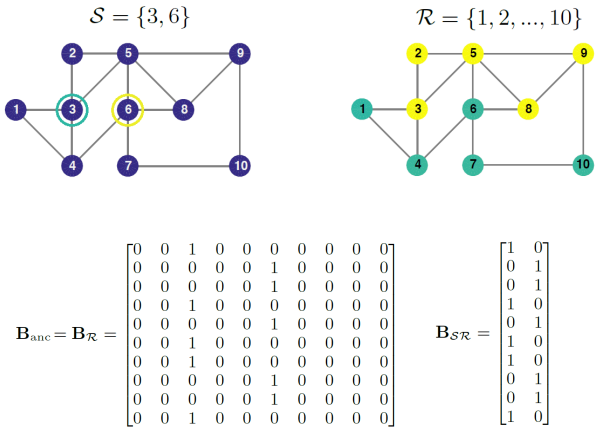


Fig. 1: Example of desired operators $\mathbf{B}_{\text{anc}}$, $\mathbf{B}_{\mathcal{R}}$ and $\mathbf{B}_{\mathcal{SR}}$ in the context of ANC. A graph with $N = 10$ nodes is considered. The sources are $n = 3$, whose destinations are $\{1, 4, 6, 7, 10\}$, and $n = 6$, whose destinations are $\{2, 3, 5, 8, 9\}$, so that every node in graph is a sink and thus $\mathcal{R} = \mathcal{N}$. The sources and destinations are identified in the top-left and top-right graphs, respectively. The corresponding full and reduced ANC matrices are provided at the bottom of the figure.

(surjective) function $s: \mathcal{R} \to \mathcal{S}$, which identifies the source for each receiver. In ANC one is interested in transformations $\mathbf{B} = \mathbf{B}_{\text{anc}}$ where the i-th row of $\mathbf{B}_{\text{anc}}$ is equal to the canonical vector $\mathbf{e}_{s(i)}^T$ for all $i \in \mathcal{R}$ (see Fig. 1 for an example). Since the values of $\mathbf{B}_{\text{anc}}$ for rows $i \notin \mathcal{R}$ are not relevant for the performance of ANC, one can define the reduced $R \times N$ matrix $\mathbf{B}_{\mathcal{R}} := \mathbf{E}_{\mathcal{R}}^T \mathbf{B}_{\text{anc}}$, where $\mathbf{E}_{\mathcal{R}} := [\mathbf{e}_{r_1}, \ldots, \mathbf{e}_{r_R}]$. Hence, the goal of ANC boils down to designing a filter $\mathbf{H}$ such that $\mathbf{E}_{\mathcal{R}}^T \mathbf{H}$ is as close to $\mathbf{B}_{\mathcal{R}}$ as possible. Most ANC setups consider that only source nodes have signals to be transmitted, so that the input signal at all other nodes can be considered zero. The GF design can leverage this fact to yield a better approximation. Indeed, it is easy to see that if the input signal for nodes $i \notin \mathcal{S}$ is zero, the values of the corresponding columns in $\mathbf{B}_{\mathcal{R}}$ are irrelevant. Hence, upon defining matrices $\mathbf{E}_{\mathcal{S}} := [\mathbf{e}_{s_1}, \ldots, \mathbf{e}_{s_S}]$ and $\mathbf{B}_{\mathcal{SR}} := \mathbf{B}_{\mathcal{R}} \mathbf{E}_{\mathcal{S}} \in \mathbb{R}^{S\times R}$, the goal for ANC is to design a filter $\mathbf{H}$ such that $\mathbf{E}_{\mathcal{R}}^T \mathbf{H} \mathbf{E}_{\mathcal{S}}$ is as close to $\mathbf{B}_{\mathcal{SR}}$ as possible. To illustrate why designing $\mathbf{B}_{\mathcal{SR}}$ is easier, consider the example in Fig. 1. While perfect implementation of $\mathbf{B}$ requires tuning the 100 entries of $\mathbf{H}$, implementation of $\mathbf{B}_{\mathcal{SR}}$ requires only fixing 20 entries (those corresponding to the 3rd and 6th columns).

Although the propositions presented throughout the paper need to be modified to accommodate the introduction of $\mathbf{B}_{\mathcal{R}}$ and $\mathbf{B}_{\mathcal{SR}}$, the main results and the structure of the proofs remain the same. To be specific, consider first the case where the nodes that are not sources do not inject any input, so that the goal is to approximate the $S \times R$ matrix $\mathbf{B}_{\mathcal{SR}}$. Then, defining the $SR \times L$ matrix $\Theta_{\mathcal{SR}} := [\text{vec}(\mathbf{E}_{\mathcal{R}}^T \mathbf{I} \mathbf{E}_{\mathcal{S}}), \ldots, \text{vec}(\mathbf{E}_{\mathcal{R}}^T \mathbf{S}^{L-1} \mathbf{E}_{\mathcal{S}})]$, we can find the coefficients that minimize $\|\mathbf{E}_{\mathcal{R}}^T \mathbf{H} \mathbf{E}_{\mathcal{S}} - \mathbf{B}_{\mathcal{SR}}\|_{\text{F}}$ as [cf. (16)]

$$\mathbf{c}_{\text{F}}^* = \mathbf{\Theta}_{\mathcal{SR}}^{\dagger} \text{vec}(\mathbf{B}_{\mathcal{SR}}) \overset{\star}{=} (\mathbf{\Theta}_{\mathcal{SR}}^T \mathbf{\Theta}_{\mathcal{SR}})^{-1} \mathbf{\Theta}_{\mathcal{SR}}^T \text{vec}(\mathbf{B}_{\mathcal{SR}}), \quad (32)$$

where $\overset{\star}{=}$ holds if $\mathbf{\Theta}_{\mathcal{SR}}$ has full column rank. Similarly, by defining $\mathbf{\Phi}_{r_i,\mathcal{S}} := \mathbf{E}_{\mathcal{S}}^T \mathbf{\Phi}_{r_i}$ and denoting the i-th row of $\mathbf{B}_{\mathcal{SR}}$

as $\mathbf{b}_{i,\mathcal{S}}^T$, we may obtain the node-variant counterpart of (32) as [cf. (28)]

$$\mathbf{c}_{r_i,\mathrm{F}}^* = \mathbf{\Phi}_{r_i,\mathcal{S}}^\dagger \mathbf{b}_{i,\mathcal{S}} \stackrel{\star}{=} (\mathbf{\Phi}_{r_i,\mathcal{S}}^T \mathbf{\Phi}_{r_i,\mathcal{S}})^{-1} \mathbf{\Phi}_{r_i,\mathcal{S}}^T \mathbf{b}_{i,\mathcal{S}}, \qquad (33)$$

for all $r_i \in \mathcal{R}$, where $\stackrel{\star}{=}$ holds if $\mathbf{\Phi}_{r_i,\mathcal{S}}$ has full column rank. For a given filter length, the L coefficients in $\mathbf{c}_{r_i,\mathrm{F}}^*$ specify the optimal weights that the sink node r_i must give to the original signal and the first $L-1$ shifted versions of it to resemble as close as possible (in terms of MSE) the desired linear combination of source signals $\mathbf{b}_{i,\mathcal{S}}$.

When the initial signal at the routing nodes cannot be assumed zero, the goal is to approximate the matrix $\mathbf{B}_\mathcal{R} \in \mathbb{R}^{R\times N}$. In that case, the previous expressions for the optimal filter coefficients still hold true. The only required modification is to substitute $\mathbf{B}_{\mathcal{SR}} = \mathbf{B}_\mathcal{R}$ and $\mathbf{E}_\mathcal{S} = \mathbf{I} \in \mathbb{R}^{N\times N}$ into the definitions of $\mathbf{\Theta}_{\mathcal{SR}}$, $\mathbf{\Phi}_{r_i,\mathcal{S}}$ and $\mathbf{b}_{i,\mathcal{S}}$. Indeed, when every node acts as both a source and a sink: i) $\mathbf{B}_\mathcal{R}$ is an $N \times N$ matrix equal to $\mathbf{B}_{\mathrm{anc}}$; and ii) the definitions of $\mathbf{\Theta}_{\mathcal{SR}}$ and $\mathbf{\Phi}_{r_i,\mathcal{S}}$ require setting $\mathbf{E}_\mathcal{S} = \mathbf{E}_\mathcal{R} = \mathbf{I}$. This readily implies that (32) and (33) reduce to their original counterparts (16) and (28), respectively.

Although not presented here, expressions analogue to those in (32) and (33) for the remaining optimal filter-design criteria can be derived too.

VI. Numerical experiments

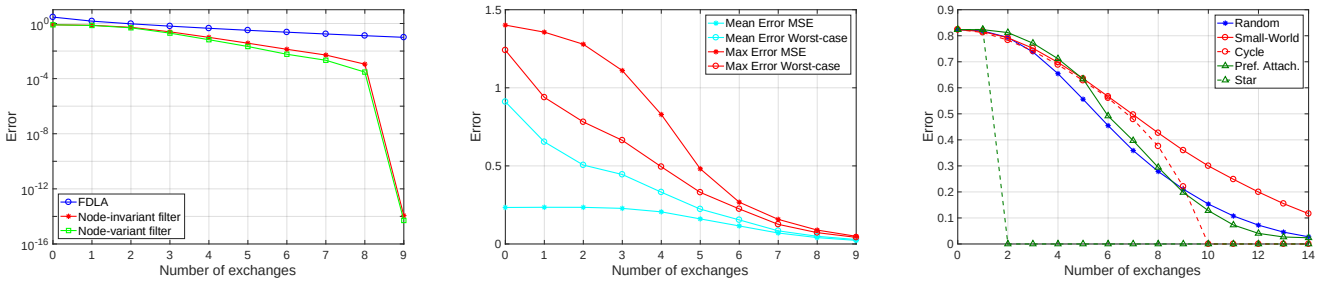
This section is structured in three parts. The first two focus on illustrating the results in Sections III and IV using as examples the two applications introduced in Section V. The last part assesses the approximation error when the eigenvectors of $\mathbf{S}$ and $\mathbf{B}$ are different, which according to Proposition 1 is the critical factor for approximation performance. The results presented here complement and confirm the preliminary findings reported in [1] and [2] for other types of graphs.

A. Finite-time average consensus

To demonstrate the practical interest of the proposed framework, we present numerical experiments comparing the performance achieved by three different consensus implementations: our design as a node-invariant GF, its node-variant counterpart, and the asymptotic fastest distributed linear averaging (FDLA) in [12]. In order to assess the performance of the approximation algorithms, we generate 1,000 unweighted symmetric small-world graphs [39] with $N = 10$ nodes, where each node is originally connected in average to four of its neighbors, and a rewiring probability of 0.2 For each graph, we define the graph-shift operator $\mathbf{S} = \mathbf{W}$ where $\mathbf{W}$ is the solution of the FDLA problem, i.e, $\lim_{l\to\infty} \mathbf{W}^l = \mathbf{B}_{\mathrm{con}}$ with fastest convergence. Moreover, on each graph we define a signal $\mathbf{x}$ drawn from a standard multivariate Gaussian distribution (smGd). For a fixed number K of local interactions, we define the FDLA approximation as $\mathbf{x}_{\mathrm{FDLA}}^{(K)} := \mathbf{W}^K\mathbf{x}$, the node-invariant GF approximation as $\mathbf{x}_{\mathrm{GF}}^{(K)} := \sum_{l=0}^K c_l^* \mathbf{W}^l\mathbf{x}$ and the node-variant GF approximation as $\mathbf{x}_{\mathrm{NVGF}}^{(K)} := \sum_{l=0}^K \mathrm{diag}(\mathbf{c}^{(l)*})\mathbf{W}^l\mathbf{x}$. The optimal coefficients c_l^* and $\mathbf{c}^{(l)*}$, which change with K, are obtained for all l using (16) and (28), respectively. Further, we define the error $e_{\mathrm{GF}}^{(K)} = \|\mathbf{x}_{\mathrm{GF}}^{(K)} - \mathbf{B}_{\mathrm{con}}\mathbf{x}\|_2$ and similarly for $e_{\mathrm{FDLA}}^{(K)}$ and $e_{\mathrm{NVGF}}^{(K)}$. Fig. 2a plots the errors averaged over the 1,000 graphs as a function of the number of local interactions ($K = L-1$) among neighbors. The error attained by the GF approaches is around one order of magnitude lower than that of the asymptotic approach for intermediate number of interactions and, when $K = N-1 = 9$, perfect recovery is achieved using GFs (cf. Corollary 3). The performance improvement attained by the GFs is due to the fact that, for a fixed K, FDLA returns the value of $\mathbf{W}^K\mathbf{x}$ whereas the GFs return an optimal linear combination of all $\mathbf{W}^l\mathbf{x}$ for $0 \le l \le K$. Notice that node-variant GFs, being a generalization of node-invariant filters, are guaranteed to achieve a lower error. Nevertheless, the additional error reduction achieved by node-variant GFs is minimal, due to the fact that $\mathbf{B}_{\mathrm{con}}$ can be perfectly implemented using node-invariant filters.

We now focus the analysis on node-invariant filters and illustrate the difference between the MSE and the WCE minimizations (cf. Proposition 2). To this end, we generate an unweighted and symmetric scale-free graph [39] with $N = 40$ nodes from a preferential attachment dynamic with four initial nodes and where each new node establishes two edges with the existing graph. On this graph, we define 1,000,000 signals $\mathbf{x}$ drawn from an smGd. For a fixed filter degree K, we compute the optimal MSE filter $\mathbf{H}_{\mathrm{MSE}}^{(K)}$ using (16), and the optimal WCE filter $\mathbf{H}_{\mathrm{WCE}}^{(K)}$ using (17). We then define the error $e_{\mathrm{MSE}}^{(K)} := ||\mathbf{H}_{\mathrm{MSE}}^{(K)}\mathbf{x} - \mathbf{B}_{\mathrm{con}}\mathbf{x}||_2$ and similarly for $e_{\mathrm{WCE}}^{(K)}$. In Fig. 2b we plot the average of these errors across the 1,000,000 realizations as well as their maximum as a function of the filter degree. If we focus on the average error, it is immediate to see that, as expected, the MSE approach outperforms the WCE. On the other hand, if we consider the maximum error across realizations, $\mathbf{H}_{\mathrm{WCE}}^{(K)}$ attains the lowest error. Notice that for $K \ge 9$, the mean and maximum errors achieved by both approaches are negligible, even though the degree K is markedly smaller than $N-1 = 39$.

To consider different graph models, the last set of simulations includes Erdős-Rényi *random* graphs [39] as well as the *deterministic* star and cycle graphs. Regarding the random graphs, the error performance is assessed by generating 1,000 instances of 20-node graphs for each class and setting $\mathbf{S} = \mathbf{L}$. To obtain graphs with comparable number of edges, for the Erdős-Rényi graphs we set $p_{edge} = 0.1$, we obtain the small-world graphs by rewiring with probability 0.2 the edges in a cycle, and we generate the scale-free graphs from a preferential attachment dynamic where each new node establishes *one* edge with the existing graph. In Fig. 2c we illustrate the mean error across 1,000 realizations as a function of the filter degree K. The approximation errors attained for random (Erdős-Rényi) and scale-free (preferential attachment) graphs are comparable, with the former showing slightly better performance for small values of K while the opposite being true for large values of K. Moreover, the approximation errors obtained for small-world graphs are consistently larger than those for the other two types of graphs. This can be in part explained by the fact that, for the simulation setup considered, the average diameters for the Erdős-Rényi and the scale-free graphs are



(a) Different types of filters (b) Different coefficient-design criteria (c) Different underlying graphs

Fig. 2: (a) Mean approximation error for consensus across 1,000 realizations in 10-node small-world graphs. GF approaches outperform the best asymptotic solution and achieve perfect consensus for degree $N-1=9$. (b) Mean and maximum errors obtained across 1,000,000 signal realizations for a 40-node scale-free graph and two different node-invariant GFs. One of the filters attains lower average error and the other attains lower maximum error, as expected. (c) Mean approximation error for consensus across 1,000 realizations of 20-node Erdős-Rényi, small-world, and scale-free graphs. Errors for star and cycle graphs are also shown.

7.0 and 6.1, respectively, whereas the average diameter for the small-world graphs is 10.9. Fig. 2c also shows the consensus approximation error for the star and the cycle, which are deterministic graphs. Notice that the star can be understood as the limit of preferential attachment graphs, where every new node attaches to the existing node of largest degree with probability 1. Furthermore, the cycle can be seen as the limit of a small-world graph, where the probability of rewiring is 0. The star graph achieves zero error after two information exchanges between neighbors. Similarly, the cycle achieves zero error after $10=N/2$ local interactions between neighbors. This implies that, for the star and the cycle graphs, perfect consensus is achieved for a number of exchanges equal to the diameter of the graph [22], which is a trivial lower bound for perfect recovery since information must travel from each node to every other node for consensus to be achievable.

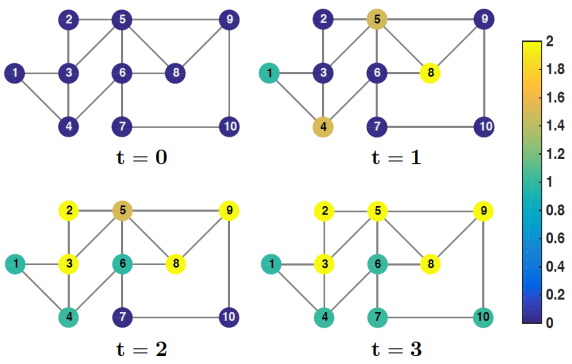
B. Analog network coding

To assess the performance obtained when node-variant GFs are designed to implement ANC schemes, we consider the example described in Fig. 1, where we recall that node 3 wants to transmit its signal to nodes 1, 4, 6, 7, and 10, while node 6 wants its signal to be transmitted to the remaining five nodes. The corresponding desired matrix $\mathbf{B}_{\mathcal{SR}}$ is shown in the figure too. For this particular example, every sink node wants to recover either one source signal *or* the other, but the same model could accommodate the case where sink nodes seek to recover a linear combination of the source signals. Denote by g (green) the signal injected by source node 3 and by w (yellow) the one injected by node 6, and set $\mathbf{S}=\mathbf{A}$. Fig. 3a summarizes the signal to be recovered at each node as well as the initially injected signal $\mathbf{z}^{(0)}$ and its first three shifted versions. In Fig. 3b we specialize this problem for the case where the green signal g and the yellow signal w are equal to 1 and 2, respectively. Moreover, we use the node colors to represent the successive best approximations of the signal to be recovered at each node, obtained using the optimal filter coefficients in (33). For example, at time $t=0$, we have that 8 of the nodes have only observed a null signal and nodes 3 and 6 have observed their own injected signal which is different from the signal they want to recover. Thus, none of the nodes has information about the signal they want to recover and the optimal estimate is 0 for all of them. However, after one shift ($t=1$) we see that four nodes have changed their estimates. The optimal coefficients $\mathbf{c}^*_{1,\mathrm{F}}=\mathbf{c}^*_{8,\mathrm{F}}=[0,1]^T$ allow nodes 1 and 8 to perfectly recover the desired signal, as can be seen from the corresponding rows of the table in Fig. 3a. For example, node 1 perfectly recovers g by computing $\mathbf{c}^{*T}_{1,\mathrm{F}}\,[0,g]^T=g$. The optimal coefficients for nodes 4 and 5 at $t=1$ are $\mathbf{c}^*_{4,\mathrm{F}}=\mathbf{c}^*_{5,\mathrm{F}}=[0,0.5]^T$ since their observation after one shift is the sum of the signal they want to recover and the other input signal. At time $t=2$, seven of the sink nodes perfectly recover the desired signal. E.g., node 3 applies the optimal coefficients $\mathbf{c}^*_{3,\mathrm{F}}=[-2,0,0.5]^T$, which yield the signal $\mathbf{c}^{*T}_{3,\mathrm{F}}\,[g,0,4g+2w]^T=w$ as can be seen from the corresponding row in Fig. 3a. By contrast, nodes 7 and 10 have not observed any information related to their desired signal and node 5 still cannot decode w since it observed $g+w$ twice. Finally, at time $t=3$ every node can successfully decode the objective signal. Fig. 3b demonstrates that, through the optimal design of node-variant filters, every node can recover its target signal after three information exchanges with neighbors. Notice that three is also the smallest number of exchanges that can yield perfect recovery since sink nodes 7 and 10 are *three* hops away from their associated source node 3. Moreover, note that the communication protocol induced by the shift operator $\mathbf{S}=\mathbf{A}$ is extremely simple: each node forwards its current signal to its neighbors and, in turn, receives the sum of their signals.

The following experiments evaluate the ensemble performance by averaging the results of multiple tests. To that end, we consider Erdős-Rényi graphs with $N=100$, $p_{edge}=0.1$ and weights drawn from a uniform distribution with support $[0.5,1.5]$, and $\mathbf{S}=\mathbf{A}$. We randomly select $S=5$ sources and, to each of these sources, we assign a sink so that $R=5$ and $\mathbf{B}_{\mathcal{SR}}=\mathbf{I}$. The goal is to evaluate the performance gap when approximating $\mathbf{B}_{\mathcal{SR}}$ vs. $\mathbf{B}_{\mathcal{R}}$. Recall that the former assumes that the nodes that are not sources inject a zero input. Alternatively, in the latter each receiver only knows its intended source, so that it must annihilate the signal from all other nodes. Since the size of $\mathbf{B}_{\mathcal{R}}$ is N/S times larger than the size of $\mathbf{B}_{\mathcal{SR}}$, the performance of the former is expected to be considerably lower. To corroborate this, Fig. 4a plots the

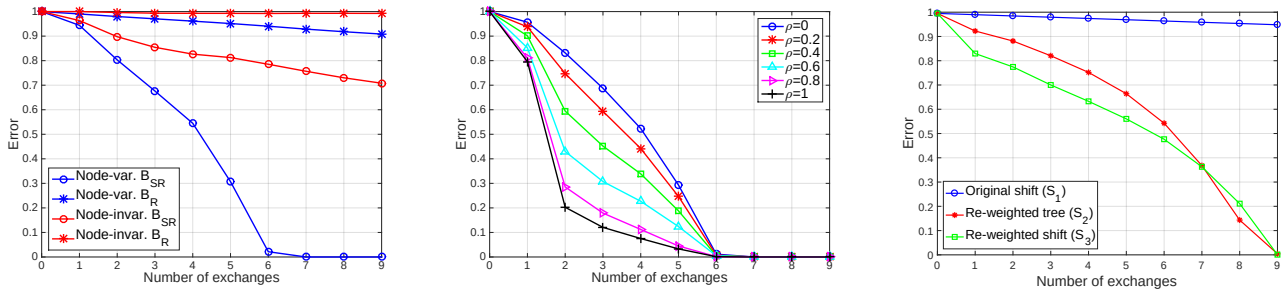
node	recover signal	observed signal $\mathbf{z}^{(t)}$ $t=0$	$t=1$	$t=2$	$t=3$
1	g	0	g	$g+w$	$5g+2w$
2	w	0	g	$g+w$	$5g+3w$
3	w	g	0	$4g+2w$	$4g+3w$
4	g	0	$g+w$	g	$7g+7w$
5	w	0	$g+w$	$g+w$	$5g+8w$
6	g	w	0	$2g+4w$	$3g+2w$
7	g	0	w	0	$2g+5w$
8	w	0	w	$g+w$	$4g+7w$
9	w	0	0	$g+2w$	$2g+3w$
10	g	0	0	w	$g+2w$

(a) Observed signal for $t=0,..,3$



(b) Best estimate of the target signal for $t=0,..,3$

Fig. 3: Example of design of a node-variant GF for the ANC setup described in Fig. 1. (a) Signal to recover at each sink node (g stands for green and w for yellow) and observed shifted signals for $t \in \{0,1,2,3\}$. (b) Optimal estimates at sink nodes shown by node colors for different times. After three interactions with neighbors ($t=3$), the bottom-right graph shows that every sink can decode its target signal.



(a) ANC uncorrelated sources (b) ANC correlated sources (c) Rank-one transformations

Fig. 4: Approximation errors as a function of the filter degree based on 1,000 realizations and Erdős-Rényi random graphs. (a) ANC with $N=100$, $S=R=5$ sources and sinks. Curves represent mean recovery error for node-variant and node-invariant GFs when the coefficients are designed to approximate $\mathbf{B}_{\mathcal{R}}$ and $\mathbf{B}_{\mathcal{SR}}$. (b) Same as in (a) for node-variant GFs and different levels of correlation ρ among the values injected at the different sources. (c) Rank-one transformations as node-invariant GFs for three different choices of graph-shift operators.

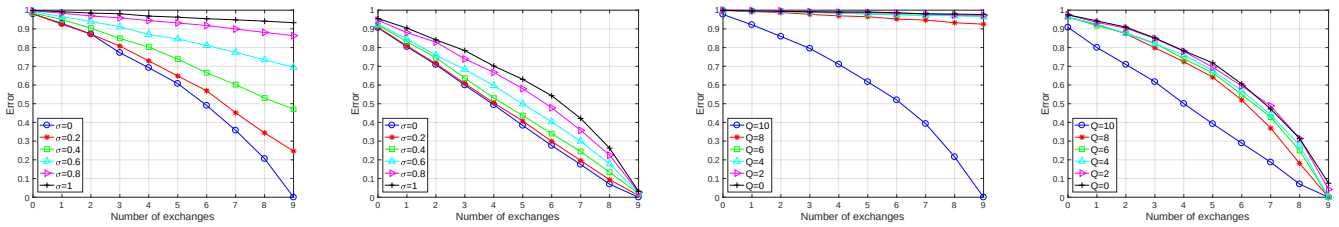
mean error across 1,000 graphs for node-invariant (red) and node-variant (blue) filters when the coefficients are designed to approximate both $\mathbf{B}_{\mathcal{R}}$ and $\mathbf{B}_{\mathcal{SR}}$. Denoting by $\mathbf{x}$ the S-sparse input signal containing the values to be transmitted by the source nodes (drawn from an smGd) and by $\mathbf{y}=\mathbf{H}\mathbf{x}$ the filtered signal, the error is defined as $e := \|\mathbf{E}_{\mathcal{R}}\mathbf{y} - \mathbf{E}_{\mathcal{S}}\mathbf{x}\|_2/\|\mathbf{E}_{\mathcal{S}}\mathbf{x}\|_2$, i.e. the normalized difference between the signal injected at the sources and the one recovered at the sinks. Notice that if the sources are known and the relay nodes inject a zero signal, after 7 local exchanges the mean error for node-variant filters is 0. For this same filter degree the error when approximating $\mathbf{B}_{\mathcal{R}}$ is 0.93. Eventually, the latter error also vanishes, but filters of degree close to $N=100$ are needed. By contrast, when node-invariant GFs are used to approximate $\mathbf{B}_{\mathcal{R}}$, the error improvement associated with increasing filter degree for the selected interval ($0 \leq L-1 \leq 9$) is negligible. Finally, by comparing the plots for node-variant and node-invariant GFs when the coefficients are designed to approximate $\mathbf{B}_{\mathcal{SR}}$, we observe that node-variant GFs achieve zero error after a few interactions, while node-invariant GFs exhibit a slow reduction of the error with the filter degree.

Correlation among the injected signals at source nodes can be leveraged to reduce the error at the receivers. To illustrate this, for different values of $\rho \in \{0, 0.2, \ldots, 1\}$, we build an $S \times S$ covariance matrix $\mathbf{R}_{\mathbf{x},\rho}$ defined as $\mathbf{R}_{\mathbf{x},\rho}^{1/2} := \mathbf{I} + \rho(\mathbf{1}\mathbf{1}^T - \mathbf{I}) + 0.1\rho\mathbf{Z}$ where the elements in the *symmetric* matrix $\mathbf{Z}$ are drawn from an smGd. In this way, the correlation between injected signals increases with ρ and, in each realization, is corrupted by additive zero-mean random noise. In Fig. 4b we plot the mean error across 1,000 graphs as a function of the node-variant filter degree parametrized by ρ. Notice first that when $\rho=0$, the values injected at different sources are uncorrelated and we recover the blue plot in Fig. 4a. More interestingly, as ρ increases, the achieved error for a given filter degree markedly decreases. The reason for this is that correlation allows sink nodes to form accurate estimates of their objective source signals based on *all* injected signals.

C. Optimal graph-shift operator design

While the previous experiments investigated the design of the filter coefficients for a given shift, the focus of this section is on designing the weights of the shift itself. Specifically, we test the results in Section III-C regarding the implementation of rank-one transformations $\mathbf{B}$ using node-invariant GFs.

To perform these tests, we generate Erdős-Rényi graphs $\mathcal{G}$ with $N=10$ nodes and edge probability $p_{\text{edge}}=0.3$, and rank-one transformations $\mathbf{B}=\mathbf{a}\mathbf{b}^T$ where $\mathbf{a}$ and $\mathbf{b}$ are drawn from an smGd. For each graph we consider three different shift



(a) Node invariant - noisy $\mathbf{V}$ (b) Node variant - noisy $\mathbf{V}$ (c) Node invariant - incomplete $\mathbf{V}$ (d) Node variant - incomplete $\mathbf{V}$

Fig. 5: Approximation errors as a function of the filter degree based on 1,000 realizations and small-world graphs. (a) Spectrum mismatch for a node-invariant GF with $N = 10$ and symmetric shift. Curves represent median recovery error parametrized by the eigenbasis perturbation magnitude σ. (b) Same as in (a) but for node-variant GFs. (c) Same as in (a) but parametrized by Q, the number of shared eigenvectors between $\mathbf{S}$ and $\mathbf{B}$. (d) Same as in (c) but for node-variant GFs.

operators: i) the *original shift* $\mathbf{S}_1$, where we assign random weights to every edge present in the graph and its diagonal; ii) the *re-weighted tree* $\mathbf{S}_2$, where we randomly select a spanning tree of $\mathcal{G}$ and then set the weights as dictated by (19); and iii) the *re-weighted shift* $\mathbf{S}_3$ where we use (19) to select weights for *all* the edges in $\mathcal{G}$, as opposed to just those forming a spanning tree. We test the performance of approximating $\mathbf{B}$ as node-invariant filters $\mathbf{H}(\mathbf{c}, \mathbf{S}_m)$ for $m \in \{1, 2, 3\}$ and for varying filter degrees. Specifically, for each filter degree or number of exchanges $K \in \{0, 1, \ldots, 9\}$ we obtain the optimal GF coefficients $\mathbf{c}_m^K$ for shift $\mathbf{S}_m$ by solving (16), where $\mathbf{R}_\mathbf{x} = \mathbf{I}$. We then measure the error as $\|\mathbf{B} - \mathbf{H}(\mathbf{c}_m^K, \mathbf{S}_m)\|_\mathrm{F}/\|\mathbf{B}\|_\mathrm{F}$ and plot its average across $1,000$ realizations in Fig. 4c.

GFs based on the original shift $\mathbf{S}_1$ approximate poorly the rank-one transformations, achieving a large error of around 0.95 even for filters of degree 9. This is not surprising since, in general, $\mathbf{S}_1$ and $\mathbf{B}$ will not be simultaneously diagonalizable (cf. Proposition 1) and node-invariant filters are very sensitive to spectral misalignments (cf. Section VI-D). The tree-based shift $\mathbf{S}_2$ can perfectly implement $\mathbf{B}$ for a large enough degree, confirming the result shown in Proposition 3. Equally interesting, the full-graph shift $\mathbf{S}_3$ also attains zero error. Recall that, for simplicity, the constructive proof of Proposition 3 that guarantees zero-error was based on spanning trees. In line with the discussion following Proposition 3, the simulations support the presumption that the construction is also valid for more general graphs. Note finally that, when comparing the performance of $\mathbf{S}_2$ and $\mathbf{S}_3$, the latter tends to outperform the tree-based design. This is somehow expected since the larger number of links in $\mathbf{S}_3$ facilitates the quick propagation of information across the graph. Indeed, as K increases and approaches the diameter of the tree both designs perform similarly.

D. Spectral robustness of GF implementations

A limitation of *node-invariant* GFs is that perfect implementation of a transformation $\mathbf{B}$ requires $\mathbf{S}$ and $\mathbf{B}$ to share the whole set of eigenvectors. In this section, we run numerical simulations to assess: i) how the implementation degrades as the eigenvectors of $\mathbf{S}$ and $\mathbf{B}$ are further apart; and ii) whether the implementation using *node-variant* filters is robust to this dissimilarity. For each experiment, we generate 1,000 unweighted and symmetric small-world graphs with $N = 10$, where each node is originally connected in average to four neighbors, and rewiring probability of 0.2. We set $\mathbf{S} = \mathbf{A}$, and define a signal $\mathbf{x}$ drawn from an smGd on each graph. For a given degree, the filter coefficients are obtained using the optimal designs in (16) and (28).

Two procedures are considered to construct linear transformations $\mathbf{B}$ whose eigenvectors have varying degrees of similarity with those of $\mathbf{S} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^H$. In the first one, we generate perturbed eigenbases from $\mathbf{V}$ by adding elementwise zero-mean Gaussian noise of varying power. Specifically, we build $\mathbf{B} = \mathbf{V}_\mathbf{B}^\sigma \mathrm{diag}(\boldsymbol{\beta})(\mathbf{V}_\mathbf{B}^\sigma)^{-1}$ with the eigenvalues $\boldsymbol{\beta}$ drawn from an smGd and the eigenvectors $\mathbf{V}_\mathbf{B}^\sigma$ obtained by normalizing the columns of $\mathbf{V} + \sigma\mathbf{Z} \circ \mathbf{V}$, where the elements in the $N \times N$ matrix $\mathbf{Z}$ also follow an smGd and $\circ$ denotes the Hadamard matrix product. Defining the error as $e := \|\mathbf{H}\mathbf{x} - \mathbf{B}\mathbf{x}\|_2/\|\mathbf{B}\mathbf{x}\|_2$, in Fig. 5a we present the median error achieved by node-invariant filters when approximating $\mathbf{B}$. Each curve corresponds to a different value of σ. For the case where $\sigma = 0$, $\mathbf{S}$ and $\mathbf{B}$ are simultaneously diagonalizable and, hence, perfect implementation is achievable after $N - 1$ exchanges among neighbors. As expected the error increases with σ, e.g., when $\sigma = 0.2$ the best achievable error is around 0.25 for filters of degree 9. Part of this insurmountable error comes from the fact that $\mathbf{B}$ is, in general, asymmetric while node-invariant filters $\mathbf{H}$ inherit the symmetry from $\mathbf{S}$. By contrast, node-variant filters can successfully implement $\mathbf{B}$ for all values of σ; see Fig. 5b. Even though there is a slight dependence with σ, all curves demonstrate that the error decreases with the number of exchanges until almost perfect implementation is achieved for $K = N-1$. The second procedure keeps only a subset of the eigenvectors of the shift. More specifically, we build linear operators of the form $\mathbf{B} = \mathbf{V}_\mathbf{B}^Q \mathrm{diag}(\boldsymbol{\beta})(\mathbf{V}_\mathbf{B}^Q)^{-1}$ where Q of the columns of matrix $\mathbf{V}_\mathbf{B}^Q$ are chosen randomly without replacement from $\mathbf{V}$. The remaining $N - Q$ columns correspond to random linearly independent, non-orthogonal, vectors. In this way, the resulting $\mathbf{V}_\mathbf{B}^Q$ is not unitary and $\mathbf{B}$ is non-symmetric (symmetry will require setting the $N - Q$ vectors as an orthonormal basis of the orthogonal complement of the space spanned by the Q columns selected from $\mathbf{V}$). In Fig. 5c we present the median error attained by node-invariant filters parametrized by Q. As expected, when $Q = 10$, perfect implementation is achieved since $\mathbf{B}$ and $\mathbf{S}$ are simultaneously diagonalizable. As

Q decreases, a noticeable detrimental effect is observed. The reason being that node-invariant GFs with symmetric shifts can give rise only to symmetric transformations, and $\mathbf{B}$ here is non-symmetric. Indeed, if the remaining $N-Q$ columns of $\mathbf{V}_{\mathbf{B}}^{Q}$ are chosen such that $\mathbf{V}_{\mathbf{B}}^{Q}$ is unitary (and, hence, $\mathbf{B}$ is symmetric), the approximation error for intermediate values of K is considerably smaller (not shown here). By contrast, node-variant filters show imperviousness to the degree of overlap Q between the eigenbases of $\mathbf{S}$ and $\mathbf{B}$; see Fig. 5d.

In a nutshell, the performance of *node-invariant* GFs depends noticeably on the similarity between the eigenvectors of $\mathbf{S}$ and $\mathbf{B}$. The approximation error grows gradually with the distance between eigenbases, and larger errors are observed when approximating asymmetric operators with symmetric shifts. This behavior was also observed in [1] for Erdős-Rényi graphs. By contrast, *node-variant* GFs are robust to the spectral differences between $\mathbf{S}$ and $\mathbf{B}$. A theoretical characterization of the described robustness as well as an analysis of its dependence on topological features of the underlying graph are interesting research directions and are left as future work.

VII. Conclusions

The optimal design of GFs for (potentially distributed) implementation of generic linear operators was investigated. Our first contribution was the introduction of two types of node-variant GFs, which generalize the existing definition of (node-invariant) GFs by allowing each node to use a different set of filter coefficients. Then, for both node-invariant and node-variant filters, we stated conditions for perfect implementation of linear operators. When these conditions were not met, we provided optimal filter-coefficient designs for the minimization of different error metrics. We also addressed the optimal design of the shift operator and provided a systematic way to define a shift that allows the implementation of any rank-one operator as a GF. The practical relevance of our approach for *distributed* setups was emphasized by particularizing our results to the design of GFs able to implement finite-time consensus and analog network coding, two problems commonly studied in network settings.

References

[1] S. Segarra, A. G. Marques, and A. Ribeiro, "Distributed implementation of linear network operators using graph filters," in *53rd Allerton Conf. on Commun. Control and Computing*, Sept. 2015, pp. 1406–1413.

[2] ——, "Linear network operators using node-variant graph filters," in *IEEE Intl. Conf. Acoust., Speech and Signal Process. (ICASSP)*, Mar. 2016, pp. 4850–4854.

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

[4] A. Sandryhaila and J. Moura, "Discrete signal processing on graphs," *IEEE Trans. Signal Process.*, vol. 61, no. 7, pp. 1644–1656, Apr. 2013.

[5] S. Chen, R. Varma, A. Sandryhaila, and J. Kovačević, "Discrete signal processing on graphs: Sampling theory," *IEEE Trans. Signal Process.*, vol. 63, no. 24, pp. 6510 – 6523, Dec. 2015.

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

[7] E. Isufi, A. Loukas, A. Simonetto, and G. Leus, "Separable autoregressive moving average graph-temporal filters," in *European Signal Process. Conf. (EUSIPCO)*, Budapest, Hungary, Aug. 29 - Sept. 2 2016.

[8] ——, "Autoregressive moving average graph filtering," *IEEE Transactions on Signal Processing*, vol. 65, no. 2, pp. 274–288, 2017.

[9] A. Sandryhaila and J. Moura, "Discrete signal processing on graphs: Frequency analysis," *IEEE Trans. Signal Process.*, vol. 62, no. 12, pp. 3042–3054, June 2014.

[10] S. Segarra, A. G. Marques, G. Mateos, and A. Ribeiro, "Network topology inference from spectral templates," *arXiv preprint arXiv:1608.03008v1*, 2016.

[11] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*. Prentice-Hall, Inc., 1989.

[12] L. Xiao and S. Boyd, "Fast linear iterations for distributed averaging," *Systems & Control Lett.*, vol. 53, no. 1, pp. 65 – 78, 2004.

[13] S. Kar, J. Moura, and H. Poor, "Distributed linear parameter estimation: asymptotically efficient adaptive strategies," *SIAM Journal on Control and Optimization*, vol. 51, no. 3, pp. 2200–2229, May 2013.

[14] I. D. Schizas, A. Ribeiro, and G. B. Giannakis, "Consensus in ad-hoc WSNs with noisy links - Part I: Distributed estimation of deterministic signals," *IEEE Trans. Signal Process.*, vol. 56, no. 1, pp. 350–364, Jan. 2008.

[15] A. Tahbaz-Salehi and A. Jadbabaie, "A necessary and sufficient condition for consensus over random networks," *IEEE Trans. Auto. Control*, vol. 53, no. 3, pp. 791–795, Apr. 2008.

[16] G. Mateos, J.-A. Bazerque, and G. B. Giannakis, "Distributed sparse linear regression," *IEEE Trans. Signal Process.*, vol. 58, no. 10, pp. 5262–5276, Oct. 2010.

[17] A. Sayed, S. Barbarossa, S. Theodoridis, and I. Yamada, "Adaptation and learning over complex networks [from the guest editors]," *IEEE Signal Process. Mag.*, vol. 30, no. 3, pp. 14–15, May 2013.

[18] S. Barbarossa, G. Scutari, and T. Battisti, "Distributed signal subspace projection algorithms with maximum convergence rate for sensor networks with topological constraints," in *IEEE Intl. Conf. Acoust., Speech and Signal Process. (ICASSP)*, Apr. 2009, pp. 2893–2896.

[19] D. I. Shuman, P. Vandergheynst, and P. Frossard, "Distributed signal processing via Chebyshev polynomial approximation," *arXiv preprint arXiv:1111.5239*, 2011.

[20] X. Wang, P. Liu, and Y. Gu, "Local-set-based graph signal reconstruction," *IEEE Trans. Signal Process.*, vol. 63, no. 9, pp. 2432–2444, Sept. 2015.

[21] S. Chen, A. Sandryhaila, and J. Kovacevic, "Distributed algorithm for graph signal inpainting," in *IEEE Intl. Conf. Acoust., Speech and Signal Process. (ICASSP)*, Apr. 2015, pp. 3731 – 3735.

[22] A. Sandryhaila, S. Kar, and J. Moura, "Finite-time distributed consensus through graph filters," in *IEEE Intl. Conf. Acoust., Speech and Signal Process. (ICASSP)*, May 2014, pp. 1080–1084.

[23] S. Safavi and U. Khan, "Revisiting finite-time distributed algorithms via successive nulling of eigenvalues," *IEEE Signal Process. Lett.*, vol. 22, no. 1, pp. 54–57, Jan. 2015.

[24] S. Segarra, A. G. Marques, G. Leus, and A. Ribeiro, "Interpolation of graph signals using shift-invariant graph filters," in *European Signal Process. Conf. (EUSIPCO)*, Aug. 2015, pp. 210–214.

[25] ——, "Reconstruction of graph signals through percolation from seeding nodes," *IEEE Trans. Signal Process.*, vol. 64, no. 16, pp. 4363 – 4378, Aug. 2016.

[26] A. Y. Kibangou, "Finite-time average consensus based protocol for distributed estimation over AWGN channels," in *IEEE Conf. Decis. Contr. (CDC)*, Dec. 2011, pp. 5595–5600.

[27] S. Katti, S. Gollakota, and D. Katabi, "Embracing wireless interference: Analog network coding," *SIGCOMM Comput. Commun. Rev.*, vol. 37, no. 4, pp. 397–408, Oct 2007.

[28] S. Zhang, S. C. Liew, and P. P. Lam, "Physical-layer network coding," in *Intl. Conf. Mobile Comp. and Netw. (MobiCom)*, 2006, pp. 358–365.

[29] C. Godsil and G. Royle, *Algebraic Graph Theory*. Springer-Verlag, Graduate Texts in Mathematics, 2001, vol. 207.

[30] F. Pukelsheim, *Optimal Design of Experiments*. SIAM, 1993, vol. 50.

[31] G. Golub and C. Van Loan, *Matrix Computations (3rd Ed.)*. Johns Hopkins Univ. Press, 1996.

[32] M. L. Overton, "On minimizing the maximum eigenvalue of a symmetric matrix," *SIAM Journal on Matrix Analysis and Applications*, vol. 9, no. 2, pp. 256–268, 1988.

[33] S. Boyd and L. El Ghaoui, "Method of centers for minimizing generalized eigenvalues," *Linear Algebra and its Applications*, vol. 188, pp. 63–111, 1993.

[34] T. Ho and D. Lun, *Network Coding: An Introduction*. Cambridge University Press, 2008.

[35] R. Ahlswede, N. Cai, S.-Y. Li, and R. Yeung, "Network information flow," *IEEE Trans. Inf. Theory*, vol. 46, no. 4, pp. 1204–1216, Jul 2000.

[36] S.-Y. Li, R. Yeung, and N. Cai, "Linear network coding," *IEEE Trans. Inf. Theory*, vol. 49, no. 2, pp. 371–381, Feb. 2003.
[37] T. Nakano, M. Moore, F. Wei, A. Vasilakos, and J. Shuai, "Molecular communication and networking: Opportunities and challenges," *IEEE Trans. Nanobiosci.*, vol. 11, no. 2, pp. 135–148, June 2012.
[38] M. S. Kuran, H. B. Yilmaz, T. Tugcu, and B. Ozerman, "Energy model for communication via diffusion in nanonetworks," *Nano Commun. Networks*, vol. 1, no. 2, pp. 86 – 95, 2010.
[39] E. D. Kolaczyk, *Statistical Analysis of Network Data: Methods and Models*. Springer, 2009.

Santiago Segarra (M'16) received the B.Sc. degree in industrial engineering with highest honors (Valedictorian) from the Instituto Tecnológico de Buenos Aires (ITBA), Argentina, in 2011, the M.Sc. in electrical engineering from the University of Pennsylvania (Penn), Philadelphia, in 2014 and the Ph.D. degree in electrical and systems engineering from Penn in 2016. Since 2016, he has been working as a postdoctoral research associate with the Institute for Data, Systems, and Society at the Massachusetts Institute of Technology. His research interests include network theory, data analysis, machine learning, and graph signal processing. Dr. Segarra received the ITBA's 2011 Best Undergraduate Thesis Award in industrial engineering, the 2011 Outstanding Graduate Award granted by the National Academy of Engineering of Argentina, the Best Student Paper Awards at the 2015 Asilomar Conference and the 2016 IEEE Statistical Signal Processing Workshop, and the Best Paper Award at the 2016 IEEE Sensor Array and Multichannel Signal Processing Workshop.

Antonio G. Marques (SM'13) received the Telecommunications Engineering degree and the Doctorate degree, both with highest honors, from the Carlos III University of Madrid, Spain, in 2002 and 2007, respectively. In 2007, he became a faculty of the Dept. of Signal Theory and Communications, King Juan Carlos University, Madrid, Spain, where he currently develops his research and teaching activities as an Associate Professor. From 2005 to 2015, he held different visiting positions at the University of Minnesota, Minneapolis. In 2015 and 2016 he was a Visiting Scholar at the University of Pennsylvania.

His research interests lie in the areas of signal processing, networking and communications. His current research focuses on stochastic resource allocation for wireless and power networks, signal processing for graphs, and nonlinear network optimization. Dr. Marques has served the IEEE in a number of posts (currently, he is an Associate Editor of the IEEE Signal Process. Letters and a member of the Technical Committee on Signal Process. Theory and Methods), and his work has been awarded in several conferences and workshops.

Alejandro Ribeiro (M'07) received the B.Sc. degree in electrical engineering from the Universidad de la República Oriental del Uruguay, Montevideo, in 1998 and the M.Sc. and Ph.D. degree in electrical engineering from the Department of Electrical and Computer Engineering, the University of Minnesota, Minneapolis in 2005 and 2007. From 1998 to 2003, he was a member of the technical staff at Bellsouth Montevideo. After his M.Sc. and Ph.D studies, in 2008 he joined the University of Pennsylvania (Penn), Philadelphia, where he is currently the Rosenbluth Associate Professor at the Department of Electrical and Systems Engineering. His research interests are in the applications of statistical signal processing to the study of networks and networked phenomena. His focus is on structured representations of networked data structures, graph signal processing, network optimization, robot teams, and networked control. Dr. Ribeiro received the 2014 O. Hugo Schuck best paper award, and paper awards at the 2016 SSP Workshop, 2016 SAM Workshop, 2015 Asilomar SSC Conference, ACC 2013, ICASSP 2006, and ICASSP 2005. His teaching has been recognized with the 2017 Lindback award for distinguished teaching and the 2012 S. Reid Warren, Jr. Award presented by Penn's undergraduate student body for outstanding teaching. Dr. Ribeiro is a Fulbright scholar class of 2003 and a Penn Fellow class of 2015.