Distributed edge-variant graph filters

Coutino, Mario; Isufi, Elvin; Leus, Geert

DOI
10.1109/CAMSAP.2017.8313105

Publication date
2018

Document Version
Final published version

Published in
2017 IEEE 7th International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)

Citation (APA)
Coutino, M., Isufi, E., & Leus, G. (2018). Distributed edge-variant graph filters. In *2017 IEEE 7th International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)* (pp. 1-5). IEEE. https://doi.org/10.1109/CAMSAP.2017.8313105

DISTRIBUTED EDGE-VARIANT GRAPH FILTERS

Mario Coutino, Elvin Isufi, and Geert Leus

Delft University of Technology, Delft, The Netherlands.

ABSTRACT

The main challenges distributed graph filters face in practice are the communication overhead and computational complexity. In this work, we extend the state-of-the-art distributed finite impulse response (FIR) graph filters to an edge-variant (EV) version, i.e., a filter where every node weights the signals from its neighbors with different values. Besides having the potential to reduce the filter order leading to amenable communication and complexity savings, the EV graph filter generalizes the class of classical and node-variant FIR graph filters. Numerical tests validate our findings and illustrate the potential of the EV graph filters to (i) approximate a user-provided frequency response; and (ii) implement distributed consensus with much lower orders than its direct contenders.

Index Terms— Edge-variant filters, FIR graph filters, graph filters, finite-time consensus, graph signal processing.

1. INTRODUCTION

With the increasing demand to process large amounts of data living on irregular domains, such as social, biological, or sensor networks, graph signal processing (GSP) [1], emerged as an exciting field that processes these network data (signals) by naturally incorporating their relations represented by a graph. By providing a specific definition of the graph Fourier transform (GFT), in GSP we can process/analyze these signals with tools from harmonic analysis. In this context, graph filters [1–3] arise as the basic building block to shape a graph signal by preserving different parts of its spectrum.

Graph filters can improve tasks of signal denoising and interpolation [1–6], smoothing [7], and diffusion [8]. Furthermore, by means of low-pass graph filtering a signal, the task of graph clustering [9] can be performed as in [10]. To improve the scalability of computation, or to perform filtering in sensor networks, graph filters can be implemented distributively either in their finite impulse response (FIR) [2, 3, 11], or autoregressive moving average (ARMA) form [12–15].

In both the centralized and distributed implementations, high-order FIR graph filters are often required to accurately approximate a given frequency response leading to high implementation costs. While ARMA graph filters are a possible choice to tackle this problem, in this work we focus on alternative forms of FIR graph filters which avoid stability issues that might arise with the ARMA implementations. Similar approaches have been considered in [11], which proposes the so-called node-variant (NV) graph filters, and in [16, 17] in the context of stochastically sparsified graph filters. NV graph filters generalize *classical* FIR graph filters, where for each filter shift different nodes can have different weighting coefficients, yet a node applies the same weight to all its neighboring signals [11]. The stochastic sparsification on the other hand, considers a random edge sampling at each filter shift reducing the filter cost, yet its output has a stochastic nature with first and second order moments respectively unbiased and bounded [16].

In this paper, we propose a generalization of the NV graph filters named edge-variant (EV) graph filters, where for each filter shift different nodes can also apply different weights to their neighboring signals. This improvement of the degrees of freedom (DoF) allows a potential reduction of the filter order, and hence of the implementation cost. We first introduce the most general EV graph filter and highlight the challenges in its design phase. Then, we present the constrained EV (C-EV) graph filter which allows a more tractable design and still specializes the *classical* FIR graph filter [3] where the for every shift, the weights are equal for all nodes and edges, and to the NV graph filter [11] where for every shift each node applies a constant weight for all its edges.

Our numerical tests show that the proposed EV graph filter approximates a desired frequency response, including consensus, with lower filter orders w.r.t. its contenders leading to lower computational and communication costs.

2. PRELIMINARIES

This section covers some background information that is used throughout the paper. We start with the basics of graph signal processing and then recall the *classical* FIR and NV graph filters.

Graph signal processing (GSP). Consider a signal $\boldsymbol{x} \in \mathbb{R}^N$ on an undirected[1] graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with $\mathcal{V}$ the set of N vertices and $\mathcal{E}$ the set of M edges. Let also $\boldsymbol{W}$ and $\boldsymbol{L}$ represent the weighted graph adjacency matrix and the corresponding graph Laplacian. Both $\boldsymbol{W}$ and $\boldsymbol{L}$ are valid candidates for the so-called graph shift operator $\boldsymbol{S}$, a symmetric matrix that carries the notion of frequency in the graph setting [1, 3]. Specifically, given the decomposition of $\boldsymbol{S} = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^\top$, the graph Fourier transform (GFT) of $\boldsymbol{x}$ is interpreted as its projection on the modes of $\boldsymbol{S}$, i.e., $\hat{\boldsymbol{x}} = \boldsymbol{U}^\top\boldsymbol{x}$. Similarly, $\boldsymbol{U}$ being an orthonormal matrix, the inverse graph Fourier transform is given by $\boldsymbol{x} = \boldsymbol{U}\hat{\boldsymbol{x}}$. Following the GSP convention, the eigenvalues $\boldsymbol{\Lambda} = \text{diag}(\lambda_1, \ldots, \lambda_N)$ are commonly referred to as the graph frequencies.

FIR graph filters. The output of a linear shift invariant graph filter in the frequency domain is given by

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.

[1] We focus on undirected graphs for simplicity of representation. However, the paper contributions are valid also for directed graphs by making use of the Jordan decomposition for instance [3].

978-1-5386-1251-4/17/$31.00 ©2017 IEEE

$$\hat{\boldsymbol{y}} = h(\boldsymbol{\Lambda})\hat{\boldsymbol{x}}, \tag{1}$$

where $h(\boldsymbol{\Lambda})$ is a diagonal matrix with the frequency response of the filter on its diagonal. Applying the inverse GFT on both sides of (1) leads to the vertex domain filter output

$$\boldsymbol{y} = \boldsymbol{H}\boldsymbol{x}, \tag{2}$$

with $\boldsymbol{H} = \boldsymbol{U}h(\boldsymbol{\Lambda})\boldsymbol{U}^{\mathsf{T}}$. A particular form of $\boldsymbol{H}$ is its expression as a polynomial in the graph shift operator [3], i.e.,

$$\boldsymbol{H}_{\mathrm{c}} \triangleq \sum_{k=0}^{K} \phi_k \boldsymbol{S}^k, \tag{3}$$

which we refer to as the *classical* FIR graph filter. Next to the centralized implementation of the graph filter (1), it is also possible to run it distributively due to the locality of $\boldsymbol{S}$ [2, 11]. Specifically, since $\boldsymbol{S}^k\boldsymbol{x} = \boldsymbol{S}(\boldsymbol{S}^{k-1}\boldsymbol{x})$ the nodes can compute locally the kth shift of $\boldsymbol{x}$ from the $(k-1)$th shift of $\boldsymbol{x}$ in their local neighborhood. Overall, for an FIR filter of order K this amounts to a computational and communication complexity of $\mathcal{O}(MK)$.

Node-variant graph filters. From [11], a NV graph filter in the vertex domain has the form

$$\boldsymbol{H}_{\mathrm{nv}} \triangleq \sum_{k=0}^{K} \mathrm{diag}(\boldsymbol{\phi}_k)\boldsymbol{S}^k, \tag{4}$$

where the vector $\boldsymbol{\phi}_k = [\phi_{k,1}, \ldots, \phi_{k,N}]^T$ contains the node coefficients to be applied at the kth shift. It is easy to see that for $\boldsymbol{\phi}_k = \phi_k \mathbf{1}$ the NV graph filter (4) reduces to the *classical* FIR graph filter (3). Similar to (3), the NV filter (4) preserves its efficient implementation since the different node coefficients are directly applied to the kth shifted input $\boldsymbol{S}^k\boldsymbol{x} = \boldsymbol{S}(\boldsymbol{S}^{k-1}\boldsymbol{x})$ keeping also a computational complexity of $\mathcal{O}(MK)$.

For more details about the state-of-the-art classical FIR and NV graph filters we redirect the reader to [2, 3, 11].

Since a desired graph filter $\tilde{\boldsymbol{H}}$ is usually approximated by means of a matrix polynomial such as (3), in accuracy-demanding applications, like the ones mentioned in the previous section, the filter order K is required to be large leading to computational and communication costs that scale linearly with K. To overcome these issues, the NV graph filters provide a solution. However, in the sequel we take it one step further and generalize (4) to an EV graph filter alternative, which due to its higher degrees of freedom can approximate a graph filter $\tilde{\boldsymbol{H}}$ with a much lower order K leading to a more efficient implementation. A benefit of both the NV and the EV graph filters, is that they can approximate a broader class of operators $\tilde{\boldsymbol{H}}$ which not necessarily share the eigenvectors of the shift operator, such as the analog network coding proposed in [11].

3. EDGE-VARIANT GRAPH FILTERS

Similar to NV graph filters [11], it is possible to define a different kind of graph filters in the vertex domain where for each node, the contribution of its neighbors is weighted independently. Such EV graph filters can be defined as

$$\begin{aligned}\boldsymbol{H}_{\mathrm{ev}} \triangleq\; & \boldsymbol{\Phi}_0 + \boldsymbol{\Phi}_1 \odot \boldsymbol{S} + (\boldsymbol{\Phi}_2 \odot \boldsymbol{S})(\boldsymbol{\Phi}_1 \odot \boldsymbol{S}) + \ldots \\ & + (\boldsymbol{\Phi}_K \odot \boldsymbol{S})(\boldsymbol{\Phi}_{K-1} \odot \boldsymbol{S}) \cdots (\boldsymbol{\Phi}_1 \odot \boldsymbol{S}) \\ =\; & \sum_{k=1}^{K} \prod_{j=1}^{k} (\boldsymbol{\Phi}_j \odot \boldsymbol{S}) + \boldsymbol{\Phi}_0,\end{aligned} \tag{5}$$

where $\boldsymbol{\Phi}_j \in \mathbb{R}^{N\times N}$ is an edge-weighting matrix that applies different weights to the entries of $\boldsymbol{S}$ by means of the Hadamard product "$\odot$" . The support of $\boldsymbol{\Phi}_j$ and $\boldsymbol{S}$ is the same. The matrix $\boldsymbol{\Phi}_0$ is considered to be a diagonal matrix. Notice that in this definition, no symmetry of the matrices $\boldsymbol{\Phi}_j$ is assumed. Depending on which information is seen *locally* by each node, the applied weights might not be required to be symmetric.

As a result, for a given desired response $\tilde{\boldsymbol{y}}$ given by

$$\tilde{\boldsymbol{y}} = \tilde{\boldsymbol{H}}\boldsymbol{x}, \tag{6}$$

an edge-variant graph filter $\boldsymbol{H}_{\mathrm{ev}}$ [cf. (5)] is designed with the aim of solving the following optimization problem

$$\underset{\{\boldsymbol{\Phi}_j\}}{\text{minimize}} \left\| \tilde{\boldsymbol{H}} - \sum_{k=1}^{K} \prod_{j=1}^{k} (\boldsymbol{\Phi}_j \odot \boldsymbol{S}) + \boldsymbol{\Phi}_0 \right\|, \tag{7}$$

where $\|\cdot\|$ is an appropriate distance measure, i.e., Frobenius norm ($\|\cdot\|_F$), spectral norm ($\|\cdot\|_2$), etc. As in practice entry-wise closeness between $\tilde{\boldsymbol{H}}$ and $\boldsymbol{H}_{\mathrm{ev}}$ is desired, in this work, we consider the Frobenius norm to measure the error of the approximation. Note that the Frobenius norm is also a natural upper bound for the spectral norm.

Unfortunately, (7) is a high-dimensional nonconvex problem that might lead to suboptimal results due to its multiple local minima. To address this issue, in this work, we propose a two-step approach for designing the weighting matrices $\{\boldsymbol{\Phi}_j\}_{j=0}^{K}$.

First, let us consider the following decomposition for approximating a given filter $\tilde{\boldsymbol{H}}$ by a finite matrix series:

$$\tilde{\boldsymbol{H}} \approx \sum_{k=1}^{K} \tilde{\boldsymbol{\Phi}}_k \odot \boldsymbol{S}^k + \tilde{\boldsymbol{\Phi}}_0, \tag{8}$$

where the matrix $\tilde{\boldsymbol{\Phi}}_k$ has the same support as $\boldsymbol{S}^k$ and $\tilde{\boldsymbol{\Phi}}_0$ is assumed diagonal. This decomposition perfectly describes any filter $\tilde{\boldsymbol{H}}$ for sufficiently large K, i.e., as large as the diameter of the graph, as then all the entries of $\boldsymbol{S}^K$ are nonzero. Although there is no unique decomposition for $\tilde{\boldsymbol{H}}$, the proposed decomposition in (8) allows us to approximate any filter as a set of matrices, $\{\boldsymbol{M}_k\}_{k=0}^{K}$, that can be decomposed through an element-wise operation, i.e., $\boldsymbol{M}_k = \tilde{\boldsymbol{\Phi}}_k \odot \boldsymbol{S}^k$. Using the definition of the Frobenius norm, we can find the weighting matrices $\{\tilde{\boldsymbol{\Phi}}_k\}_{k=0}^{K}$ from (8) by solving

$$\underset{\{\tilde{\boldsymbol{\phi}}_k\}}{\text{minimize}} \left\| \tilde{\boldsymbol{h}} - \sum_{k=0}^{K} \mathrm{diag}(\boldsymbol{s}_k)\tilde{\boldsymbol{\phi}}_k \right\|_2, \tag{9}$$

where $\tilde{\boldsymbol{h}} := \mathrm{vec}(\tilde{\boldsymbol{H}})$, $\tilde{\boldsymbol{\phi}}_k := \mathrm{vec}(\tilde{\boldsymbol{\Phi}}_k)$ and $\boldsymbol{s}_k := \mathrm{vec}(\boldsymbol{S}^k)$. As problem (9) is clearly a convex program, methods for its efficient solution are readily available. In most instances (9) is an underdetermined problem, hence a natural regularized solution for (9) could be given by the minimum norm solution. However, it is clear that any other regularization technique can be employed to obtain the matrices $\{\tilde{\boldsymbol{\Phi}}_k\}_{k=0}^{K}$.

After the weighting matrices $\{\tilde{\boldsymbol{\Phi}}_k\}$ are obtained, as a second step, we require to obtain the weighting matrices of the edge-variant filter in (5). To do so we propose to fit the matrices $\{\boldsymbol{\Phi}_k\}_{k=0}^{K}$ by solving sequentially the following problem:

$$\underset{\boldsymbol{\Phi}_k}{\text{minimize}} \left\| \tilde{\boldsymbol{\Phi}}_k \odot \boldsymbol{S}^k - (\boldsymbol{\Phi}_k \odot \boldsymbol{S}) \prod_{j=1}^{k-1} (\boldsymbol{\Phi}_j \odot \boldsymbol{S}) \right\|_F, \tag{10}$$

where it is assumed that the matrices $\{\mathbf{\Phi}_j\}_{j<k}$ have been already obtained, $\mathbf{\Phi}_0 = \tilde{\mathbf{\Phi}}_0$ and $\mathbf{\Phi}_1 = \tilde{\mathbf{\Phi}}_1$.

Even though the EV graph filter proposed in (5) is the most general version of linear graph filters that weight the entries of the shift operator, the two-step design of the weighting matrices might result in an involved and ill-conditioned task. This is due to the fact that: (i) the decomposition in (8), and (ii) the solution obtained in (9) are not optimal choices for a particular problem instance. Considering these facts, in the next section we propose a constrained version of the EV graph filter that provides a simpler design for the weighting matrices and still preserves the efficient implementation with complexity $\mathcal{O}(MK)$.

4. CONSTRAINED EDGE-VARIANT GRAPH FILTERS

Instead of considering the filter structure in (5), consider the following description for a ready-to-distribute constrained EV (C-EV) graph filter:

$$\boldsymbol{H}_{\text{c-ev}} \triangleq \sum_{k=1}^{K} (\mathbf{\Phi}_k \odot \boldsymbol{S})\boldsymbol{S}^{k-1} + \mathbf{\Phi}_0. \tag{11}$$

Similar to the previous case, the matrix $\mathbf{\Phi}_k \in \mathbb{R}^{N\times N}$ is an edge-weighting matrix whose support is equal to the one of $\boldsymbol{S}$. In this formulation, the matrix $\mathbf{\Phi}_0$ is assumed to be diagonal.

From the definition of the C-EV graph filter in (11), it can be observed that it allows a distributed implementation. More specifically, every node can track two values which can be obtained by local interactions: (i) the output of a regular shift $\boldsymbol{x}^{(k)} = \boldsymbol{S}\boldsymbol{x}^{(k-1)}$, and (ii) the output of a weighted shift $\boldsymbol{y}^{(k)} = (\mathbf{\Phi}_k \odot \boldsymbol{S})\boldsymbol{x}^{(k-1)}$, which can both be computed locally by appropriately combining neighboring data.

Following the same approach as in the previous section, we can design the weighting matrices $\{\mathbf{\Phi}_k\}_{k=0}^K$ through a least squares approach when the criterion in (7) is employed. That is, the optimal weighting matrices are the solution to the following optimization problem

$$\underset{\{\boldsymbol{\phi}_k\}}{\text{minimize}} \quad \left\| \tilde{\boldsymbol{h}} - \sum_{k=1}^{K} (\boldsymbol{S}^{k-1} \otimes \boldsymbol{I})\text{diag}(\boldsymbol{s})\boldsymbol{\phi}_k + \boldsymbol{\phi}_0 \right\|_2, \tag{12}$$

where $\boldsymbol{s}$ and $\boldsymbol{\phi}_k$ are defined as diag($\boldsymbol{S}$) and vec($\mathbf{\Phi}_k$), respectively. The vector $\tilde{\boldsymbol{h}} = \text{vec}(\tilde{\boldsymbol{H}})$ is the vectorized desired filter response matrix.

As the support of the weighting matrices is known, i.e., the locations of the nonzero entries are given by the nonzero entries of the shift operator, the filter (in its vectorized form) can be expressed as

$$\boldsymbol{h}_{\text{c-ev}} = \sum_{k=1}^{K} \tilde{\boldsymbol{S}}_k \tilde{\boldsymbol{\phi}}_k + \tilde{\boldsymbol{\phi}}_0, \tag{13}$$

where $\boldsymbol{h}_{\text{c-ev}} = \text{vec}(\boldsymbol{H}_{\text{c-ev}})$, $\tilde{\boldsymbol{\phi}}_k$ is the vector with the nonzero entries of $\boldsymbol{\phi}_k$, and $\tilde{\boldsymbol{S}}_k$ is the matrix $(\boldsymbol{S}^{k-1} \otimes \boldsymbol{I})\text{diag}(\boldsymbol{s})$ with the zero columns removed. That is, as the support for each $\mathbf{\Phi}_k$ is known, only its nonzero entries must be estimated.

Therefore, the optimal weights for the edges are given by the solution to the linear system

$$\tilde{\boldsymbol{h}} = [\boldsymbol{I} \quad \tilde{\boldsymbol{S}}_1 \quad \dots \quad \tilde{\boldsymbol{S}}_K] \begin{bmatrix} \tilde{\boldsymbol{\phi}}_0 \\ \tilde{\boldsymbol{\phi}}_1 \\ \vdots \\ \tilde{\boldsymbol{\phi}}_K \end{bmatrix} = \mathbf{\Psi}\boldsymbol{\theta}, \tag{14}$$

where $\mathbf{\Psi} \in \mathbb{R}^{N^2 \times \text{nnz(S)}\cdot\text{K+N}}$ and $\boldsymbol{\theta} \in \mathbb{R}^{\text{nnz(S)}\cdot\text{K+N}}$. Here, nnz($\cdot$) is the number of nonzero entries of a matrix. In addition, if a regularized solution is desired, a natural penalisation term might be the convex ℓ_1-norm which induces sparsity in the solution leading to a reduced number of active coefficients.

The linear system in (14) has a unique solution as long as the condition rank($\mathbf{\Psi}$) = nnz(S)·K+N holds. Otherwise, regularization should be used in order to obtain a unique solution for (14). We conclude this section with the following remark:

Remark 1. *It is worth noticing that the previously proposed NV graph filter [11] [cf.* (4)*] is a particular case of the C-EV graph filter, in which every row of the matrices* $\{\mathbf{\Phi}_k\}_{k=1}^K$ *has equal non-zero elements, i.e.,*

$$\boldsymbol{H}_{\text{nv}} = \sum_{k=1}^{K} \left((\boldsymbol{\phi}_k \mathbf{1}^T) \odot \boldsymbol{S}\right)\boldsymbol{S}^{k-1} + \text{diag}(\boldsymbol{\phi}_0). \tag{15}$$

In the sequel, we illustrate the performance of the proposed methods via numerical tests.

5. NUMERICAL RESULTS

In this section, we test the proposed C-EV graph filter for approximating a user-provided frequency response[2]. For these tests, we consider a random community graph generated with the GSP Toolbox [18], with $N = 256$ nodes and shift operator $\boldsymbol{S} = \boldsymbol{L}_{\text{n}}$, i.e., the normalized Laplacian. Further, the maximum order of the FIR graph filters that is used as baseline is $K_{\max} = 25$. For analysis, we first consider two frequency responses commonly used in the graph community: (i) an exponential kernel, i.e.,

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

and (ii) an ideal low pass filter, i.e.,

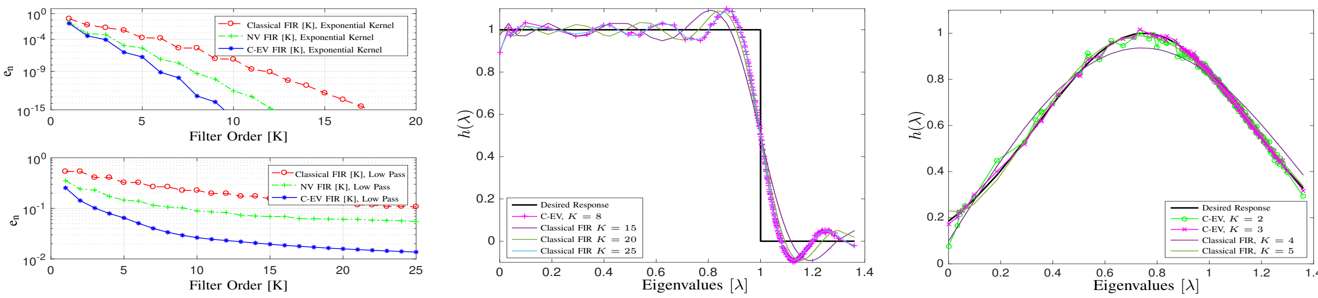
$$\tilde{h}(\lambda) = \begin{cases} 1 & 0 \le \lambda \le \lambda_{\text{c}} \\ 0 & \text{otherwise,} \end{cases}$$

with γ and μ being the spectrum decaying factor and the central parameter for the exponential kernel, respectively, and λ_{c} the cut-off frequency in the ideal low pass filter. These two ideal filters are considered as the desired filter responses.

Fig. 1(a) shows a comparison of the normalized approximation error in terms of the Frobenius norm between the fitted, $\boldsymbol{H}_{\text{fit}}$, and the desired, $\tilde{\boldsymbol{H}}$, frequency response, for both scenarios, i.e., $e_{\text{n}} = \|\tilde{\boldsymbol{H}} - \boldsymbol{H}_{\text{fit}}\|_F^2 / \|\tilde{\boldsymbol{H}}\|_F^2$. In the low pass scenario, we remark that the C-EV graph filter achieves the error floor for $K = 8$, while the NV graph filter for $K = 13$ and the *classical* FIR for $K = 17$. In the exponential kernel scenario we observe that the gap for a fixed NMSE between the constrained EV and NV graph filters is only two orders while the *classical* FIR requires much higher orders to meet the imposed NMSE target.

In Fig. 1(b) we depict the approximation accuracy of the C-EV graph filter ($K = 8$) with that of the *classical* FIRs of different orders for the low pass filter case. We note that the approximation quality for the C-EV graph filter is similar to that of the *classical* FIR with $K = 25$. This order reduction implies that less communication rounds in the graph are required to implement such a low pass filter. Similarly, the comparison between the responses of the different filters approximating an exponential kernel filter, is shown

[2]The code to reproduce the figures in this paper can be found in https://gitlab.com/fruzti/EdgeVariantGraphFilter



(a) (b) (c)

Fig. 1: (a) Error comparison between the proposed EV graph filter, the NV graph filter and the *classical* FIR for different orders. (Top) Performance in the ideal low pass scenario. (Bottom) Performance for the exponential kernel. (b) Comparison of the frequency response for a FIR filter of increasing order with the response of the C-EV filter of order $K = 8$ when approximating a perfect low pass filter. (c) Comparison of the frequency response for a FIR filter of increasing order with the response of the C-EV filter of different orders when approximating an exponential kernel with parameters $\mu = 0.75$ and $\gamma = 3$.

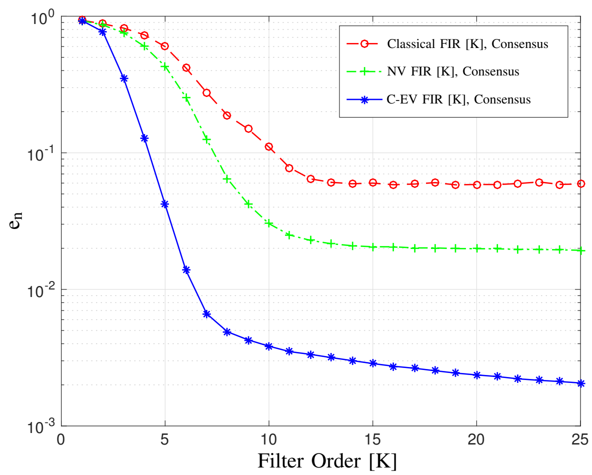


Fig. 2: Comparison approximation error of FIR filters for increasing order with the response of the C-EV filter when approximating a consensus operation, i.e., $\tilde{\boldsymbol{H}} = 1/N\mathbf{1}\mathbf{1}^T$.

in Fig. 1(c). In this case, the exponential kernel, with parameters $\{\mu = 0.75, \gamma = 3\}$, is well-approximated by the C-EV graph filter for $K = 3$. This approximation produces a similar result with a *classical* FIR of order $K = 5$, providing less communication rounds. Notice that the C-EV filter of order $K = 2$ outperforms all *classical* FIRs with $K \leq 4$.

Finally, we compare the performance of the FIR filters to implement distributed finite-time consensus. The latter is a particular case of the ideal low pass example with $\tilde{\boldsymbol{H}} = \frac{1}{N}\mathbf{1}\mathbf{1}^T$. Fig. 2 shows the normalized approximation error, e_n, of the three approaches where we note that the C-EV graph filter improves the approximation accuracy by up to one order w.r.t. the other alternatives.

It is worth noticing that the improved accuracy of the C-EV graph filter is due to its larger DoFs, i.e., $\mathrm{DoF}_{\mathrm{c-ev}} = \mathrm{nnz}(\mathbf{S}) \cdot \mathrm{K} + \mathrm{N} > \mathrm{N} \cdot (\mathrm{K}+1) = \mathrm{DoF}_{\mathrm{nv}}$ leading to a saving in terms of communication and computational complexity compared to a distributed implementation of FIR graph filters. However, due to this design freedom, we would like to remark that the filter design for the C-EV graph filters might suffer from numerical issues for large filter orders due to the conditioning of the matrix $\boldsymbol{\Psi}$, i.e., the number of parameters to be estimated $(\mathrm{nnz}(\mathrm{S}) \cdot \mathrm{K} + \mathrm{N})$ becomes larger than the effective rank of $\boldsymbol{\Psi}$. Similar problems might arise for NV filters of high order, although the number of parameters to estimate are smaller compared to the C-EV graph filter.

6. CONCLUSION

In this work, a generalization of the finite impulse response graph filters, named the edge-variant graph filter, has been proposed in order to reduce the communication and computational complexity related to a distributed filter implementation. By designing edge-weighting matrices, it is possible to locally weight, possibly in an asymmetric way, the information acquired from each node's neighbors giving higher/lower weight to the most/less informative nodes. Despite that the more general edge-variant graph filter may encounter numerical challenges in the filter design phase, we show that it is possible to obtain a constrained version of it that still generalizes the state of the art and reduces the communication and computational complexity while ensuring the same approximation accuracy.

7. REFERENCES

[1] 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 Signal Processing Magazine*, vol. 30, no. 3, pp. 83–98, 2013.

[2] D. I. Shuman, P. Vandergheynst, and P. Frossard, "Chebyshev Polynomial Approximation for Distributed Signal Processing," in *IEEE International Conference on Distributed Computing in Sensor Systems and Workshops (DCOSS)*, 2011, pp. 1–8.

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

[4] S. Chen, A. Sandryhaila, J. M. Moura, and J. Kovacevic, "Signal Denoising on Graphs via Graph Filtering," in *IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, Atlanta, GA, USA, December 2014, pp. 872–876.

[5] A. Sandryhaila and J. M. Moura, "Discrete Signal Processing on Graphs: Frequency Analysis," *IEEE Transactions on Signal Processing*, vol. 62, no. 12, pp. 3042–3054, 2014.

[6] S. K. Narang, A. Gadde, and A. Ortega, “Signal processing techniques for interpolation in graph structured data,” in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE, 2013, pp. 5445–5449.

[7] F. Zhang and E. R. Hancock, “Graph Spectral Image Smoothing Using the Heat Kernel,” *Pattern Recognition*, vol. 41, no. 11, pp. 3328–3342, 2008.

[8] J. Ma, W. Huang, S. Segarra, and A. Ribeiro, “Diffusion filtering of graph signals and its use in recommendation systems,” in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2016, pp. 4563–4567.

[9] M. Belkin and P. Niyogi, “Laplacian eigenmaps and spectral techniques for embedding and clustering,” in *NIPS*, vol. 14, no. 14, 2001, pp. 585–591.

[10] N. Tremblay, G. Puy, R. Gribonval, and P. Vandergheynst, “Compressive spectral clustering,” in *Machine Learning, Proceedings of the Thirty-third International Conference (ICML 2016), June*, 2016, pp. 20–22.

[11] S. Segarra, A. Marques, and A. Ribeiro, “Optimal graph-filter design and applications to distributed linear network operators,” *IEEE Transactions on Signal Processing*, vol. 65, no. 15, pp. 4117–4131, August 2017.

[12] A. Loukas, A. Simonetto, and G. Leus, “Distributed Autoregressive Moving Average Graph Filters,” *Signal Processing Letters*, vol. 22, no. 11, pp. 1931–1935, 2015.

[13] E. Isufi, A. Loukas, A. Simonetto, and G. Leus, “Autoregressive moving average graph filtering,” *IEEE Transactions on Signal Processing*, vol. 65, no. 2, pp. 274–288, January 2017.

[14] E. Isufi, A. Loukas, and G. Leus, “Autoregressive moving average graph filters: A stable distributed implementation,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, March 2017, pp. 4119–4123.

[15] J. Liu, E. Isufi, and G. Leus, “Autoregressive Moving Average Graph Filter Design,” in *IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, November 2017.

[16] E. Isufi, A. Loukas, A. Simonetto, and G. Leus, “Filtering random graph processes over random time-varying graphs,” *IEEE Transactions on Signal Processing*, vol. 65, no. 16, pp. 4406–4421, August 2017.

[17] E. Isufi and G. Leus, “Distributed Sparsified Graph Filters for Denoising and Diffusion Tasks,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, March 2017, pp. 5865–5869.

[18] N. Perraudin, J. Paratte, D. Shuman, L. Martin, V. Kalofolias, P. Vandergheynst, and D. K. Hammond, “GSPBOX: A toolbox for signal processing on graphs,” *ArXiv e-prints*, Aug. 2014.