

## Advances in Distributed Graph Filtering

Coutiño, Mario; Isufi, Elvin; Leus, Geert

**DOI**

[10.1109/TSP.2019.2904925](https://doi.org/10.1109/TSP.2019.2904925)

**Publication date**

2019

**Document Version**

Accepted author manuscript

**Published in**

IEEE Transactions on Signal Processing

**Citation (APA)**

Coutiño, M., Isufi, E., & Leus, G. (2019). Advances in Distributed Graph Filtering. *IEEE Transactions on Signal Processing*, 67(9), 2320-2333. [8666778]. <https://doi.org/10.1109/TSP.2019.2904925>

**Important note**

To cite this publication, please use the final published version (if applicable).  
Please check the document version above.

**Copyright**

Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

**Takedown policy**

Please contact us and provide details if you believe this document breaches copyrights.  
We will remove access to the work immediately and investigate your claim.

# Advances in Distributed Graph Filtering

Mario Coutino, *Student Member, IEEE*, Elvin Isufi, *Student Member, IEEE*,  
Geert Leus, *Fellow, IEEE*

**Abstract**—Graph filters are one of the core tools in graph signal processing. A central aspect of them is their direct distributed implementation. However, the filtering performance is often traded with distributed communication and computational savings. To improve this tradeoff, this work generalizes state-of-the-art distributed graph filters to filters where every node weights the signal of its neighbors with different values while keeping the aggregation operation linear. This new implementation, labeled as edge-variant graph filter, yields a significant reduction in terms of communication rounds while preserving the approximation accuracy. In addition, we characterize the subset of shift-invariant graph filters that can be described with edge-variant recursions. By using a low-dimensional parametrization the proposed graph filters provide insights in approximating linear operators through the succession and composition of local operators, i.e., fixed support matrices, which span applications beyond the field of graph signal processing. A set of numerical results shows the benefits of the edge-variant filters over current methods and illustrates their potential to a wider range of applications than graph filtering.

**Index Terms**—consensus, distributed beamforming, distributed signal processing, edge-variant graph filters, FIR, ARMA, graph filters, graph signal processing.

## I. INTRODUCTION

**F**ILTERING is one of the core operations in signal processing. The necessity to process large amounts of data defined over non-traditional domains characterized by a graph triggers advanced signal processing of the complex data relations embedded in that graph. Examples of the latter include biological, social, and transportation network data. The field of graph signal processing (GSP) [2–4] has been established to incorporate the underlying structure in the processing techniques.

Through a formal definition of the graph Fourier transform (GFT), harmonic analysis tools employed for filtering in traditional signal processing have been adapted to deal with signals defined over graphs [5–10]. Similarly to time-domain filtering, graph filters manipulate the signal by selectively amplifying/attenuating its graph Fourier coefficients. Graph filters have seen use in applications including signal analysis [11, 12], classification [13, 14], reconstruction [7, 15, 16], denoising [8, 17–19] and clustering [20]. Furthermore, they are the central block in graph filterbanks [21, 22], wavelets [23], and convolutional neural networks [24, 25].

Distributed graph filter implementations emerged as a way to deal with the ubiquity of big data applications and to

improve the scalability of computation. By allowing nodes to exchange only local information, finite impulse response (FIR) [4, 6, 9] and infinite impulse response (IIR) [10, 26] architectures have been devised to implement a variety of responses.

However, being inspired by time domain filters, the above implementations do not fully exploit the structure in the data. The successive signal aggregations are locally weighted with similar weights often leading to high orders in approximating the desired response. To overcome this challenge, this paper proposes a generalization of the distributed graph filtering concept by applying edge-based weights to the information coming from different neighbors. While the detailed contributions are provided in Section I-B, let us here highlight that the above twist yields in graph filters that are flexible enough to capture complex responses with much lower complexity.

### A. Related Works

Driven by the practical need to implement a linear function with few local operations, the works in [9, 27] have put efforts to ease the communication and computational costs of graph filters (GF).

In [9], the authors modified the polynomial graph filters (i.e., the FIR structure) to graph filters with node-dependent weights. This architecture, referred to as a node-variant (NV) FIR graph filter, assigns different weights to different nodes and yields the same distributable implementation as the classical FIR graph filter [4, 6]. The NV FIR filter addresses a broader family of linear operators (e.g., analog network coding) that goes beyond the class of shift-invariant graph filters. However, the NV FIR filter uses the same weight for all signals arriving at a particular node, ignoring the affinity between neighbors. As we show next, this limits the ability of the NV FIR filter in approximating the desired linear operator with very low orders.

The work in [27] introduced stochastic sparsification to reduce the cost of a distributable implementation. Here, the authors considered random edge sampling in each aggregation step to implement the filter output with a lower complexity. Although conceptually similar to this work, the filter following [27] is stochastic and, therefore, the results hold only in expectation. Moreover, since this approach applies only to shift invariant filters, such as the FIR filter [4, 6] and the IIR [10] implementations, it cannot address linear operators that are not shift invariant.

Another related problem, which can be interpreted like graph filtering, is the multilayer sparse approximation of matrices [28]. Different from the previous two approaches, here a dense linear transform (matrix) is approximated through a sequence of sparse matrix multiplications to obtain a computational speedup. While this framework can be considered

The authors are with the faculty of Electrical Engineering, Mathematics and Computer Science, Delft University of Technology, 2826 CD Delft, The Netherlands. E-mails: {m.a.coutinominguez, e.isufi-1, g.j.t.leus}@tudelft.nl. This research is supported in part by the ASPIRE project (project 14926 within the STW OTP programme), financed by the Netherlands Organization for Scientific Research (NWO). Mario Coutino is partially supported by CONACYT. A conference version of this work was presented in [1].

as sequential diffusions over a network, the support of such sparse matrices differs in each iteration. This in practice can be a limitation since it often requires information from non-adjacent nodes within an iteration. Finally, in [29] the problem of optimal subspace projection by local interactions was studied. This paper proposed an algorithm to design the weights of a network to achieve the fastest convergence rate for this kind of linear operators. Although their method provides optimal weights for projecting the data to a particular subspace, it does not address the GSP setup of interest: implementation of graph filters or general linear operators.

### B. Paper Contributions

The main contribution of this work is the extension of the state-of-the-art graph filters to edge-variant (EV) graph filters. Due to the increased degrees of freedom (DoF), these filters allow for a complexity reduction of the distributed implementation while maintaining the approximation accuracy of current approaches. The salient points that broaden the existing literature are listed below.

- We present edge-variant architectures to implement FIR and IIR graph filtering. This framework extends the state-of-the-art graph filters by allowing nodes to weigh differently the information coming from different neighbors. In this way, only local exchanges are needed for each shift, thus yielding an efficient distributable implementation. Three forms are analyzed: First, the general class of linear edge-variant FIR filters is presented and its distributable implementation is discussed. Then, following the per-tone equalization idea [30], the constrained edge-variant FIR graph filter is introduced. This filter maintains a similar distributable implementation as the general form, yet allowing a simple least-squares design. Finally, the family of edge-variant autoregressive moving average graph filters of order one (ARMA<sub>1</sub>) is treated. This new IIR distributable architecture allows a better trade-off between approximation accuracy and convergence rate than current approaches.
- Through the definition of the filter modal response, we give a Fourier interpretation to a particular family of edge-variant graph filters. This subfamily shows a shift-invariant nature and links the filtering operation with the scaling applied on the graph modes (e.g., the graph shift eigenvectors).
- Besides outperforming state-of-the-art graph filters in GSP tasks such as approximating a user-provided frequency response, distributed consensus, and Tikhonov denoising, we present two new applications that could be addressed distributively with the proposed edge-variant graph filter. The latter includes a distributed solution of an inverse problem and distributed beamforming.

### C. Outline and Notation

This paper is organized as follows: Section II reviews the preliminaries of GSP, distributed graph filtering, and further defines the modal response of a graph filter. Section III generalizes the FIR graph filters to the edge-variant version. Here, we introduce the shift-invariant edge-variant graph filter

and characterize its graph modal response. Section IV analyzes a particular subfamily of edge-variant FIR graph filters, which enjoys a similar distributed implementation and a least-squares design strategy. In Section V, we generalize the idea of edge-variant filtering to the class of IIR graph filters. Section VI corroborates our findings with numerical results and Section VII concludes this paper.

In this paper, we adopt the following notation. Scalars, vectors, matrices, and sets are denoted by lowercase letters ( $x$ ), lowercase boldface letters ( $\mathbf{x}$ ), uppercase boldface letters ( $\mathbf{X}$ ), and calligraphic letters ( $\mathcal{X}$ ), respectively.  $[\mathbf{X}]_{ij}$  denotes the  $(i, j)$ th entry of the matrix  $\mathbf{X}$  whereas  $[x]_i$  represents the  $i$ th entry of the vector  $\mathbf{x}$ .  $\mathbf{X}^T$ ,  $\mathbf{X}^H$ , and  $\mathbf{X}^{-1}$  are respectively the transpose, the Hermitian, and inverse of  $\mathbf{X}$ . The Moore-Penrose pseudoinverse of  $\mathbf{X}$  is  $\mathbf{X}^\dagger$ . The Khatri-Rao product between  $\mathbf{X}$  and  $\mathbf{Y}$  is written as  $\mathbf{X} * \mathbf{Y}$ , while their Hadamard product as  $\mathbf{X} \odot \mathbf{Y}$ .  $\mathbf{1}$  and  $\mathbf{I}$  are the all-one vector and identity matrix of appropriate size, respectively.  $\text{vec}(\cdot)$  is the vectorization operation.  $\text{diag}(\cdot)$  refers to a diagonal matrix with its argument on the main diagonal.  $\text{null}\{\cdot\}$  and  $\text{span}\{\cdot\}$  denote the nullspace and span of their argument.  $\text{nnz}(\mathbf{X})$  and  $\text{supp}\{\mathbf{X}\}$  are the number of nonzero entries and the support of  $\mathbf{X}$ . Finally, we define the set  $[K] = \{1, 2, \dots, K\}$ .

## II. PRELIMINARIES

This section recalls the preliminary material that will be useful in the rest of the paper. It starts with the definition of the graph Fourier transform (GFT) and graph filtering. Then, two distributed recursions that implement FIR and IIR filtering operations on graphs are presented. Finally, the modal response of a graph filter is defined.

**Graph Fourier transform.** Consider an  $N$ -dimensional signal  $\mathbf{x}$  residing on the vertices of a graph  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$  with  $\mathcal{V} = \{v_1, \dots, v_N\}$  the set of  $N$  vertices and  $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$  the set of  $M$  edges. Let  $\mathbf{W}$  be the weighted graph adjacency matrix with  $\mathbf{W}_{i,j} \neq 0$  if  $(v_j, v_i) \in \mathcal{E}$  and  $\mathbf{W}_{i,j} = 0$ , otherwise. For an undirected graph, the graph Laplacian matrix is  $\mathbf{L}$ . Both  $\mathbf{W}$  and  $\mathbf{L}$  are valid candidates for the so-called graph shift operator  $\mathbf{S}$ , an  $N \times N$  matrix that carries the notion of delay in the graph setting [2–5]. Given the decomposition  $\mathbf{S} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^{-1}$  (assuming it exists), the GFT of  $\mathbf{x}$  is defined as the projection of  $\mathbf{x}$  onto the modes of  $\mathbf{S}$ , i.e.,  $\hat{\mathbf{x}} = \mathbf{U}^{-1}\mathbf{x}$ . Likewise, the inverse GFT is  $\mathbf{x} = \mathbf{U}\hat{\mathbf{x}}$ . Following the GSP convention, the eigenvalues  $\mathbf{\Lambda} = \text{diag}(\lambda_1, \dots, \lambda_N)$  are referred to as the graph frequencies.

**Graph filtering.** A linear shift-invariant graph filter is an operation on the graph signal with graph frequency domain output

$$\hat{\mathbf{y}} = h(\mathbf{\Lambda})\hat{\mathbf{x}}. \quad (1)$$

Here,  $h(\mathbf{\Lambda})$  is a diagonal matrix with the filter frequency response on its diagonal. More formally, the frequency response of a graph filter is a function

$$h : \mathbb{C} \mapsto \mathbb{R}, \quad \lambda_i \mapsto h(\lambda_i) \quad (2)$$

that assigns a particular value  $h(\lambda_i)$  to each graph frequency  $\lambda_i$ . This definition is akin to the one used in traditional signal processing, however depending on the underlying topology some shift operators might not be simple, i.e., the multiplicity

of some eigenvalues is greater than one. So, there is no one-to-one mapping between the graph frequencies  $\lambda_i$  and the graph modes  $\mathbf{u}_i$ . For this reason, at the end of this section, we will introduce the notion of graph modal response which treats the graph filters from the graph shift eigenvector perspective. Finally, by applying the inverse GFT on both sides of (1), we have the vertex domain filter output

$$\mathbf{y} = \mathbf{H}\mathbf{x}, \quad (3)$$

with  $\mathbf{H} = \mathbf{U}h(\Lambda)\mathbf{U}^{-1}$ .

**FIR graph filters.** A popular form of  $\mathbf{H}$  is its expression as a polynomial of the graph shift operator [4–6], i.e.,

$$\mathbf{H}_c \triangleq \sum_{k=0}^K \phi_k \mathbf{S}^k, \quad (4)$$

which we refer to as the *classical* FIR graph filter. It is possible to run the FIR filter (4) distributively due to the locality of  $\mathbf{S}$  [6, 9]. In particular, since  $\mathbf{S}^k \mathbf{x} = \mathbf{S}(\mathbf{S}^{k-1} \mathbf{x})$  the nodes can compute locally the  $k$ th shift of  $\mathbf{x}$  from the former ( $k - 1$ )th shift. Overall, an FIR filter of order  $K$  requires  $K$  local exchanges between neighbors and amounts to a computational and communication complexity of  $\mathcal{O}(MK)$ .

To expand the possible set of operations that can be implemented distributively through FIR recursions, [9] proposed the NV FIR graph filter. These filters have the node domain form

$$\mathbf{H}_{\text{nv}} \triangleq \sum_{k=0}^K \text{diag}(\phi_k) \mathbf{S}^k, \quad (5)$$

where the vector  $\phi_k = [\phi_{k,1}, \dots, \phi_{k,N}]^T$  contains the node dependent coefficients applied at the  $k$ th shift. For  $\phi_k = \phi_k \mathbf{1}$ , the NV FIR filter (5) reduces to the classical FIR filter (4). The NV FIR filter preserves also the efficient implementation of (4) since it applies the node coefficients to the  $k$ th shifted input  $\mathbf{S}^k \mathbf{x} = \mathbf{S}(\mathbf{S}^{k-1} \mathbf{x})$  with a computational complexity of  $\mathcal{O}(MK)$ .

If a linear operator  $\tilde{\mathbf{H}}$  needs to be approximated by a matrix polynomial as in (4), the filter order  $K$  can become large if a high accuracy is required. As the computational complexity scales with  $K$ , large-order graph filters incur high costs. The NV graph filters provide a first approach to tackle this issue. Starting from Section III, we generalize these ideas towards an *edge-variant* (EV) graph filter alternative, which due to its enhanced DoF can approximate  $\tilde{\mathbf{H}}$  with even a lower order  $K$ . Therefore, it leads to a more efficient implementation. One of the main benefits of both the NV and the EV graph filters is that they address a broader class of operators  $\tilde{\mathbf{H}}$  which not necessarily share the eigenvectors with  $\mathbf{S}$ , such as the analog network coding [9].

**IIR graph filters.** In [10], the authors introduced an ARMA recursion on graphs to implement distributively IIR graph filtering, i.e., a filtering operation characterized by a rational frequency response. The building block of this filter is the so-called ARMA graph filter of order one (ARMA<sub>1</sub>). This filter is obtained as the steady-state of the first-order recursion

$$\mathbf{y}_t = \psi \mathbf{S} \mathbf{y}_{t-1} + \varphi \mathbf{x}, \quad (6)$$

with arbitrary  $\mathbf{y}_0$  and scalar coefficients  $\psi$  and  $\varphi$ . The operation (6) is a distributed recursion on graphs, where neighbors now exchange their former output  $\mathbf{y}_{t-1}$  rather than the input  $\mathbf{x}$ . The per-iteration complexity of such a recursion is  $\mathcal{O}(M)$ .

Given  $\psi$  satisfies the convergence conditions for (6) [10], the steady-state output of the ARMA<sub>1</sub> is

$$\begin{aligned} \mathbf{y} &\triangleq \lim_{t \rightarrow \infty} \mathbf{y}_t = \varphi \sum_{\tau=0}^{\infty} (\psi \mathbf{S})^\tau \mathbf{x} = \varphi (\mathbf{I} - \psi \mathbf{S})^{-1} \mathbf{x} \quad (7) \\ &\triangleq \mathbf{H}_{\text{arma}_1} \mathbf{x}. \end{aligned}$$

Such a filter addresses several GSP tasks including Tikhonov denoising, graph signal interpolation under smoothness prior [10], and aggregate graph signal diffusion [31]. In Section V, we extend (6) to an edge-variant implementation with the aim to improve its convergence speed without heavily affecting the approximation accuracy.

**Graph modal response.** Before moving to the main contributions of this work, we define next the *modal response* of a graph filter. The latter represents the scaling that the graph modes experience when a graph signal undergoes a linear shift-invariant graph filtering operation.

**Definition 1.** (Graph modal response) *The modal response of a linear shift-invariant graph filter*

$$\mathbf{H} = \mathbf{U} \text{diag}(h_1, \dots, h_N) \mathbf{U}^{-1}, \quad (8)$$

is defined as the function

$$h : [N] \rightarrow \mathbb{C}, \quad i \mapsto h_i,$$

where  $h_i$  is the scaling experienced by the  $i$ th graph mode.

This definition is equivalent to the graph frequency response (2) when the shift operator has a simple spectrum. Since this is not always the case, we feel that the graph modal response is closer in meaning to the classical frequency response, and use it in the rest of the paper.

### III. EDGE-VARIANT FIR GRAPH FILTERS

Let us assume a scenario in which each node *trusts* differently the information coming from different neighbors, e.g., a person is likely to weigh more the opinion of his/her partner than that of a colleague on a personal recommendation. So, it is reasonable to treat this case as a graph filter, where each node weighs differently the information of its neighbors.

Here, we formalize the above intuition in terms of EV FIR graph filters. We first introduce the general form of these filters while in Section III-B we focus on the class of shift-invariant edge-variant (SIEV) FIR graph filter. The filter design strategy is discussed in Section III-C.

#### A. General Form

Consider an extension of the above edge-dependent fusion to several diffusion steps (signal shifts) where in each shift a different set of weights is used. At the  $k$ th diffusion, node  $v_i$  weighs its neighbouring node  $v_l$  with the weight  $\phi_{i,l}^{(k)}$ . Hence, in each shift  $k \in [K]$ , and for each node  $v_i$ , there is a set of coefficients  $\{\phi_{i,l}^{(k)}\}$  for  $l \in \mathcal{N}_{v_i}$ . Here,  $\mathcal{N}_{v_i}$  denotes the set of nodes adjacent to  $v_i$ , and  $K$  is the number of shifts. Mathematically, the above behavior can be written through an order- $K$  general EV FIR graph filter defined as

$$\begin{aligned} \mathbf{H}_{\text{ev}} &\triangleq \Phi_1 + \Phi_2 \Phi_1 + \dots + \Phi_K \Phi_{K-1} \cdots \Phi_1 \\ &= \sum_{k=1}^K \Phi_{k:1}, \end{aligned} \quad (9)$$

where  $\Phi_{k:1} = \Phi_k \Phi_{k-1} \cdots \Phi_1$  and  $\Phi_j \in \mathbb{C}^{N \times N}$  is an edge-weighting matrix constructed from the coefficient set  $\{\{\phi_{1,l}^{(j)}\}, \dots, \{\phi_{N,l}^{(j)}\}\}$ , more specifically  $[\Phi_j]_{il} = \phi_{i,l}^{(j)}$ . By construction, the support of  $\Phi_j$  and  $\mathbf{S} + \mathbf{I}$  is the same  $\forall j \in [K]$ . Since  $\mathbf{S}$  might have zero entries on its diagonal, i.e.,  $\mathbf{S} = \mathbf{W}$ , here we extend the support of  $\{\Phi_j\}_{j \in [K]}$  to allow each node to use also its own information. Note that definition (9) does not impose any symmetry on the coefficient matrices  $\Phi_j$ . Depending on how adjacent nodes trust each other, the applied weights can be different.

The filter can differently be interpreted through time-varying shift operators [32, 33], where  $\Phi_j$  is the weighted, possibly directed shift operator for the  $j$ th diffusion step with the support of  $\mathbf{S} + \mathbf{I}$ . Therefore, the general EV FIR filter accounts for signals that are generated through time-varying systems in directed subgraphs of the original graph. Here, the filter coefficient matrix only allows for *edge deletion* or a re-weighting of graph flows.

Note that recursion (9) is a distributed graph filter. To compute the output  $\mathbf{y} = \mathbf{H}_{\text{ev}} \mathbf{x}$ , each node is only required to track the following quantities:

- the shifted signal output  $\mathbf{x}^{(k)} = \Phi_k \mathbf{x}^{(k-1)}$ ,  $\mathbf{x}^{(0)} = \mathbf{x}$ ,
- the accumulator output  $\mathbf{y}^{(k)} = \mathbf{y}^{(k-1)} + \mathbf{x}^{(k)}$ ,  $\mathbf{y}^{(0)} = \mathbf{0}$ .

Both these operations can be computed locally in each node by combining only neighboring data. Hence, (9) preserves the efficient distributed implementation of the classical FIR graph filter (4) with a complexity of  $\mathcal{O}(MK)$ .

Before addressing the design strategy of the filter (9), in the sequel, we introduce a particular structure of EV FIR graph filters that enjoy a graph Fourier domain interpretation.

### B. Shift-Invariant Edge-Variant Graph Eigenfilters

An important family of graph filters is that of *shift-invariant graph filters*, i.e., filters that commute with the graph shift operator  $\mathbf{S}$ . That is, given the shift  $\mathbf{S}$  and the filter matrix  $\mathbf{H}$ , the following holds

$$\mathbf{S}\mathbf{H} = \mathbf{H}\mathbf{S}. \quad (10)$$

For a non-defective shift operator  $\mathbf{S}$  and filter  $\mathbf{H}$ , i.e., the matrices accept an eigenvalue decomposition, condition (10) is equivalent to saying that the matrices  $\mathbf{S}$  and  $\mathbf{H}$  are jointly diagonalizable, or that their eigenbases coincide.

There is no reason to believe that the graph filters of form (9) are shift invariant. However, it is possible to characterize a subset of edge-variant graph filters that satisfy this property. To do that, we rely on the following assumptions:

(A.0)  $\mathbf{S}$  is diagonalizable;

(A.1) Each  $\Phi_j$ ,  $j \in [K]$  is diagonalizable with the eigenbasis of  $\mathbf{S}$ ;

(A.2) Each  $\Phi_j$ ,  $j \in [K]$  shares the support with  $\mathbf{S} + \mathbf{I}$ .

Given the above assumptions hold, we can rewrite (9) as

$$\mathbf{H}_{\text{ev}} = \sum_{k=1}^K \Phi_{k:1} = \mathbf{U} \left[ \sum_{k=1}^K \prod_{j=1}^k \Lambda_j \right] \mathbf{U}^{-1}, \quad (11)$$

where we substituted  $\Phi_j = \mathbf{U} \Lambda_j \mathbf{U}^{-1}$ . To provide a closed-form expression for the effect of such graph filters on the graph modes, let us first describe the set of *fixed-support matrices that are diagonalizable with a particular eigenbasis*

(i.e., matrices that meet (A.1) and (A.2)). Mathematically, this set is defined as

$$\mathcal{J}_{\mathbf{U}}^{\mathbf{A}} = \{\mathbf{A} : \mathbf{A} = \mathbf{U} \Omega \mathbf{U}^{-1}, [\text{vec}(\mathbf{A})]_i = 0, \forall i \in \mathcal{A}\}, \quad (12)$$

where  $\mathcal{A}$  is the index set defining the zero entries of  $\mathbf{S} + \mathbf{I}$  and  $\Omega$  is diagonal. The fixed-support condition in  $\mathcal{J}_{\mathbf{U}}^{\mathbf{A}}$  can be expressed in the linear system form

$$\Phi_{\mathcal{A}} \text{vec}(\mathbf{A}) = \mathbf{0}, \quad (13)$$

with  $\Phi_{\mathcal{A}} \in \{0, 1\}^{|\mathcal{A}| \times N^2}$  denoting the selection matrix whose rows are the rows of an  $N^2 \times N^2$  identity matrix indexed by the set  $\mathcal{A}$ . By leveraging the vectorization operation properties and the knowledge of the eigenbasis of  $\mathbf{A}$ , we can rewrite (13) as

$$\Phi_{\mathcal{A}} \text{vec}(\mathbf{A}) = \Phi_{\mathcal{A}} (\mathbf{U}^{-\text{T}} * \mathbf{U}) \boldsymbol{\omega} = \mathbf{0}, \quad (14)$$

where “\*” represents the Kathri-Rao product and  $\boldsymbol{\omega} = [[\Omega]_{11}, [\Omega]_{22}, \dots, [\Omega]_{NN}]^{\text{T}}$  is the vector containing the eigenvalues of  $\mathbf{A}$ . From (14), we see that  $\boldsymbol{\omega}$  characterizes the intersection of the nullspace of  $\Phi_{\mathcal{A}}$  and the range of  $\mathbf{U}^{-\text{T}} * \mathbf{U}$ . More formally, we write

$$\boldsymbol{\omega} \in \text{null}\{\mathbf{T}_{\mathbf{U}}^{\mathbf{A}}\}, \quad (15)$$

with  $\mathbf{T}_{\mathbf{U}}^{\mathbf{A}} = \Phi_{\mathcal{A}} (\mathbf{U}^{-\text{T}} * \mathbf{U})$ .

With this in place, the following proposition characterizes the matrices that belong to the set  $\mathcal{J}_{\mathbf{U}}^{\mathbf{A}}$ .

**Proposition 1.** (Graph shift nullspace property) *Given an orthonormal basis  $\mathbf{U}$  and a sparsity pattern defined by the set  $\mathcal{A}$ , the matrices within the set  $\mathcal{J}_{\mathbf{U}}^{\mathbf{A}}$  are of the form  $\mathbf{A} = \mathbf{U} \Omega \mathbf{U}^{-1}$  and have eigenvalues given by*

$$\Omega = \text{diag}(\mathbf{B}_{\mathbf{U}}^{\mathbf{A}} \boldsymbol{\alpha}), \quad (16)$$

where the matrix  $\mathbf{B}_{\mathbf{U}}^{\mathbf{A}}$  is a basis for the nullspace of  $\mathbf{T}_{\mathbf{U}}^{\mathbf{A}}$ , i.e.,

$$\text{span}\{\mathbf{B}_{\mathbf{U}}^{\mathbf{A}}\} = \text{null}\{\mathbf{T}_{\mathbf{U}}^{\mathbf{A}}\},$$

and  $\boldsymbol{\alpha}$  is the basis expansion coefficient vector.

*Proof.* The proof follows from (14)-(15).  $\square$

The above result is not entirely surprising and has been used for assessing the uniqueness of the graph shift operator in topology identification [34]. Here, we leverage Proposition 1 for interpreting the response of the SIEV graph filters. More specifically, under (A.1) and (A.2) we can express each matrix  $\Phi_j$  of (9) as

$$\Phi_j = \mathbf{U} \text{diag}(\mathbf{B}_{\mathbf{U}}^{\mathbf{A}} \boldsymbol{\alpha}_j) \mathbf{U}^{-1}, \quad (17)$$

and write any SIEV FIR filter as

$$\mathbf{H}_{\text{siev}} = \mathbf{U} \left[ \sum_{k=1}^K \prod_{j=1}^k \text{diag}(\mathbf{B}_{\mathbf{U}}^{\mathbf{A}} \boldsymbol{\alpha}_j) \right] \mathbf{U}^{-1}. \quad (18)$$

The following proposition formally characterizes the frequency interpretation of such filters in terms of the modal response.

**Proposition 2.** (Modal Response of SIEV FIR) *An FIR graph filter of the form (9) satisfying (A.1) and (A.2) has  $i$ th modal response*

$$h_i = \sum_{k=1}^K \prod_{j=1}^k (\mathbf{b}_{\mathbf{U},i}^{\mathbf{A}})^{\text{T}} \boldsymbol{\alpha}_j + (\mathbf{b}_{\mathbf{U},i}^{\mathbf{A}})^{\text{T}} \boldsymbol{\alpha}_0, \quad (19)$$

where  $(\mathbf{b}_{\mathbf{U},i}^{\mathbf{A}})^{\text{T}}$  is the  $i$ th row of  $\mathbf{B}_{\mathbf{U}}^{\mathbf{A}}$ .

*Proof.* The proof follows directly from (18).  $\square$

An interesting outcome from Proposition 2 is that the filter response is independent of the graph frequencies. This is clear from (19), where we see that the eigenvalue  $\lambda_i$  does not appear in the expression of  $h_i$ . Therefore, we can interpret the SIEV FIR graph filters as *eigenvector filters*, since they act on the eigenmodes of the graph. That is, for each graph eigenmode (eigenvector)  $\mathbf{u}_i$ ,  $\mathbf{H}_{\text{siev}}$  might apply a different gain given by (19) (independent of  $\lambda_i$ ) to the component of the input signal  $\mathbf{x}$  in the direction of  $\mathbf{u}_i$ . This is in contrast to classical FIR graph filters which apply the same polynomial expression to all modes  $\{\mathbf{u}_i\}_{i \in [N]}$ .

The following section introduces methods for designing EV FIR graph filters in the node domain and SIEV FIR graph filters using the parametrization in (19).

### C. Filter Design

**General form.** Given a desired operator  $\tilde{\mathbf{H}}$ , we design an EV FIR filter  $\mathbf{H}_{\text{ev}}$  [cf. (9)] that approximates  $\tilde{\mathbf{H}}$  as the solution of the optimization problem

$$\begin{aligned} \underset{\{\Phi_k\}}{\text{minimize}} \quad & \|\tilde{\mathbf{H}} - \sum_{k=1}^K \Phi_{k:1}\| \\ \text{subject to} \quad & \Phi_{k:1} = \Phi_k \Phi_{k-1} \cdots \Phi_1, \\ & \text{supp}\{\Phi_k\} = \text{supp}\{\mathbf{S} + \mathbf{I}\} \forall k \in [K], \end{aligned} \quad (20)$$

where  $\|\cdot\|$  is an appropriate distance measure, e.g., the Frobenius norm ( $\|\cdot\|_F$ ), or the spectral norm ( $\|\cdot\|_2$ ).

Unfortunately, (20) is a high-dimensional nonconvex problem and hard to optimize. An approach to finding a local solution for it is through block coordinate methods, which provide local convergence guarantees when applied to such problems [35]. In fact, the cost in (20) is a *block multi-convex* function, i.e., the cost function is a convex function of  $\Phi_i$  with all the other variables fixed.

Starting then with an initial set of matrices  $\{\Phi_j^{(0)}\}_{j \in [K]}$  (potentially initialized with an order- $K$  classical FIR filter), we solve a sequence of optimization problems where at the  $i$ th step, the matrix  $\Phi_i$  is found. That is, at the  $i$ th iteration, we fix the matrices  $\{\Phi_j^{(0)}\}_{j \in [K] \setminus \{i\}}$  and solve the convex problem

$$\begin{aligned} \underset{\Phi_i}{\text{minimize}} \quad & \|\tilde{\mathbf{H}} - \sum_{k=1}^K \Phi_{k:(i+1)}^{(0)} \Phi_i \Phi_{(i-1):1}^{(0)}\| \\ \text{subject to} \quad & \text{supp}\{\Phi_i\} = \text{supp}\{\mathbf{S} + \mathbf{I}\}, \end{aligned} \quad (21)$$

where  $\Phi_{a:b}^{(0)} = \Phi_a^{(0)} \Phi_{a-1}^{(0)} \cdots \Phi_{b+1}^{(0)} \Phi_b^{(0)}$  for  $a \geq b$  and  $\Phi_{a:b}^{(0)} = \mathbf{I}$ , otherwise. Then, the matrix  $\Phi_i^{(0)}$  is updated with its solution and the procedure is repeated for all  $\{\Phi_j\}_{j \in [K]}$ . If the final fitting error is large, the whole process can be repeated until the desired performance is reached, or until a local minimum is found.

Although filter (9) is the most general EV FIR filter form, the non-convexity encountered in the above design strategy may often lead to a local solution with an unacceptable performance. To tackle such issue, in Section IV, we introduce a constrained EV FIR filter which provides a higher flexibility than the state-of-the-art graph filters while accepting a simple least squares design.

**SIEV form.** Besides enjoying the modal response interpretation, the SIEV FIR filter also has a simpler design than the

general form (9). For  $\{\tilde{h}_i\}_{i=1}^N$  being the desired graph modal response<sup>1</sup>, the SIEV FIR filter design consists of solving the optimization problem

$$\underset{\{\alpha_j\}}{\text{minimize}} \quad \sum_{i=1}^N \|\tilde{h}_i - \sum_{k=1}^K \prod_{j=1}^k (\mathbf{b}_{\mathcal{V},i}^A)^T \alpha_j\|_2^2. \quad (22)$$

Similarly to (20), problem (22) is nonconvex and cannot in general be solved up to global optimality with standard convex optimization methods. However, (22) is also a block multi-convex function in each  $\alpha_i$ ,  $i \in [K]$  individually and, therefore, the block coordinate descent methods [35] can be employed to find a local minimum. Alternatively, the straightforward analytical expression of the gradient of the cost function allows the use of off-the-shelf solvers for global optimization, such as the MATLAB's built-in `fmincon` function [36].

## IV. CONSTRAINED EDGE-VARIANT FIR GRAPH FILTERS

To overcome the design issues of the general EV FIR filter, here we present a constrained version of it that retains both the distributed implementation and the edge-dependent weighting. This reduction of the DoF will, in fact, allow us to design the filter coefficients in a least squares fashion. The structure of these filters along with their distributed implementation is presented in the next section. In Section IV-B we provide a modal response interpretation of these filters, while in Section IV-C we present the design strategy.

### A. General Form

The *constrained* EV (CEV) FIR graph filter is defined as

$$\mathbf{H}_{\text{cev}} = \Phi_1 + \Phi_2 \mathbf{S} + \cdots + \Phi_K \mathbf{S}^{K-1} \triangleq \sum_{k=1}^K \Phi_k \mathbf{S}^{k-1}, \quad (23)$$

where the edge-weighting matrices  $\{\Phi_k\}_{k \in [K]}$  again share the support with  $\mathbf{S} + \mathbf{I}$ . These filters enjoy the same distributed implementation of the general form (9). In fact, each node can compute locally the filter output by tracking the following quantities:

- the regular shift output  $\mathbf{x}^{(k)} = \mathbf{S} \mathbf{x}^{(k-1)}$ ,  $\mathbf{x}^{(0)} = \mathbf{x}$ ,
- the weighted shift output  $\mathbf{z}^{(k)} = \Phi_k \mathbf{x}^{(k-1)}$ ,
- the accumulator output  $\mathbf{y}^{(k)} = \mathbf{y}^{(k-1)} + \mathbf{z}^{(k)}$ ,  $\mathbf{y}^{(0)} = \mathbf{0}$ .

From the locality of  $\mathbf{S}$  and  $\Phi_k$ , both  $\mathbf{x}^{(k)}$  and  $\mathbf{z}^{(k)}$  require only neighboring information. The final filter output is  $\mathbf{y} = \mathbf{y}^{(K)}$  which yields the same computational complexity of  $\mathcal{O}(MK)$ .

Note that construction (23) still applies different weights to the signal coming from different edges. However, instead of adopting a different *diffusion* matrix at every step, the signal diffusion occurs through the graph shift  $\mathbf{S}$ . The additional extra step mixes locally  $\mathbf{x}^{(k-1)}$  using edge-dependent weights, which are allowed to vary for each  $k$ . We here adopt the term constrained for this implementation from the observation that the diffusion is performed using only a single shift operator matrix. Fig. 1(a) visually illustrates the differences between the different graph filters analyzed so far.

**Remark 1.** The NV graph filter from [9] [cf. (5)] is a particular case of the CEV graph filter. The local matrices

<sup>1</sup>This can be for instance a low-pass form if we want to keep only the eigenvector contribution associated with the low graph frequencies.

$\{\Phi_k\}_{k=1}^K$  are in fact substituted by diagonal matrices with distinct elements across their diagonals.

### B. Shift-Invariant Constrained Edge-Variant Graph Eigenfilters

Following the same lines of Section III-B, we can use the set  $\mathcal{J}_U^A$  (12) to characterize the graph modal response of the CEV FIR graph filter when the matrices  $\{\Phi_k\}_{k=1}^K$  satisfy (A.1) and (A.2). This subset of CEV FIR graph filters, which we refer to as shift-invariant CEV (SICEV) FIR graph filters, can again be expressed in terms of  $B_U^A$  and  $\{\alpha_k\}_{k=0}^K$  as

$$\mathbf{H}_{\text{sicev}} = \mathbf{U} \left[ \sum_{k=1}^K \text{diag}(B_U^A \alpha_k \odot \lambda^{\odot(k-1)}) \right] \mathbf{U}^{-1}, \quad (24)$$

where  $\lambda^{\odot k}$  denotes the  $k$ th element-wise power of the eigenvalue vector of the shift operator  $\mathbf{S}$ . The subsequent proposition formalizes the modal response of these filters.

**Proposition 3.** (Modal Response of SICEV FIR) *An FIR graph filter of the form (23) satisfying (A.1) and (A.2) has  $i$ th modal response*

$$h_i = \sum_{k=1}^K \gamma_{ik} \lambda_i^{(k-1)}, \quad (25)$$

where  $\gamma_{ik} = (\mathbf{b}_{U,i}^A)^T \alpha_k$  is the  $k$ th polynomial coefficient for the  $i$ th graph frequency and  $(\mathbf{b}_{U,i}^A)^T$  is the  $i$ th row of  $B_U^A$ .

*Proof.* The proof follows directly from (24).  $\square$

From (25), we see that there is a substantial difference between the SICEV FIR filter and the more general SIEV FIR graph filters. Here, the modal response is a polynomial in the graph frequencies. This is similar as for the classical FIR filter (4), but now each frequency has a different set of coefficients. In other words, the modal response of the SICEV FIR filter is a *mode-dependent polynomial*. For readers more familiar with traditional discrete-time processing, this behavior can be interpreted as applying different polynomial filters to each frequency bin (see e.g., [30]).

**Remark 2.** *The particular form of the SICEV FIR filter allows it to match all shift-invariant polynomial responses of order  $K$  and a subset of higher-order polynomials of order up to  $N - 1$ . The latter property derives from the observation that any shift-invariant graph filter is a polynomial of the graph shift operator [4] and from the filter response in (25). In fact, the SICEV FIR filter is still a polynomial of the shift  $\mathbf{S}$ , though with a different polynomial response per graph frequency. This additional freedom extends the set of functions that can be approximated by a SICEV FIR filter of order  $K$ . Fig. 1(b) further illustrates the relation among different graph filters.*

### C. Filter Design

**General form.** Following a similar approach as in Section III-C, we can approximate a desired operator  $\tilde{\mathbf{H}}$  with a CEV FIR filter by solving the problem

$$\begin{aligned} & \underset{\{\Phi_k\}}{\text{minimize}} \quad \|\tilde{\mathbf{H}} - \sum_{k=1}^K \Phi_k \mathbf{S}^{k-1}\|_F^2 \\ & \text{subject to} \quad \text{supp}\{\Phi_k\} = \text{supp}\{\mathbf{S} + \mathbf{I}\} \forall k \in [K]. \end{aligned} \quad (26)$$

Exploiting then the properties of the vectorization operator and the Frobenius norm, we can transform (26) into

$$\begin{aligned} & \underset{\{\phi_k\}}{\text{minimize}} \quad \|\tilde{\mathbf{h}} - \sum_{k=1}^K (\mathbf{S}^{k-1} \otimes \mathbf{I}) \phi_k\|_2 \\ & \text{subject to} \quad \tilde{\mathbf{h}} \triangleq \text{vec}(\tilde{\mathbf{H}}), \quad \phi_k \triangleq \text{vec}(\Phi_k), \\ & \quad \text{supp}\{\Phi_k\} = \text{supp}\{\mathbf{S} + \mathbf{I}\} \forall k \in [K]. \end{aligned} \quad (27)$$

Since the support of the weighting matrices is known, problem (27) can be written in the reduced-size form

$$\begin{aligned} & \underset{\{\phi_k\}}{\text{minimize}} \quad \|\tilde{\mathbf{h}} - \Psi \boldsymbol{\theta}\|_2^2 \\ & \text{subject to} \quad \Psi = [\mathbf{I} \check{\mathbf{S}} \cdots \check{\mathbf{S}}_K] \\ & \quad \boldsymbol{\theta} = [\check{\phi}_0^T \check{\phi}_1^T \cdots \check{\phi}_K^T]^T, \end{aligned} \quad (28)$$

where,  $\check{\phi}_k$  is the vector  $\phi_k$  with the zero entries removed,  $\check{\mathbf{S}}_k$  is the matrix  $(\mathbf{S}^k \otimes \mathbf{I})$  with the appropriate columns removed. In addition, if a regularized solution is desired, a natural penalization term might be the convex  $\ell_1$ -norm which induces sparsity in the solution yielding only few active coefficients.

Problem (27) has a unique solution as long as  $\Psi$  is full column rank, i.e.,  $\text{rank}(\Psi) = \text{nnz}(\mathbf{S}) \cdot K + N$ . Otherwise, regularization must be used to obtain a unique solution.

**Remark 3.** *Besides leading to a simple least squares problem, the design of the CEV FIR filter can also be computed distributively. Given that each node knows the desired filter response and the graph shift operator (i.e., the network structure), it can be shown that by reordering the columns of  $\Psi$  and the entries of  $\boldsymbol{\theta}$  the framework of splitting-over-features [37] can be employed for a decentralized estimation of  $\boldsymbol{\theta}$ .*

**SICEV form.** Similar to the more general CEV FIR filter, the design of  $\{\alpha_k\}_{k=1}^K$  for the SICEV form can be performed in a least squares fashion.

First, for a set of vectors  $\{\alpha_k\}_{k=1}^K$  the modal response for the SICEV FIR filter reads as

$$\mathbf{h}_\lambda = \sum_{k=1}^K [B_U^A \alpha_k \odot \lambda^{\odot(k-1)}], \quad (29)$$

where  $\mathbf{h}_\lambda$  is obtained by stacking the modal responses, i.e.,  $\{h_i\}_{i=1}^N$ , in a column vector. By using the properties of the Hadamard product, we can rewrite (29) as

$$\mathbf{h}_\lambda = \sum_{k=1}^K \text{diag}(\lambda^{\odot(k-1)}) B_U^A \alpha_k = \sum_{k=1}^K M_k \alpha_k, \quad (30)$$

with  $M_k = \text{diag}(\lambda^{\odot(k-1)}) B_U^A$ . Defining then  $\mathbf{M} = [M_1, \dots, M_K]$ , and  $\boldsymbol{\alpha} = [\alpha_1^T, \dots, \alpha_K^T]^T$ , we obtain the linear relation

$$\mathbf{h}_\lambda = \mathbf{M} \boldsymbol{\alpha}. \quad (31)$$

Therefore, the approximation of a desired response  $\tilde{\mathbf{h}}_\lambda = [\tilde{h}_1, \dots, \tilde{h}_N]^T$  consists of solving the least squares problem

$$\underset{\boldsymbol{\alpha} \in \mathbb{R}^{d(K+1)}}{\text{minimize}} \quad \|\tilde{\mathbf{h}}_\lambda - \mathbf{M} \boldsymbol{\alpha}\|_2 \quad (32)$$

which has a unique solution when  $\mathbf{M}$  is full column rank, i.e.,  $\text{rank}(\mathbf{M}) = d(K+1) \leq N$ .

## V. EDGE-VARIANT IIR GRAPH FILTERS

We now extend the edge-variant filtering concept to the class of IIR graph filters. As stated in Section II, we focus on the

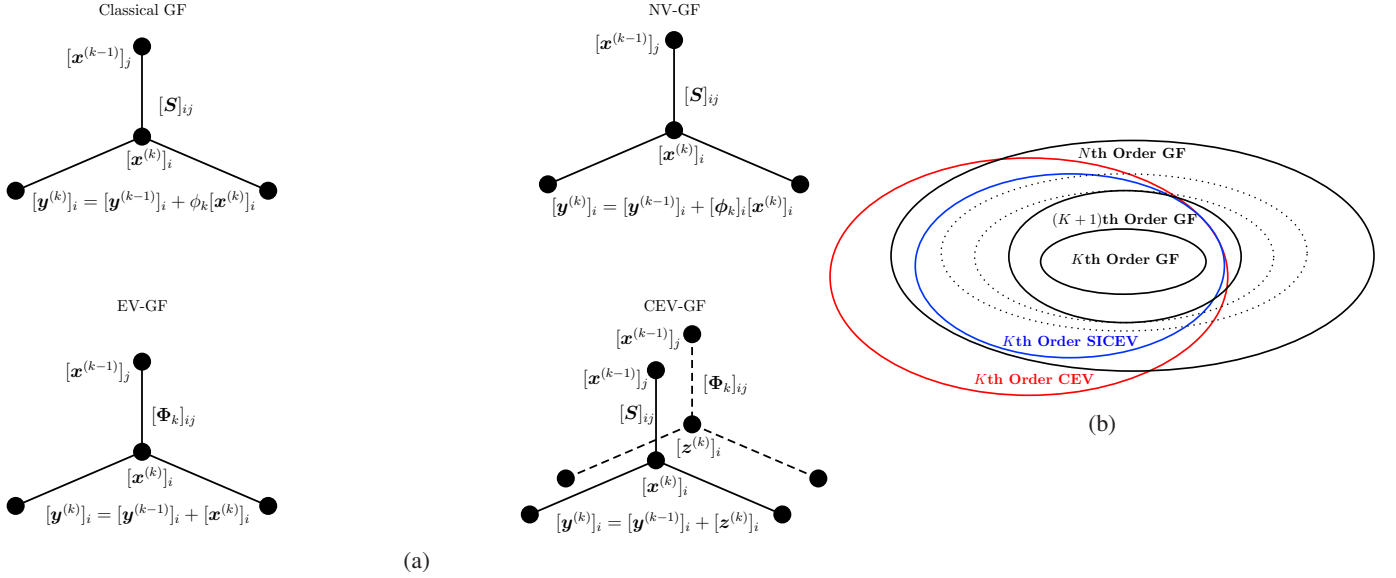


Fig. 1: (a) Illustration of the required transmission, scaling, and recursion performed by the different graph filters. (b) Relation between the classical and CEV FIR graph filters. This figure depicts the possibility of obtaining higher-order polynomial graph filters with reduced order CEV graph filters.

basic building block of these filters, i.e., the  $\text{ARMA}_1$  recursion (6). We follow the same organization of the former two sections, by introducing the edge-variant  $\text{ARMA}_1$  structure in Section V-A, the shift-invariant version in Section V-B, and the design strategies in Section V-C.

#### A. Edge-Variant $\text{ARMA}_1$

We build an edge-variant  $\text{ARMA}_1$  recursion on graphs by modifying (6) as

$$\mathbf{y}_t = \Phi_1 \mathbf{y}_{t-1} + \Phi_0 \mathbf{x}, \quad (33)$$

where  $\Phi_0$  and  $\Phi_1$  are the edge-weighting matrices having the support of  $\mathbf{S} + \mathbf{I}$  that respectively weight locally the entries of  $\mathbf{y}_{t-1}$  and  $\mathbf{x}$ . Proceeding similarly as in [10], for  $\|\Phi_1\|_2 < 1$ , the steady-state output of (33) is

$$\mathbf{y} = \lim_{t \rightarrow \infty} \mathbf{y}_t = (\mathbf{I} - \Phi_1)^{-1} \Phi_0 \mathbf{x} \triangleq \mathbf{H}_{\text{eva}_1} \mathbf{x}, \quad (34)$$

where now we notice the inverse relation w.r.t. the edge-weighting matrix  $\Phi_1$ . Recursion (33) converges to (34) linearly with a rate governed by  $\|\Phi_1\|_2$ . The classical form (6) can be obtained by substituting  $\Phi_1 = \psi \mathbf{S}$  and  $\Phi_0 = \varphi \mathbf{I}$ .

The edge-variant  $\text{ARMA}_1$  filter presents the same frequency interpretation challenges as the FIR filter counterpart. Therefore, we next analyze the shift-invariant version of it and we will see a rational modal response.

#### B. Shift-Invariant EV $\text{ARMA}_1$

By limiting the choices of  $\{\Phi_0, \Phi_1\}$  to the one that satisfy (A.1) and (A.2), we obtain the shift-invariant edge-variant  $\text{ARMA}_1$  (SIEVA<sub>1</sub>) graph filter

$$\mathbf{H}_{\text{sieva}_1} = \mathbf{U}[(\mathbf{I} - \text{diag}(\mathbf{B}_{\mathbf{U}}^A \alpha_1))^{-1} \text{diag}(\mathbf{B}_{\mathbf{U}}^A \alpha_0)] \mathbf{U}^{-1}, \quad (35)$$

where  $\alpha_0$  and  $\alpha_1$  are the respective basis expansion vectors of  $\Phi_0$  and  $\Phi_1$  onto the nullspace of  $\mathbf{T}_{\mathbf{U}}^A$  (see Proposition 1). From (35), we see that the inverse relation that appears in (34) indeed appears as a function affecting the graph eigenmodes.

The following proposition concludes this section by stating this finding in a formal way.

**Proposition 4.** (Modal Response of SIEVA<sub>1</sub>) *An  $\text{ARMA}_1$  graph filter of the form (34) satisfying (A.1) and (A.2) for  $K = 1$  has  $i$ th modal response*

$$h_i = \frac{(\mathbf{b}_{\mathbf{U},i}^A)^\top \alpha_0}{1 - (\mathbf{b}_{\mathbf{U},i}^A)^\top \alpha_1} \quad (36)$$

where  $(\mathbf{b}_{\mathbf{U},i}^A)^\top$  is the  $i$ th row of the matrix  $\mathbf{B}_{\mathbf{U}}^A$ .

*Proof.* The proof follows directly from (35).  $\square$

#### C. Filter Design

**Edge-Variant  $\text{ARMA}_1$  form.** Here, we extend the design approach of [38] and design  $\{\Phi_0, \Phi_1\}$  by using the Prony's method. For  $\tilde{\mathbf{H}}$  being the desired operator, we can define the fitting error matrix

$$\mathbf{E} = \tilde{\mathbf{H}} - (\mathbf{I} - \Phi_1)^{-1} \Phi_0, \quad (37)$$

which similar to the classical Prony design presents nonlinearities in the denominator coefficients, i.e., in  $\Phi_1$ . To tackle these issues, we consider the modified fitting error matrix

$$\mathbf{E}' = \tilde{\mathbf{H}} - \Phi_1 \tilde{\mathbf{H}} - \Phi_0, \quad (38)$$

which is obtained by multiplying both sides of (37) by  $\mathbf{I} - \Phi_1$ .

This way, the filter design is transformed in solving the convex optimization problem

$$\begin{aligned} & \underset{\Phi_0, \Phi_1}{\text{minimize}} && \|\tilde{\mathbf{H}} - \Phi_1 \tilde{\mathbf{H}} - \Phi_0\| \\ & \text{subject to} && \|\Phi_1\|_2 < \delta, \quad \delta < 1, \end{aligned} \quad (39)$$

$$\text{supp}\{\Phi_0\} = \text{supp}\{\Phi_1\} = \text{supp}\{\mathbf{S} + \mathbf{I}\}.$$

The objective function in (39) aims at reducing the modified error  $\mathbf{E}'$ , while the first constraint trades the convergence rate of (33) with approximation accuracy.



**SIEVA<sub>1</sub> form.** Following the same idea as in (37)-(39), the modified fitting error of a SIEVA<sub>1</sub> graph filter is

$$e'_i = \tilde{h}_i - \tilde{h}_i (\mathbf{b}_{\mathcal{U},i}^A)^T \boldsymbol{\alpha}_1 - (\mathbf{b}_{\mathcal{U},i}^A)^T \boldsymbol{\alpha}_0, \quad (40)$$

with  $\tilde{h}_i$ ,  $(\mathbf{b}_{\mathcal{U},i}^A)^T \boldsymbol{\alpha}_0$ , and  $(\mathbf{b}_{\mathcal{U},i}^A)^T \boldsymbol{\alpha}_1$  denoting respectively the desired modal response and the eigenvalues of  $\tilde{\Phi}_0$  and  $\tilde{\Phi}_1$  w.r.t. the  $i$ th mode. In vector form, (40) is written as

$$\mathbf{e}' = \tilde{\mathbf{h}}_\lambda - \Psi_\lambda \bar{\boldsymbol{\alpha}}, \quad (41)$$

with  $\mathbf{e}' = [e'_1, \dots, e'_N]^T$ ,  $\tilde{\mathbf{h}}_\lambda = [\tilde{h}_1, \dots, \tilde{h}_N]^T$ ,  $\Psi_\lambda = [\mathbf{B}_{\mathcal{U}}^A, \text{diag}(\tilde{\mathbf{h}}_\lambda) \mathbf{B}_{\mathcal{U}}^A]$ , and  $\bar{\boldsymbol{\alpha}} = [\boldsymbol{\alpha}_0^T \boldsymbol{\alpha}_1^T]^T$ . Then,  $\{\boldsymbol{\alpha}_0, \boldsymbol{\alpha}_1\}$  can be estimated as the solution of the constrained least squares problem

$$\begin{aligned} & \underset{\boldsymbol{\alpha}_0, \boldsymbol{\alpha}_1 \in \mathbb{R}^d}{\text{minimize}} && \|\tilde{\mathbf{h}}_\lambda - \Psi_\lambda \boldsymbol{\alpha}\|_2^2 \\ & \text{subject to} && \|\mathbf{B}_{\mathcal{U}}^A \boldsymbol{\alpha}_1\|_\infty < \delta, \delta < 1, \boldsymbol{\alpha} = [\boldsymbol{\alpha}_0^T \boldsymbol{\alpha}_1^T]^T. \end{aligned} \quad (42)$$

Problem (42) again aims at minimizing the modified fitting error, while tuning the convergence rate through  $\delta$ .

Differently from the general EV-ARMA<sub>1</sub>, here the number of unknowns is reduced to  $2d$ , as now only the vectors  $\boldsymbol{\alpha}_0$  and  $\boldsymbol{\alpha}_1$  need to be designed. Therefore, due to this low dimensionality, one can also opt for global optimization solvers to find an acceptable local minimum of the true error (i.e., the equivalent of (37)).

**Remark 4.** *The approximation accuracy of the EV ARMA<sub>1</sub> filters can be further improved by following the Shank's method [39] used in [10, 38], or the iterative least-squares approach proposed in [40]. These methods have shown to improve the approximation accuracy of Prony's design by not only taking the modified fitting error into account but also the true one. However, as this idea does not add much to this work, interested readers are redirected to the above references for more details.*

## VI. NUMERICAL RESULTS

We now present a set of numerical examples to corroborate the applicability of the proposed filters for several distributed tasks. For convenience, Table I presents a summary of the different graph filters mentioned in this work along with their specifications. In our simulations<sup>2</sup>, we made use of the GSP toolbox [41].

### A. Graph Filter Approximation

We here test the proposed FIR graph filters in approximating a user-provided frequency response. We consider a random community graph of  $N = 256$  nodes and shift operator  $\mathbf{S} = \mathbf{L}$ . The frequency responses of interest are two commonly used responses in the GSP community, i.e.,

(i) the exponential kernel

$$\tilde{h}(\lambda) := e^{-\gamma(\lambda-\mu)^2},$$

with  $\gamma$  and  $\mu$  being the spectrum decaying factor and the central parameter respectively;

(ii) the ideal low-pass filter

$$\tilde{h}(\lambda) = \begin{cases} 1 & 0 \leq \lambda \leq \lambda_c \\ 0 & \text{otherwise,} \end{cases}$$

with  $\lambda_c$  being the cut-off frequency.

The approximation accuracy of the different filters is evaluated in terms of the normalized squared error  $\text{NSE} = \|\tilde{\mathbf{H}} - \mathbf{H}_{\text{fit}}\|_F^2 / \|\tilde{\mathbf{H}}\|_F^2$ .  $\mathbf{H}_{\text{fit}}$  stands for the filter matrix of the fitted filters.

Fig. 2 illustrates the performances of the different filters. In the exponential kernel scenario, we observe that the CEV FIR filter outperforms the other alternatives by showing a performance improvement of up to two orders of magnitude. A similar result is also seen in the low-pass example, where the CEV FIR filter achieves the error floor for  $K = 8$ , while the NV graph filter for  $K = 13$  and the *classical* FIR filter for  $K = 17$ . Additionally, we observe that the SIEV FIR filter achieves the same performance as the NV FIR filter. This result suggests that despite the additional DoF of the SIEV FIR filter, the nonconvex design strategy (22) yields in a local minimum that does not exploit the filter full capabilities.

The above observations further motivate the use of the CEV FIR filter, which trades off better the simplicity of the design and the available DoF. In fact, even though the CEV FIR filter has less DoF than the SIEV graph filter, it performs better than the latter. Similarly, the larger DoF of the CEV FIR filter compared to the NV FIR filter (i.e.,  $\text{nnz}(\mathbf{S}) \cdot K + N$  vs  $N \cdot (K + 1)$ ) allow the CEV FIR filter to better approximate the desired response. In a distributed setting, these benefits translate into communication and computational savings.

### B. Distributed Linear Operator Approximation

Several distributed tasks of interest consist of performing a linear operation  $\mathbf{A} \in \mathbb{R}^{N \times N}$  over a network. This can be for instance a beamforming matrix over a distributed array or consensus matrix. In most of these cases, such linear operators cannot be straightforwardly distributed. In this section, we illustrate the capabilities of the developed graph filters in addressing this task.

Given a desired linear operator  $\mathbf{A}$ , we aim at implementing this linear operator distributively through the solution of the optimization problem

$$\begin{aligned} & \underset{\boldsymbol{\theta}}{\text{minimize}} && \|\mathbf{A} - \mathbf{H}(\mathbf{S}, \boldsymbol{\theta})\| \\ & \text{subject to} && \boldsymbol{\theta} \in \Theta, \end{aligned} \quad (43)$$

where  $\mathbf{H}(\mathbf{S}, \boldsymbol{\theta})$  stands for the used graph filter parametrized by the shift  $\mathbf{S}$  and a set of parameters  $\boldsymbol{\theta}$  living in the domain  $\Theta$ .

**Distributed Consensus.** For distributed consensus, the operator  $\mathbf{A}$  has the form  $\mathbf{A} = \frac{1}{N} \mathbf{1} \mathbf{1}^T$ , which for  $\mathbf{S} = \mathbf{L}$  translates into a low-pass graph filter passing only the DC signal component.

Fig. 3 compares the fitting  $\text{NSE} = \|\mathbf{A} - \mathbf{H}_{\text{fit}}\|_F^2 / \|\tilde{\mathbf{H}}\|_F^2$  for the different FIR graph filters. We note once again that the CEV implementation offers the best approximation accuracy among the contenders achieving an NSE of order  $10^{-4}$  in only 10 exchanges. These results yield also different insights about the SIEV and SICEV graph filters.

First, both the SIEV and the SICEV implementations fail to compare well with the CEV, though the linear operator  $\mathbf{A}$  is shift invariant. We attribute this degradation in performance to assumption (A.1) necessary for these filters to have a modal

<sup>2</sup>The code to reproduce the figures in this paper can be found at <https://gitlab.com/fruzzi/graphFilterAdvances>

TABLE I: Summary of the different graph filters. (\*) indicates a contribution of this work. Here, numIt stands for the maximum number of iterations that the recursion is run.

Filter Type	Expression	Shift-Invariant	Design Strategy	Distributed Costs	Coefficients
Classical FIR [4]	$\mathbf{H}_c \triangleq \sum_{k=0}^K \phi_k \mathbf{S}^k$	always	LS [4], Chebyshev [5, 6]	$\mathcal{O}(MK)$	scalars: $\{\phi_k\}$
NV FIR [9]	$\mathbf{H}_{nv} \triangleq \sum_{k=0}^K \text{diag}(\phi_k) \mathbf{S}^k$	not in general	LS, convex program [9]	$\mathcal{O}(MK)$	vectors: $\{\phi_k\}$
General EV FIR (*)	$\mathbf{H}_{ev} \triangleq \sum_{k=1}^K (\Phi_k \dots \Phi_1)$	not in general	iterative design [Sec. III-C]	$\mathcal{O}(MK)$	matrices: $\{\Phi_k\}$
SIEV FIR (*)	(18)	always	iterative design [Sec. III-C]	$\mathcal{O}(MK)$	vectors: $\{\alpha_k\}$
CEV FIR (*)	$\mathbf{H}_{cev} \triangleq \sum_{k=1}^K \Phi_k \mathbf{S}^{k-1}$	not in general	LS [Sec. IV-C]	$\mathcal{O}(MK)$	matrices: $\{\Phi_k\}$
SICEV FIR (*)	(23)	always	LS [Sec. IV-C]	$\mathcal{O}(MK)$	vectors: $\{\alpha_k\}$
Classical ARMA <sub>1</sub> [10]	$\mathbf{H}_{arma_1} \triangleq \varphi(\mathbf{I} - \psi \mathbf{S})^{-1}$	always	closed-form, iterative design [10]	$\mathcal{O}(\text{numIt} \cdot M)$	scalars: $\{\varphi, \psi\}$
EV ARMA <sub>1</sub> (*)	$\mathbf{H}_{evarma_1} \triangleq (\mathbf{I} - \Phi_1)^{-1} \Phi_0$	not in general	two-step design [Sec. V-C]	$\mathcal{O}(\text{numIt} \cdot M)$	matrices: $\{\Phi_0, \Phi_1\}$
SIEVA <sub>1</sub> (*)	(35)	always	two-step design [Sec. V-C]	$\mathcal{O}(\text{numIt} \cdot M)$	vectors: $\{\alpha_0, \alpha_1\}$

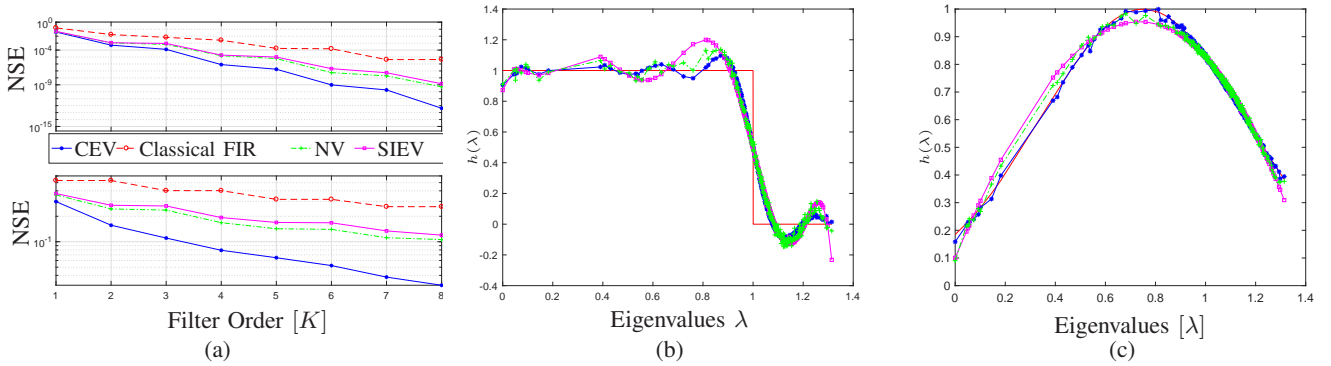


Fig. 2: (a) NSE vs. filter order for different FIR graph filters. (Top) Results in approximating a low-pass response. (Bottom) Results in approximating the exponential kernel response. (b) Frequency response of the graph filters when approximating a perfect low pass filter. (c) Frequency response of the graph filters when approximating an exponential kernel with parameters  $\mu = 0.75$  and  $\gamma = 3$ .

response interpretation. In fact, forcing each filter coefficient matrix to be shift invariant seems limiting the filter ability to match well the consensus operator.

Second, the different design strategies used in SIEV and SICEV further discriminate the two filters. We can see that the least squares design of the SICEV implementation is more beneficial, though the SIEV filter has more DoF. Unfortunately, this is the main drawback of the latter graph filter, which due to the nonconvexity of the design problem leads to suboptimal solutions. However, we remark that both these filters outperform (or compare equally with) the classical FIR filter. Further investigation in this direction is needed to understand if the SIEV, or SICEV structures can be used to achieve finite-time consensus as carried out in [42, 43].

**Wiener-based denoising.** For instances when the statistics of the graph signal and noise signal are available, a typical approach for performing denoising is the Wiener filter. This filter is obtained by minimizing the mean-squared error, i.e.,

$$\tilde{\mathbf{H}} = \underset{\mathbf{H} \in \mathbb{R}^{N \times N}}{\text{argmin}} \mathbb{E}[\|\mathbf{H}\mathbf{z} - \mathbf{x}\|_2^2], \quad (44)$$

where  $\mathbf{z} = \mathbf{x} + \mathbf{n}$  is the graph signal corrupted with additive noise. For the case of zero-mean signals  $\mathbf{x}$  and  $\mathbf{n}$  with covariance matrices  $\Sigma_x$  and  $\Sigma_n$ , respectively, the solution for (44) is

$$\tilde{\mathbf{H}} = \Sigma_x (\Sigma_x + \Sigma_n)^{-1}, \quad (45)$$

given  $\Sigma_x + \Sigma_n$  is not singular. When the covariance matrices

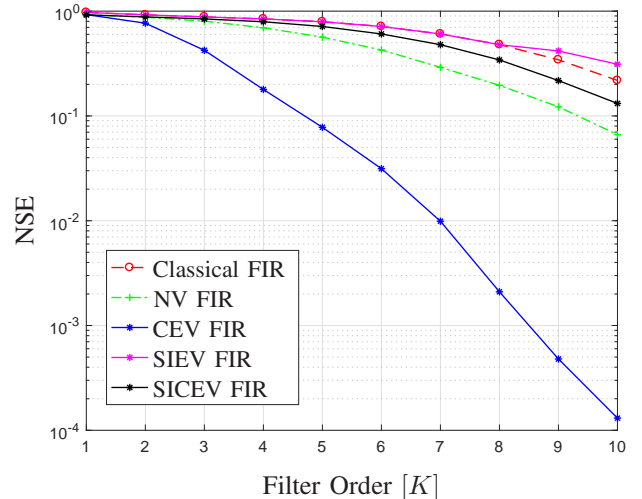


Fig. 3: NSE versus filter order for different distributed FIR filter implementations when approximating the consensus operator  $\mathbf{H} = \mathbf{1}/N\mathbf{1}\mathbf{1}^T$ .

$\Sigma_x$  and  $\Sigma_n$  share the eigenvectors with the graph shift operator, the optimal filter  $\tilde{\mathbf{H}}$  can be applied through classical graph filters. However, in many instances, the signal covariance matrix  $\Sigma_x$  is not diagonalizable by the eigenvectors of  $\mathbf{S}$  [44]. When  $\tilde{\mathbf{H}}$  is not jointly diagonalizable, a typical approach is to consider only the diagonal entries of its projection onto the

eigenvectors of the shift operator,  $\mathbf{D} = \mathbf{U}^{-1}\tilde{\mathbf{H}}\mathbf{U}$ . Then, a filter  $\tilde{\mathbf{H}} = \mathbf{U}\text{diag}([D]_{11}, \dots, [D]_{NN})\mathbf{U}^{-1}$  is used instead of  $\tilde{\mathbf{H}}$  as an approximation. For cases where  $\mathbf{D}$  is approximately diagonal this is a good way to approximate the Wiener filter in a distributable manner. However, for general matrices  $\tilde{\mathbf{H}}$  this is not a necessary good approach.

We illustrate an example where instead of approximating the Wiener filter through a classical FIR graph filter, we employ a CEV FIR filter. For this example we consider the Molene dataset<sup>3</sup>, where the temperature data of several cities in France has been recorded. The graph employed is taken from [45] and the graph signal has been corrupted with white Gaussian noise. The results in terms of NSE for the different fitted graph filters are shown in Fig. 4. From this plot we observe that the CEV FIR filter outperforms all the other alternatives. This is due to the fact that the optimal Wiener filter is not jointly diagonalizable with the eigenbasis of the shift operator, i.e., covariance matrix of data is not shift invariant, hence classical graph filters are not appropriate to approximate the filter.

**Distributed Beamforming.** We here consider the task of applying a beamforming matrix  $\mathbf{W}^H$  to signals acquired on a distributed array. More specifically, we aim at obtaining the output

$$\mathbf{y} = \mathbf{W}^H \mathbf{x}, \quad (46)$$

where  $\mathbf{x}$  is the data acquired in a distributed way. Since  $\mathbf{W}^H$  might often be a dense matrix, e.g., in zero-forcing beamforming, operation (46) cannot be readily distributed. To obtain the output at each node, we approximate the beamforming matrix with different graph filters.

We quantify this scenario in a distributed 2D sensor array. The network is generated using  $N = 40$  random locations on a 2D plane where the communication network is an 8-nearest neighbors graph. The beamforming matrix is the matched filter [46] matrix for a uniform angular grid of  $N = 40$  points in the range  $(-180, 180]$ . In other words, every node will see the information from a small sector of approximately nine degrees. Since in general  $\mathbf{W}^H$  does not share the eigenbasis with  $\mathbf{S}$ , classical graph filters fail to address this task. Therefore, here we compare only the CEV FIR filter and the NV FIR filter. Fig. 5 shows two output beampatterns obtained by solving (43) with  $\mathbf{A} = \mathbf{W}^H$  for the two considered filters with order  $K = 5$ . We notice that the CEV outperforms the NV FIR filter as it follows more closely the desired beampattern.

Note that the above framework treats the distributed beamforming differently from approaches based on distributed optimization tools [47]. The latter methods usually aim at computing the beamforming matrix (i.e., the weighting matrix is data dependent) and then perform consensus. On the other hand we assume that  $\mathbf{W}^H$  is fixed and that it must be applied to the array data. However, this problem can also be solved through distributed convex optimization tools by solving the least squares problem

$$\underset{\mathbf{y}}{\text{minimize}} \quad \|\mathbf{x} - (\mathbf{W}^H)^\dagger \mathbf{y}\|_2^2. \quad (47)$$

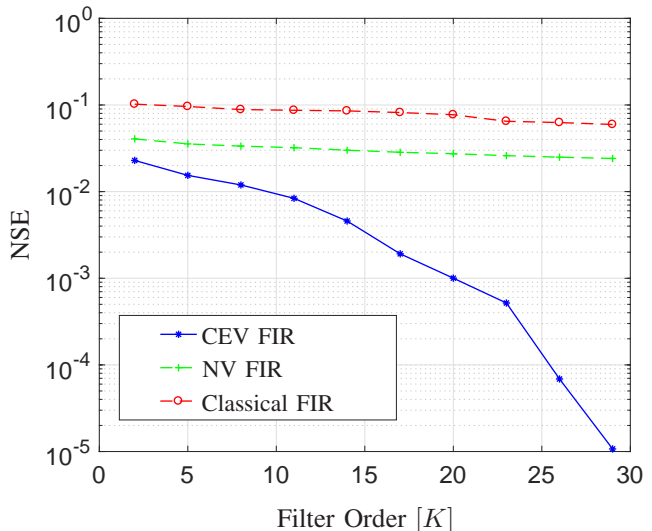


Fig. 4: NSE versus filter order for different distributed FIR filter implementations when approximating the Wiener Filter for the Molene temperature dataset.

Differently from (43), formulation avoids the computation of the pseudo-inverse and the graph-filtering based approach requires only five iterations to compute the final beampattern.

In the upcoming section, we compare the CEV and the NV graph filters with distributed optimization tools in solving a general inverse problem.

### C. Comparison with Distributed Optimization

We now compare the proposed graph filters with the primal dual method of multipliers (PDMM)<sup>4</sup> [48] solving distributively the least squares problem

$$\underset{\mathbf{x}}{\text{minimize}} \quad \|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2 \quad (48)$$

for a matrix  $\mathbf{A}$ . Without loss of generality we consider  $\mathbf{A}$  to be an  $N \times N$  matrix. The baseline assumption for all distributed optimization methods is that  $v_i$  knows its own regressor, i.e., the  $i$ th row of  $\mathbf{A}$ ,  $\mathbf{a}_i^\top$ . The task is that each node retrieves the full vector  $\mathbf{x}_{is} = \mathbf{A}^\dagger \mathbf{y}$  by means of local communications.

For the graph filter-based approaches, we approximate  $\mathbf{A}^\dagger$  through a set of rank one matrices  $\{\tilde{\mathbf{H}}_i \triangleq \mathbf{1}\tilde{\mathbf{a}}_i^\top\}_{i=1}^N$  with  $\tilde{\mathbf{a}}_i$  being the  $i$ th row of  $\mathbf{A}^\dagger$ . This means that in contrast to distributed optimization methods, here every node  $v_i$  needs to know the full  $\mathbf{A}$ . Each  $\tilde{\mathbf{H}}_i$  is then fitted with the NV and CEV recursions to approximate  $\mathbf{x}_{is}$  as the output after filtering the graph signal  $\mathbf{y}$ . It must be noticed that the number of communications between adjacent nodes does not scale with  $N$ . In fact, both the NV and the CEV will shift the signal only  $K$  times and the nodes can locally apply the respective coefficients to obtain the outputs.

To quantify the performance, we perform 100 Monte Carlo simulations with a randomly generated system matrix and solution vector. Fig. 6 compares the graph filter approaches with the distributed optimization methods in terms of the NSE  $\|\mathbf{x} - \hat{\mathbf{x}}^{(k)}\|_2^2 / \|\mathbf{x}\|_2^2$ . The graph filter methods achieve a faster decay compared to the distributed optimization method

<sup>3</sup>Access to the raw data through the link [donnees.meteofrance.fr/donnees\\_libres/Hackathon/RADOMEH.tar.gz](https://donnees.meteofrance.fr/donnees_libres/Hackathon/RADOMEH.tar.gz)

<sup>4</sup>PDMM is an alternative distributed optimization tool to the classical alternating direction method of multipliers (ADMM), which is often characterized by a faster converge [48].

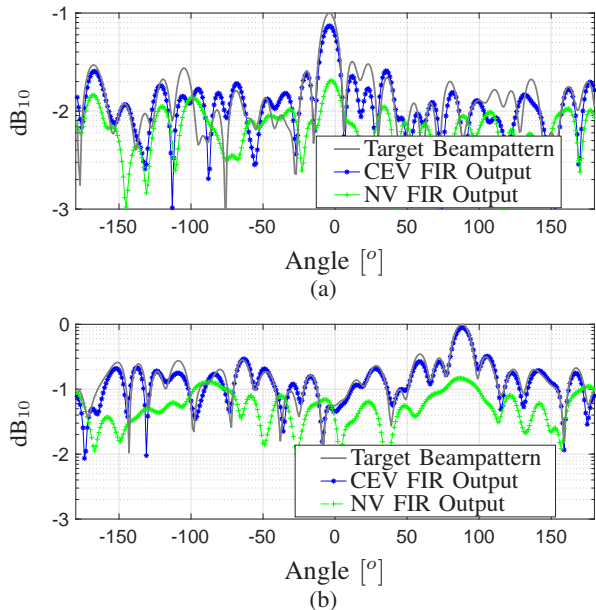


Fig. 5: Comparison of beampatterns for different node outputs and desired steering angle,  $\theta_0$ . Beampattern for beamformer focusing at (a)  $\theta_0 = 0^\circ$  and (b)  $\theta_0 = 90^\circ$ .

in the first hundred iterations. However, since perfect approximation of the desired response is not possible both graph filters exhibit an error floor. PDMM, on the other hand, does not run into this issue and guarantees convergence to the true solution. Despite this difference in performance, the graph filter approaches can be employed for cases where the accuracy requirements are not strict, or as *warm starts* for the distributed optimization methods. The above comparison, besides proposing graph filters as an alternative for solving distributed least squares problems, raises the question on *how graph filters relate to distributed convex optimization*. In fact, further research is needed to relate the design and implementation of distributed EV graph filters with the well-established theory of distributed optimization.

#### D. Tikhonov-based denoising

One of the central problems in GSP is that of recovering an unknown signal  $\mathbf{x}$  from a noisy realization  $\mathbf{z} = \mathbf{x} + \mathbf{n}$  given that  $\mathbf{x}$  is smooth w.r.t. the underlying graph [3]. Differently known as the Tikhonov denoiser, the estimation of  $\mathbf{x}$  can be obtained by solving the regularized least squares problem

$$\mathbf{x}^* = \arg \min_{\mathbf{x} \in \mathbb{R}^N} \|\mathbf{z} - \mathbf{x}\|_2^2 + \mu \mathbf{x}^T \mathbf{S} \mathbf{x}, \quad (49)$$

for  $\mathbf{S} = \mathbf{L}$  and where  $\mu$  trades off the noise removal with the smoothness prior. Problem (49) has the well-known solution  $\mathbf{x}^* = (\mathbf{I} + \mu \mathbf{S})^{-1} \mathbf{z}$ , which in terms of the terminology used in Section II is an ARMA<sub>1</sub> graph filter with  $\varphi = 1$  and  $\psi = -\mu$  (see also [10] for further analysis). While recursion (6) can implement this problem distributively, the convergence of the Neumann series in (7) cannot be controlled as the rate is fixed by  $|\mu| \lambda_{\max}\{\mathbf{S}\}$ .

Here, we show that through the EV ARMA<sub>1</sub> (33) it is possible to improve the convergence speed of the ARMA<sub>1</sub> graph filter by exploiting the additional DoF given by the

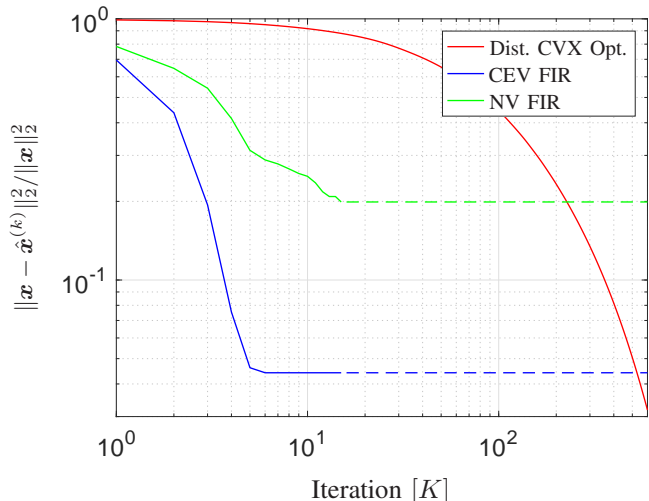


Fig. 6: Convergence error versus the number of iterations for the NV and the CEV graph filters and for the PDMM solver [48]. Dashed lines indicates the saturation floor of the NV and CEV FIRs.

edge-weighting matrices  $\{\Phi_0, \Phi_1\}$ . However, since now the design is not exact and involves the modified error [cf. (39)], this speed benefit will come at the expense of accuracy. To illustrate this, we consider an example of problem (49) with  $\mu = 0.8$  and  $\mathbf{S} = \lambda_{\max}^{-1}(\mathbf{L})\mathbf{L}$ , such that  $\mathbf{S}$  has unitary spectral norm. Fig. 7 shows the convergence error of the EV ARMA<sub>1</sub> for different values of  $\delta$  in (39) and compares it with the classical ARMA<sub>1</sub> and the CEV of order  $K = 15$ .

We make the following observations. First, low values of  $\delta$  are preferred to improve the convergence speed. However, values below 0.7 should in general be avoided since this restricts too much the feasible set of (39), hence leading to a worse approximation error. Second, values of  $\delta \approx 0.7$  seem to give the best tradeoff, since the convergence speed is doubled w.r.t the ARMA<sub>1</sub> and the approximation error is close to machine precision. Additionally, the fact that the solution  $\delta = 0.7$  achieves a better performance than the solution with  $\delta = 0.8$  arises from the fact that due to the two-step procedure, the solution obtained by minimizing the modified error might not lead to the best matrix for minimizing the true error during the second step. Finally, we did not plot the classical FIR filter for solving this problem, since its performance is identical to the ARMA<sub>1</sub> for the same distributed costs [10].

## VII. CONCLUSIONS

In this work, a generalization of the distributed graph filters was proposed. These filters, that we referred to as edge-variant graph filters, have the ability to assign different weights to the information coming from different neighbors. Through the design of edge-weighting matrices, we have shown that it is possible to weigh, possibly in an asymmetric fashion, the information propagated in the network and improve the performance of state-of-the-art graph filters.

By introducing the notion of filter modal response, we showed that a subclass of the edge-variant graph filters have a graph Fourier interpretation that illustrates the filter action on the graph modes. Despite that the most general edge-variant graph filter encounters numerical challenges in the design phase, a constrained version of it was introduced to tackle this

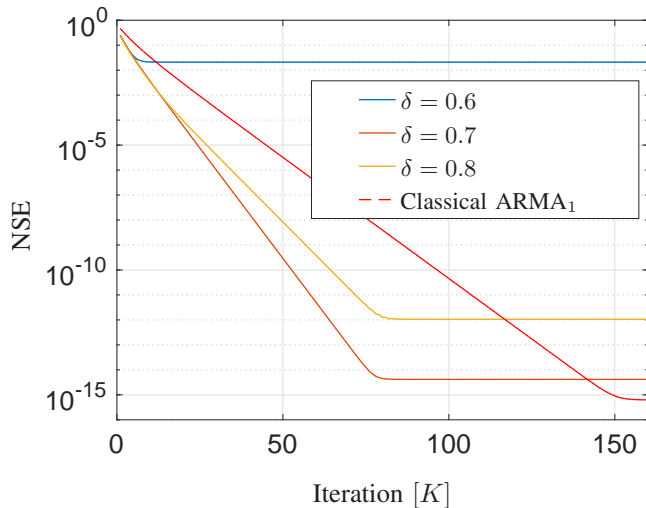


Fig. 7: Convergence error versus the number of iterations for the Tikhonov denoising problem. The EV ARMA<sub>1</sub> results are plotted for different values of  $\delta$  in (39) to highlight the tradeoff between convergence speed and approximation accuracy.

issue. The so-called constrained edge-variant graph filter still enjoys a similar distributed implementation, generalizes the state-of-the-art approaches, and is characterized by a simple least squares design. For the constrained version, we also showed that there exists a subclass which has a modal response interpretation.

Finally, we extended the edge-variant idea to the family of IIR graph filters, particularly to the ARMA<sub>1</sub> graph filter. We showed that by adopting the same local structure a distributed rational filter can be achieved, yet with a much faster convergence speed. Several numerical tests corroborate our findings and show the potential of the proposed filters to improve state-of-the-art techniques.

Future research in this direction should concern the following points: *i*) improve the design strategy for the more general edge-variant version; *ii*) improve the saturation accuracy of the proposed methods when dealing with a distributed implementation of linear operators; *iii*) conciliate the world of GSP with that of distributed optimization and exploit the latter to design distributed graph filters; and *iv*) extend the edge-variant concept beyond the ARMA<sub>1</sub> implementation to the global family of IIR graph filters.

#### REFERENCES

- [1] M. Coutino, E. Isufi, and G. Leus, “Distributed edge-variant graph filters,” in *IEEE 7th Int. Workshop Comp. Adv. in Multi-Sensor Adap. Proc.(CAMSAP)*. IEEE, 2017.
- [2] G. Taubin, “Geometric signal processing on polygonal meshes,” in *EUROGRAPHICS*, 2000.
- [3] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst, “The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains,” *IEEE Sig. Proc. Mag.*, vol. 30, no. 3, pp. 83–98, 2013.
- [4] A. Sandryhaila and J. M. Moura, “Discrete signal processing on graphs,” *IEEE Trans. Signal Process.*, vol. 61, no. 7, pp. 1644–1656, 2013.

- [5] G. Taubin, T. Zhang, and G. Golub, “Optimal surface smoothing as filter design,” in *European Conf. on Computer Vision*. Springer, 1996, pp. 283–292.
- [6] D. I. Shuman, P. Vandergheynst, and P. Frossard, “Distributed signal processing via chebyshev polynomial approximation,” *arXiv preprint arXiv:1111.5239*, 2011.
- [7] S. K. Narang, A. Gadde, and A. Ortega, “Signal processing techniques for interpolation in graph structured data,” in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*. IEEE, 2013, pp. 5445–5449.
- [8] M. Onuki, S. Ono, M. Yamagishi, and Y. Tanaka, “Graph signal denoising via trilateral filter on graph spectral domain,” *IEEE Trans. on Sig. and Inf. Proc. over Netw.*, vol. 2, no. 2, pp. 137–148, 2016.
- [9] S. Segarra, A. Marques, and A. Ribeiro, “Optimal graph-filter design and applications to distributed linear network operators,” *IEEE Trans. Signal Process.*, 2017.
- [10] E. Isufi, A. Loukas, A. Simonetto, and G. Leus, “Autoregressive moving average graph filtering,” *IEEE Trans. Signal Process.*, vol. 65, no. 2, pp. 274–288, 2017.
- [11] D. I. Shuman, B. Ricaud, and P. Vandergheynst, “Vertex-frequency analysis on graphs,” *Applied and Computational Harmonic Analysis*, vol. 40, no. 2, pp. 260–291, 2016.
- [12] A. Sandryhaila and J. M. Moura, “Discrete signal processing on graphs: Frequency analysis,” *IEEE Trans. Signal Processing*, vol. 62, no. 12, pp. 3042–3054, 2014.
- [13] M. Belkin, P. Niyogi, and V. Sindhvani, “Manifold regularization: A geometric framework for learning from labeled and unlabeled examples,” *Journal of machine learning research*, vol. 7, no. Nov, pp. 2399–2434, 2006.
- [14] J. Ma, W. Huang, S. Segarra, and A. Ribeiro, “Diffusion filtering of graph signals and its use in recommendation systems,” in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*. IEEE, 2016, pp. 4563–4567.
- [15] B. Girault, P. Gonçalves, E. Fleury, and A. S. Mor, “Semi-supervised learning for graph to signal mapping: A graph signal wiener filter interpretation,” in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*. IEEE, 2014, pp. 1115–1119.
- [16] E. Isufi, P. Di Lorenzo, P. Banelli, and G. Leus, “Distributed wiener-based reconstruction of graph signals,” in *IEEE Stat. Sig. Proc. (SSP) Workshop*, 2018.
- [17] F. Zhang and E. R. Hancock, “Graph spectral image smoothing using the heat kernel,” *Pattern Recognition*, vol. 41, no. 11, pp. 3328–3342, 2008.
- [18] A. C. Yağın and M. T. Özgen, “A spectral graph wiener filter in graph fourier domain for improved image denoising,” in *Sig. and Inf. Proc. (GlobalSIP), 2016 IEEE Global Conference on.* IEEE, 2016, pp. 450–454.
- [19] E. Isufi and G. Leus, “Distributed sparsified graph filters for denoising and diffusion tasks,” in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*. IEEE, 2017, pp. 5865–5869.
- [20] N. Tremblay, G. Puy, R. Gribonval, and P. Vandergheynst, “Compressive spectral clustering,” in *Int. Conf. on Machine Learning*, 2016, pp. 1002–1011.
- [21] D. B. Tay and Z. Lin, “Design of near orthogonal graph filter banks,” *IEEE Sig. Proc. Lett.*, vol. 22, no. 6, pp.

- 701–704, 2015.
- [22] O. Teke and P. P. Vaidyanathan, “Extending classical multirate signal processing theory to graphs part ii: M-channel filter banks,” *IEEE Trans. Signal Process.*, vol. 65, no. 2, pp. 423–437, 2017.
- [23] D. K. Hammond, P. Vandergheynst, and R. Gribonval, “Wavelets on graphs via spectral graph theory,” *Applied and Computational Harmonic Analysis*, vol. 30, no. 2, pp. 129–150, 2011.
- [24] M. Defferrard, X. Bresson, and P. Vandergheynst, “Convolutional neural networks on graphs with fast localized spectral filtering,” in *Advances in Neural Information Processing Systems*, 2016, pp. 3844–3852.
- [25] F. Gama, A. G. Marques, G. Leus, and A. Ribeiro, “Convolutional neural networks architectures for signals supported on graphs,” *arXiv preprint arXiv:1805.00165*, 2018.
- [26] X. Shi, H. Feng, M. Zhai, T. Yang, and B. Hu, “Infinite impulse response graph filters in wireless sensor networks,” *IEEE Sig. Proc. Lett.*, vol. 22, no. 8, pp. 1113–1117, 2015.
- [27] E. Isufi, A. Loukas, A. Simonetto, and G. Leus, “Filtering Random Graph Processes Over Random Time-Varying Graphs,” *IEEE Trans. Signal Process.*, 2017.
- [28] L. L. Magoarou and R. Gribonval, “Flexible multilayer sparse approximations of matrices and applications,” *IEEE Journal of Sel. Topics in Sig. Proc.*, vol. 10, no. 4, pp. 688–700, June 2016.
- [29] S. Barbarossa, G. Scutari, and T. Battisti, “Distributed signal subspace projection algorithms with maximum convergence rate for sensor networks with topological constraints,” in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*. IEEE, 2009.
- [30] K. Van Acker, G. Leus, M. Moonen, O. Van de Wiel, and T. Pollet, “Per tone equalization for dmt-based systems,” *IEEE Trans. Commun.*, vol. 49, no. 1, pp. 109–119, 2001.
- [31] E. Isufi and G. Leus, “Distributed sparsified graph filters for denoising and diffusion tasks,” in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*. IEEE, 2017, pp. 5865–5869.
- [32] V. Kalofolias, A. Loukas, D. Thanou, and P. Frossard, “Learning time varying graphs,” in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*. IEEE, 2017, pp. 2826–2830.
- [33] M. Kolar, L. Song, A. Ahmed, and E. P. Xing, “Estimating time-varying networks,” *The Annals of App. Stats.*, pp. 94–123, 2010.
- [34] M. Coutino, S. Chepuri, and G. Leus, “Sparsest network support estimation: A submodular approach,” in *2018 IEEE Data Science Workshop*, 2018.
- [35] Y. Xu and W. Yin, “A block coordinate descent method for regularized multiconvex optimization with applications to nonnegative tensor factorization and completion,” *SIAM Journal on imaging sciences*, vol. 6, no. 3, pp. 1758–1789, 2013.
- [36] *MATLAB Optimization Toolbox*, The MathWorks, Inc., Natick, Massachusetts, United States.
- [37] C. Manss, D. Shutin, and G. Leus, “Distributed splitting-over-features sparse bayesian learning with alternating direction method of multipliers,” in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, 2017.
- [38] E. Isufi, A. Loukas, and G. Leus, “Autoregressive moving average graph filters a stable distributed implementation,” in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, 2017.
- [39] J. L. Shanks, “Recursion filters for digital processing,” *Geophysics*, vol. 32, no. 1, pp. 33–51, 1967.
- [40] J. Liu, E. Isufi, and G. Leus, “Filter design for autoregressive moving average graph filters,” *arXiv preprint arXiv:1711.09086*, 2017.
- [41] N. Perraudin, J. Paratte, D. Shuman, L. Martin, V. Kalofolias, P. Vandergheynst, and D. K. Hammond, “GSP-BOX: A toolbox for signal processing on graphs,” *ArXiv e-prints*, Aug. 2014.
- [42] L. Wang and F. Xiao, “Finite-time consensus problems for networks of dynamic agents,” *IEEE Trans. on Autom. Control*, vol. 55, no. 4, pp. 950–955, 2010.
- [43] A. Sandryhaila, S. Kar, and J. M. Moura, “Finite-time distributed consensus through graph filters,” in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*. IEEE, 2014, pp. 1080–1084.
- [44] A. G. Marques, S. Segarra, G. Leus, and A. Ribeiro, “Stationary graph processes and spectral estimation,” *IEEE Trans. Signal Process.*, 2017.
- [45] N. Perraudin and P. Vandergheynst, “Stationary signal processing on graphs,” *IEEE Trans. Signal Process.*, vol. 65, no. 13, pp. 3462–3477, 2017.
- [46] L. Harry and V. Trees, “Optimum array processing,” *Part IV of Detection, Estimation and Modulation Theory*, 2002.
- [47] T. Sherson, W. B. Kleijn, and R. Heusdens, “A distributed algorithm for robust lcmv beamforming,” in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*. IEEE, 2016, pp. 101–105.
- [48] G. Zhang and R. Heusdens, “Distributed optimization using the primal-dual method of multipliers,” *IEEE Trans. on Sig. and Inf. Proc. over Netw.*, vol. 4, no. 1, pp. 173–187, 2018.