Rational Chebyshev Graph Filters

Rimleanscaia, Oxana; Isufi, Elvin

DOI
10.1109/IEEECONF51394.2020.9443317

Publication date
2020

Document Version
Accepted author manuscript

Published in
Conference Record of the 54th Asilomar Conference on Signals, Systems and Computers, ACSSC 2020

RATIONAL CHEBYSHEV GRAPH FILTERS

Oxana Rimleanscaia† *and Elvin Isufi*‡

† Engineering Department, University of Perugia, Italy
‡Intelligent Systems Department, Delft University of Technology, The Netherlands
E-mails: oxana.rimleanscaia@studenti.unipg.it; e.isufi-1@tudelft.nl

ABSTRACT

This paper proposes rational Chebyshev graph filters to approximate step graph spectral responses with arbitrary precision, which are of interest in graph filter banks and spectral clustering. The proposed method relies on the well-known Chebyshev filters of the first kind and on a domain transform of the angular frequencies to the graph frequencies. This approach identifies in closed-form the filter coefficients, hence it avoids the costs of solving a nonlinear problem. Rational Chebyshev graph filters improve the control on the ripples in the pass- and stop-band and on the transition decay. Numerical experiments show the proposed approach approximates better ideal step responses than competing alternatives and reaches the performance of the ideal filters in compressive spectral clustering.

***Index Terms*—** Graph filters, graph signal processing, spectral clustering.

1. INTRODUCTION

Graph filters are mathematical operators employed to process and learn from network data [1,2]. They serve as the main building block for graph wavelets [3], graph neural networks [4], and distributed processing over networks [5,6]. Graph filters are also an integral part of graph filter banks [7–10] and spectral clustering [11, 12]. Their primary role is to shape the signal's spectrum while retaining a linear complexity. This is accomplished with two main implementations: the finite impulse response (FIR) form [13, 14], which implements a polynomial frequency response; and the autoregressive moving average (ARMA) [15] form, which implements a rational frequency response.

Filters implementing rational responses are of interest when we need to approximate accurately an ideal step function with a narrow transition band [8, 12]. While FIR filters are also a valid option, they suffer capturing the abrupt transition between the pass- and stop-band. Rational filters capture these transitions better at expenses of inverting a linear problem with iterative solvers [16, 17]. However, designing rational filters is challenging because of their nonlinear structure. The works in [15, 17, 18] propose Prony-inspired design strategies to get the filter coefficients. These approaches apply to any frequency response but require solving an iterative least-squares problem in the design phase. The latter translates into a cost overhead and several combinations of filter orders should be tried to determine the right orders. The same issues hold also for the approach in [19], which rephrases the filter design as a sum-of-squares optimization problem. Contrarily, the work in [16] proposes a Butterworth-like design to find the filter coefficients in closed-form if the desired response is an ideal step. Despite avoiding the design costs and the flat pass-band response, Butterworth filters suffer also in narrow transition bands. Hence, if the cutoff frequency is close to a graph frequency, as happens in graph filter banks and compressive spectral clustering (CSC), the filter performance reduces to that of FIR filters.

To address the above limitations, we propose a design strategy for rational graph filters based on the principles of Chebyshev filters of the first kind [20]. We bring the latter design to the graph filter setting through a variable transformation between the angular frequency interval and the graph frequency interval in a form akin to Chebyshev FIR design [14]. Rational Chebyshev graph filters have the following features of interest. Compared with current rational designs, they identify the coefficients in closed-form with controlled ripples and width of the transition band. The proposed design identifies the smallest order to meet the desired requirements; hence, bypassing the search over different orders as done by state-of-the-art methods. Compared with FIR graph filters, rational Chebyshev filters have a sharper transition. Numerical comparisons show the proposed design approximates an ideal low pass up to 6dB better that competing alternatives, and it improves the CSC at the point of meeting the ideal filter performance.

The rest of the paper is organized as follows. Section 2 contains the preliminary material and formulates our problem. Section 3 discusses the Chebyshev graph filters of the first kind to design low-pass responses. Section 4 focuses on designing different frequency responses via graph frequency transformations. Section 5 contains the numerical results and Section 6 the paper conclusions.

2. PROBLEM FORMULATION

Basics of GSP. We consider an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ comprising a set of N nodes $\mathcal{V} = \{1, \ldots, N\}$ and a set of M edges $\mathcal{E}$. The structure of this graph is captured by the adjacency matrix $\boldsymbol{A}$ such that $A_{ij} > 0$ if there is an edge connecting nodes $\{i, j\}$ and $A_{ij} = 0$ otherwise. Undirected graphs are also represented by the Laplacian matrix $\boldsymbol{L}$, which we use as a generic variable to denote both the discrete and the normalized Laplacian. Since $\boldsymbol{L}$ is symmetric and positive semi-definite, it can always be decomposed as $\boldsymbol{L} = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^{\mathsf{H}}$ where $\boldsymbol{U}$ is the matrix containing the eigenvectors of $\boldsymbol{L}$ and $\boldsymbol{\Lambda} = \text{diag}(\lambda_0, \ldots, \lambda_{N-1})$ is the diagonal matrix containing the eigenvalues $0 = \lambda_0 \leq \lambda_1 \leq \ldots \leq \lambda_{N-1}$ on the main diagonal.

Along with the graph, we are interested in processing graph signals, which are defined as a mapping from the vertex set to the set of real numbers $x : \mathcal{V} \to \mathbb{R}$. We represent the graph signal in the vector form $\boldsymbol{x} = [x_1, \ldots, x_N]^\top$ where entry x_i is the signal value at node i. By exploiting the Laplacian eigendecomposition, we can project the signal as $\hat{\boldsymbol{x}} = \boldsymbol{U}^{\mathsf{H}}\boldsymbol{x}$ which is known as the graph Fourier transform (GFT). Vector $\hat{\boldsymbol{x}}$ contains the signal Fourier coefficients and the inverse transform is denoted by $\boldsymbol{x} = \boldsymbol{U}\hat{\boldsymbol{x}}$. The eigenvalues of $\boldsymbol{L}$ are referred to as the graph frequencies; smaller valued eigenvalues represent lower frequencies since the corresponding eigenvectors vary less over the graph [1]. The GFT allows processing signal $\boldsymbol{x}$ in the spectral domain by altering its Fourier coefficients $\hat{\boldsymbol{x}}$. The tool for spectral processing is the graph filter, which is defined in the spectral domain by the diagonal matrix $h(\boldsymbol{\Lambda}) = \text{diag}(h(\lambda_0), \dots, h(\lambda_{N-1}))$ in which the ith diagonal entry $h(\lambda_i)$ denotes the filter response at frequency λ_i. The filtered signal has the GFT $\hat{\boldsymbol{y}} = h(\boldsymbol{\Lambda})\hat{\boldsymbol{x}}$, which by the inverse GFT becomes $\boldsymbol{y} = \boldsymbol{U}h(\boldsymbol{\Lambda})\boldsymbol{U}^{\mathsf{H}}\boldsymbol{x} := \boldsymbol{H}(\boldsymbol{L})\boldsymbol{x}$ with graph filtering matrix $\boldsymbol{H}(\boldsymbol{L}) = \boldsymbol{U}h(\boldsymbol{\Lambda})\boldsymbol{U}^{\mathsf{H}}$.

Rational filters. While we can define $h(\boldsymbol{\Lambda})$ as the desired shape –an ideal low/band/high-pass response for spectral clustering [11, 12] or filter banks [8, 9, 21]– the latter requires computing the eigendecomposition of $\boldsymbol{L}$ which has a cost of order $\mathcal{O}(N^3)$. To reduce this cost, we can specify an analytic desired response $h^*(\lambda)$ in the graph frequency variable λ and approximate $h^*(\lambda)$ by a graph filter that can be implemented with a lower cost. One way of doing so is to consider graph filters with the rational transfer function

$$h(\lambda) = \frac{q(\lambda)}{p(\lambda)} = \frac{\sum_{q=0}^{Q} b_q \lambda^q}{1 + \sum_{p=1}^{P} a_p \lambda^p} \tag{1}$$

where scalars $\{a_p\}$ and $\{b_q\}$ are the filter coefficients and (P, Q) the filter orders. Because of their rational transfer function, filters in the form (1) can approximate the desired response $h^*(\lambda)$ better than FIR filters (the numerator of (1)) [17].

By exploiting the inverse GFT and substituting variable λ with the Laplacian variable $\boldsymbol{L}$, we can write the filter output in the vertex domain as

$$\left(\boldsymbol{I} + \sum_{p=1}^{P} a_p \boldsymbol{L}^p\right)\boldsymbol{y} = \left(\sum_{q=0}^{Q} b_q \boldsymbol{L}^q\right)\boldsymbol{x} \;\Rightarrow\; \boldsymbol{P}(\boldsymbol{L})\boldsymbol{y} = \boldsymbol{Q}(\boldsymbol{L})\boldsymbol{x} \tag{2}$$

where $\boldsymbol{P}(\boldsymbol{L}) := \boldsymbol{I} + \sum_{p=1}^{P} a_p \boldsymbol{L}^p$ and $\boldsymbol{Q}(\boldsymbol{L}) := \sum_{q=0}^{Q} b_q \boldsymbol{L}^q$ are FIR filters of orders P and Q, respectively. To get the output $\boldsymbol{y}$ we need to solve the linear system in (2). We can do the latter via iterative methods such as gradient descent [16] or conjugate gradients [17] with a computational cost of order $\mathcal{O}(M(PT + Q))$, and where T is the number of iterations of the descent algorithm. Notice that while we can solve (2) in its direct form, better numerical properties and faster convergence can be achieved if we rephrase (2) in the cascade or in the parallel form [15, 16]. In the numerical results, we shall use the parallel form because of its faster convergence.

When approximating the desired response $h^*(\lambda)$, the coefficients of the rational filters are obtained by minimizing the error

$$e(\lambda) = h^*(\lambda) - \frac{\sum_{q=0}^{Q} b_q \lambda^q}{1 + \sum_{p=1}^{P} a_p \lambda^p} \tag{3}$$

over a grid in the graph frequency interval $[0, \lambda_{\text{max}}]$. Since (3) is nonlinear in the denominator coefficients a_p, Prony's non-equivalent versions have been used as a proxy [17]. However, the latter lead only to a coarse approximation of $h^*(\lambda)$ which can be improved by iterative design methods at expenses of complexity (i.e., solving a sequence of least-squares problems). A search for different order tuples (P, Q) needs also to be done to meet the desired approximation error in the pass- and stop-band. The cost of the least square design and order search is also present when approximating $h^*(\lambda)$ with a FIR filter (setting $P = 0$ in (3)) [22]. Instead, Chebyshev FIR filters and Butterworth rational filters [16] avoid the design costs since they obtain the filter coefficients in closed-form [14].

Paper objective. The goal of this paper is to propose a closed-form design for rational graph filters [cf. (1)] when the desired response is an ideal step function such as that encountered in spectral clustering and graph filter banks. We do so by leveraging the analogy between rational graph filters and Chebyshev filters of the first kind (Ch_1) [20]. The proposed design relies on a change of variable to map the angular frequency interval $[0, \pi]$, where conventional Ch_1 filters are defined, to the graph frequency interval $[0, \lambda_{\text{max}}]$, where graph filters are defined, akin to the variable transformation used for Chebyshev FIRs [14].

3. CHEBYSHEV GRAPH FILTER OF THE FIRST KIND

In this section, we detail the design of low pass Chebyshev graph filters of the first kind. The latter will serve as the basis for designing other step filters in the next section.

Recall a Chebyshev polynomial of degree K in the variable x is defined as

$$T_K(x) = \begin{cases} \cos\big(K\text{arccos}(x)\big) & |x| \le 1 \\ \cosh(K\text{arccosh}(x)) & |x| > 1 \end{cases}. \tag{4}$$

Chebyshev polynomials can also be obtained via the recursion $T_K(x) = 2xT_{K-1}(x) - T_{K-2}(x)$ with initialization $T_0(x) = 1$ and $T_1(x) = x$. Denote then by ω the continuous angular frequency and recall an order K low-pass Chebyshev filter of the first kind is defined by its square-magnitude gain response

$$|h(\omega)|^2 = \frac{1}{1 + \varepsilon^2 T_K^2\left(\omega/\omega_{\text{c}}\right)} \tag{5}$$

where ε is the ripple factor, ω_{c} the angular cutoff frequency, and the filter specifications are given in the interval $[0, \pi]$ due to symmetry [20]. By changing the domain of the angular frequencies $\omega \in [0, \pi]$ to graph frequencies $\lambda \in [0, \lambda_{\text{max}}]$ via the transformation of variable $\omega = \pi\lambda/\lambda_{\text{max}}$, we can represent the rational Ch_1 in the graph frequency variable λ as

$$h_{\text{Ch}_1}(\lambda) = \frac{1}{1 + \varepsilon^2 T_K^2\left(\lambda/\lambda_{\text{c}}\right)} \tag{6}$$

where $\lambda_{\text{c}} = \lambda_{\text{max}}\omega_{\text{c}}/\pi$ transforms the cutoff frequency[1]. Subsequently, we can decompose the transfer function (6) as $h_{\text{Ch}_1}(\lambda) = h'(s)h'(-s)$ to identify the poles needed for the design. The rational function $h'(s)$ contains k poles $\{s_k\}$ that allow factorizing $h'(s)$ as

$$h'(s) = h_0 \prod_{k=0}^{K-1} \frac{-s_k}{s - s_k} \tag{7}$$

with constant

$$h_0 = \begin{cases} (1 + \varepsilon^2)^{-1/2} & K \text{ even} \\ 1 & K \text{ odd} \end{cases}. \tag{8}$$

The poles can be obtained in closed-form as

$$\begin{aligned} s_k = &-\lambda_{\text{c}}\text{sinh}\Big(\frac{1}{K}\text{arcsinh}\frac{1}{\varepsilon}\Big)\sin\left[\frac{(2k+1)\pi}{2K}\right] \\ &+ j\lambda_{\text{c}}\text{cosh}\Big(\frac{1}{K}\text{arcsinh}\frac{1}{\varepsilon}\Big)\cos\left[\frac{(2k+1)\pi}{2K}\right] \end{aligned} \tag{9}$$

for $k = 0, \dots, K-1$. By exploiting the relationship between variables s and λ, i.e., $s = \jmath\lambda$ with $\jmath = \sqrt{-1}$, we have the rational Chebyshev graph filter of the first kind

$$h_{\text{Ch}_1}(\lambda) = h'(j\lambda)h'(-j\lambda). \tag{10}$$

[1] Notice that if we were to define $h_{\text{Ch}_1}(\lambda)$ w.r.t. $h(\omega)$ and not its squared gain, we would have a square root on the right-hand side of (6). The latter would imply computing the square root of a filtering matrix prior to inversion. Further, defining the graph filter w.r.t. the squared gain allows a direct use of the Ch_1 design strategy in Algorithm 1.

Algorithm 1 Low-pass Ch_1 graph filter design.

1: Set the filter specifications: λ_p, λ_s, δ_p, δ_s;
2: Compute the discrimination factor δ

$$\delta = \left[\frac{(1-\delta_\mathrm{p})^{-2}-1}{\delta_\mathrm{s}^{-2}-1}\right]^{1/2};$$

3: Compute the selectivity factor $\sigma = \lambda_\mathrm{p}/\lambda_\mathrm{s}$;
4: Set the filter order K as

$$K = \left\lceil \frac{\operatorname{arccosh}(1/\delta)}{\operatorname{arccosh}(1/\sigma)} \right\rceil;$$

5: Set $\lambda_\mathrm{c} = \lambda_\mathrm{p}$ and $\varepsilon = \left[(1-\delta_\mathrm{p})^{-2}-1\right]^{1/2}$;
6: Compute the poles s_k from (9);
7: Compute the gain h_0 from (8);
8: Compute $h'(s)$ and $h'(-s)$ from (7);
9: Compute the graph filter response $h_{\mathrm{Ch}_1}(\lambda)$ from (10);

Design. Given the equivalence between the rational Ch_1 graph filter (10) and the squared gain of the conventional Ch_1 filter (5), we can obtain the coefficients defining $h_{\mathrm{Ch}_1}(\lambda)$ in closed-form following Algorithm 1. The user provides the pass-band frequency λ_p, the stop-band frequency λ_s, and the ripple bounds δ_p and δ_s. Step 4 identifies the minimum order K that satisfies the imposed requirements. This order is used to find the K complex conjugate poles [cf. (9)] and the graph filter response as per (10).

Differently from Prony's method [17], the rational Chebyshev design in Algorithm 1 has controlled ripples in the pass- and stop-band and a closed-form set of coefficients. Prony's method is however more versatile and not limited to sharp desired responses. The difference in terms of reduced design complexity holds also when comparing the rational Ch_1 filter with the least-square design of FIR filter [22]. Contrarily, the Chebyshev polynomial design [14] provides also closed-form coefficients, but it does not have controlled ripples and suffers from the Gibbs phenomenon. The rational Butterworth design [16] avoids the Gibbs phenomenon by forcing a flat response, but it pays the latter in transition band sharpness. Consequently, a higher order Butterworth filter is required, which incurs a higher computational cost. The rational Ch_1 filter does not suffer from the latter and keeps the filter order and cost low.

As it follows from (5), we define the design requirements of the conventional Ch_1 w.r.t. the squared magnitude response. These requirements can be achieved by implementing the transfer function $h(\omega)$ [20]. In other words, this implies that in conventional Ch_1 filters we work with squared Chebyshev polynomials $T_K^2(\omega/\omega_\mathrm{c})$ during design but implement only the transfer function $h(\omega)$ (i.e., the square root of (5)). Contrarily, (6) and (10) show that in graph filters we need to implement directly a filter having squared Chebyshev polynomials $T_K^2(\omega/\omega_\mathrm{c})$. This is a critical aspect to get as output the filtering operation designed by Algorithm 1. In turn, this implies the effective rational order is $2K$, therefore, it doubles the computational effort. The latter can be reduced by relaxing the requirements (hence K) during design.

4. GRAPH FREQUENCY TRANSFORMATIONS

Given the low-pass design in Algorithm 1, we can transfer this approach to designing band-pass and high-pass filters through transforming graph frequency variables. Without loss of generality, we detail here the design of high-pass filters.

Algorithm 2 High-pass Ch_1 graph filter design.

1: Set the filter specifications: $\tilde{\lambda}_\mathrm{p}$, $\tilde{\lambda}_\mathrm{s}$, $\tilde{\delta}_\mathrm{p}$, $\tilde{\delta}_\mathrm{s}$;
2: Design a low-pass graph filter using Algorithm 1 with $\lambda_\mathrm{p} = 1$, $\lambda_\mathrm{s} = \tilde{\lambda}_\mathrm{p}/\tilde{\lambda}_\mathrm{s}$ and $\delta_\mathrm{p} = \tilde{\delta}_\mathrm{p}$, $\delta_\mathrm{s} = \tilde{\delta}_\mathrm{s}$;
3: Obtain the high-pass graph filter frequency response $h_{\mathrm{Ch}_1}(\tilde{\lambda})$ as

$$\begin{aligned} h_{\mathrm{Ch}_1}(\tilde{\lambda}) &= h'(s)h'(-s) \text{ for } s = \tilde{\lambda}_\mathrm{p}/j\tilde{\lambda} \\ &= h_0^2 \left(\prod_{k=0}^{K-1} \frac{-js_k\tilde{\lambda}}{\tilde{\lambda}_\mathrm{p} - js_k\tilde{\lambda}}\right)\left(\prod_{k=0}^{K-1} \frac{js_k\tilde{\lambda}}{\tilde{\lambda}_\mathrm{p} + js_k\tilde{\lambda}}\right) \end{aligned}$$

with constant h_0 in (8).

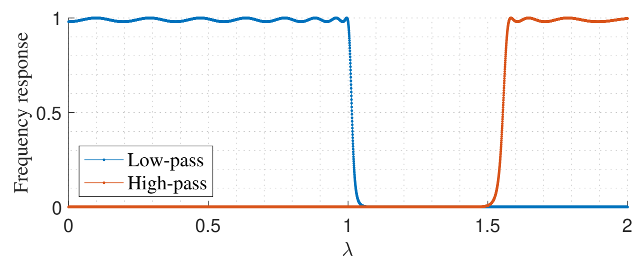

Fig. 1: Illustration of a high-pass graph filter obtained by transforming the low-pass counterpart via Algorithm 2. The high-pass filter requirements are $\tilde{\delta}_\mathrm{p} = \tilde{\delta}_\mathrm{s} = 0.01$; $\tilde{\lambda}_\mathrm{p} = 1.575$, $\tilde{\lambda}_\mathrm{s} = 1.425$.

Denote by $\tilde{\lambda}_\mathrm{p}$ the pass graph frequency of the high-pass filter and by $\tilde{\lambda}_\mathrm{s}$ the stop graph frequency. Let also $\tilde{\delta}_\mathrm{p}$ and $\tilde{\delta}_\mathrm{s}$ be the ripple requirements of the high-pass filter in the pass- and stop-band, respectively. With the details in Algorithm 2, we first design a low-pass filter [cf. Alg 1] with the transformed requirements in step 2. Then, denoting by $\tilde{\lambda}$ the graph frequency variable for the high-pass filter and transforming the graph frequency variables as

$$\lambda = \tilde{\lambda}_\mathrm{p}/j\tilde{\lambda} \tag{11}$$

we can project the designed low-pass frequency response $h(\lambda)$ in step 2 onto the high-pass frequency response $h(\tilde{\lambda})$ that satisfies the requirements in step 1. We illustrate an example in Fig.1. This straightforward variable transformation comes however with its price. The high-pass filter will have now polynomials in both the numerator and denominator parts. Hence, a high-pass rational Chebyshev filter requires solving a linear system of the form $\boldsymbol{P}(\boldsymbol{L})\boldsymbol{y} = \boldsymbol{Q}(\boldsymbol{L})\boldsymbol{x}$, while the low-pass Ch_1 has $\boldsymbol{Q} = \boldsymbol{I}$. Therefore, the computational cost of the high-pass filter is higher since we need to perform the FIR prefiltering $\boldsymbol{Q}(\boldsymbol{L})\boldsymbol{x}$.

5. NUMERICAL RESULTS

This section evaluates the rational Chebyshev graph filters by means of numerical results in approximating an ideal low-pass and performing compressive spectral clustering [12]. The proposed design method is compared with competing alternatives that provide the filter coefficients in closed-form, namely the Chebyshev FIR [14], the Jackson-Chebyshev FIR [12], and the rational Butterworth graph filter [16]. In these numerical results, we made use of the GSPBOX [23].

Approximation accuracy. First, we evaluate the design strategy in Algorithm 1 to approximate an ideal low-pass filter in the interval $[0, 2]$ with cut-off $\lambda_\mathrm{c} = 1$. For the rational Chebyshev filter, we set

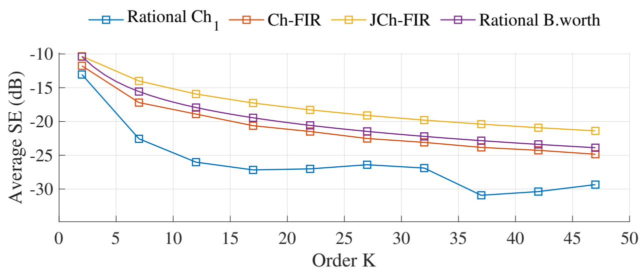

Fig. 2: Average SE per graph frequency as a function of the filter order $K \in [2, 48]$ for the rational Ch_1 design [cf. Alg. 1], the Chebyshev FIR filter [14], Jackson-Chebyshev FIR filter [12], and the rational Butterworth filter [16].

$\lambda_{\mathrm{p}} = 0.95$, $\lambda_{\mathrm{s}} = 1.05$ and evaluate different orders K (the value of ε [step 5, Alg. 1] has been grid searched for each order).

In Fig. 2, we show the average squared error (SE) in dB over the 500 graph frequency grid points of the interval $[0, 2]$ as a function of order K. The rational Ch_1 design outperforms the FIR counterparts and shows substantial benefits especially for low orders K. This result corroborates the well-known potential of rational Chebyshev functions in approximating sharp responses. Following our observation in Section 3 that the effective order of the rational Ch_1 filter is $2K$, we remark the rational Ch_1 still shows a lower SE for $K \geq 5$; i.e., the rational Ch_1 of order $K = 5$ outperforms other alternatives with orders much higher than ten. The main component contributing to this performance gap is in the error in the transition band where rational Ch_1 filters decay substantially faster than the other alternatives. The performance of the Butterworth and JCh-FIR is worse than the Chebyshev FIR because they force a flat response in pass-band at expenses of a lower decay in transition band.

Compressive Spectral Clustering (CSC). We now use the rational Ch_1 filters to substitute the Jackson-Chebyshev FIR in CSC [12]. Since we can control the sharpness in the transition band, our rationale is that Ch_1 filters will improve the CSC performance. The main steps of CSC are summarized in Algorithm 3. Graph filters are used in step 5 ($\boldsymbol{H}_k(\boldsymbol{L})$) to filter random signals and in step 8 ($\boldsymbol{G}(\boldsymbol{L})$) to interpolate the indicators from the low to the higher dimension. We focus at substituting $\boldsymbol{H}_k(\boldsymbol{L})$ with a Ch_1 rational filter[2].

We considered a stochastic block model (SBM) graph composed of $N = 1000$ nodes divided among $C = 10$ communities. We measured the ability of the different CSC methods to identify the communities of the SBM under different connectivity levels. The graph connectivity is dictated through a parameter ϵ measuring the ratio between inter- and intra-cluster edge formation probabilities. A larger ϵ represents a more difficult setting. We measured the clustering performance through the adjusted rand similarity index (ARI) between the SBM ground truth and the obtained communities; higher values indicate better performance [24]. The results are approximated over 15 different graph realizations.

Fig. 3 shows the ARI difference between the CSC methods using the rational Ch_1 implemented with the parallel form [15, 16] and the JCh-FIR. The order of the rational filter is $K = 11$ –identified with a transition bandwidth of 10% around each λ_k and ripple requirements of at most 0.05– while for the JCh-FIR the order is $K = 50$ [12]. We observe rational filters improve up to 4% the clustering performance

Algorithm 3 Compressive spectral clustering [12].

1: Set the number of clusters C and parameters $n = 2C\log(C)$, $d = 4\log(n)$, and $\gamma = 10^{-3}$;
2: Estimate eigenvalue λ_k of the Laplacian $\boldsymbol{L}$ via eigencount;
3: Generate a graph filter $\boldsymbol{H}_k(\boldsymbol{L})$ that approximates an ideal low-pass with cutoff frequency $\lambda_c = \lambda_k$;
4: Generate d zero-mean random Gaussian graph signals with variance d^{-1} : $\boldsymbol{R} = [\boldsymbol{r}_1, \boldsymbol{r}_2, ..., \boldsymbol{r}_d]^\top \in \mathbb{R}^{Nxd}$;
5: Filter all signals in $\boldsymbol{R}$ with $\boldsymbol{H}_k(\boldsymbol{L})$ and define the feature vector $\boldsymbol{f}_i \in \mathbb{R}^d$ for node i as
$$\tilde{\boldsymbol{f}}_i = \left[(\boldsymbol{H}_k(\boldsymbol{L})\boldsymbol{R})^\top \boldsymbol{\delta}_i\right] / \left\|_2 (\boldsymbol{H}_k(\boldsymbol{L})\boldsymbol{R})^\top \boldsymbol{\delta}_i\right\|_2$$
with $\boldsymbol{\delta}_i$ being a Dirac vector;
6: Generate a random binary sampling matrix $\boldsymbol{M} \in \mathbb{R}^{nxN}$ and keep only n feature vectors
$$[\tilde{\boldsymbol{f}}_1, \ldots, \tilde{\boldsymbol{f}}_n]^\top = \boldsymbol{M}[\boldsymbol{f}_1, \ldots, \boldsymbol{f}_N]^\top;$$
7: Run k-means on the reduced features with Euclidean distance
$$D^r_{ij} = \|\tilde{\boldsymbol{f}}_i - \tilde{\boldsymbol{f}}_j\|_2;$$
to obtain k (one per cluster) reduced indicator vectors $\boldsymbol{c}^r_j \in \mathbb{R}^n$;
8: Interpolate each reduced indicator vector $\boldsymbol{c}^r_j$ by solving the optimisation problem
$$\underset{\boldsymbol{x} \in \mathbb{R}^N}{\mathrm{argmin}} \, ||\boldsymbol{M}\boldsymbol{x} - \boldsymbol{c}^r_j||^2_2 + \gamma \boldsymbol{x}^\top \boldsymbol{G}(\boldsymbol{L})\boldsymbol{x}$$
to obtain the k indicator vectors $\tilde{\boldsymbol{c}}^\star_j \in \mathbb{R}^N$ for all nodes.

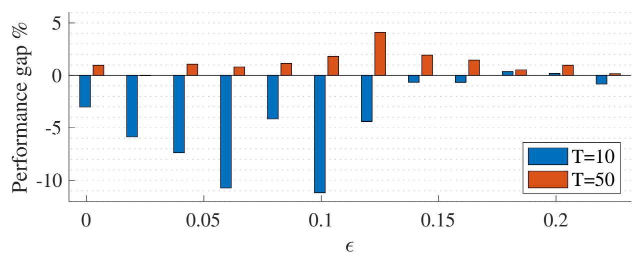

Fig. 3: ARI difference between the CSC with rational Chebyshev filters of order $K = 11$ and the CSC with polynomial Jackson-Chebyshev filters of order $K = 50$ [12]. The results are measured for different connectivity coefficient ϵ of the SBM graph and shown for different running iterations of the conjugate gradients.

if the conjugate gradient algorithm is run long enough to approximate the inverse. This implies that benefits in CSC can be achieved by a slight increase of computational complexity (still linear in the graph dimensions) induced by running the conjugate gradient in step 5 of Algorithm 3. The performance gap is larger for intermediate values of ϵ (around 0.1) which yields graphs with defined community structures but also with several inter-cluster links. In our opinion, values of ϵ around 0.1 are most indicative of the algorithmic performance since lower values of ϵ lead to well-defined and easy-to-identify clusters while larger values destroy the clustering structure of the graph.

Finally, in Fig. 4 we compare the recovery performance of the CSC using the rational Ch_1 with the conventional spectral clustering [11] and the CSC where filters $\boldsymbol{H}_k(\boldsymbol{L})$ and $\boldsymbol{G}(\boldsymbol{L})$ are the ideal filters defined in the Laplacian spectrum. We observe CSC with the proposed filter achieves the same result as using the ideal filter without needing the eigendecomposition. This yields two main insights. First, using the rational Ch_1 filter in CSC only in step 5 of Algorithm 3 achieves similar performance as using the optimal filter; hence, it suggests the importance of a sharper filter in step 5 compared

[2]In principle, rational graph filters can also substitute $\boldsymbol{G}(\boldsymbol{L})$ in step 8. We have observed the filter sharpness does not play an important role here as it does in step 5. Since using the Ch_1 in step 8 requires running two iterative algorithms (one for the Ch_1 filter output and one ofr solving the quadratic problem), we considered $\boldsymbol{G}(\boldsymbol{L})$ to be the JCh-FIR filter suggested in [12].

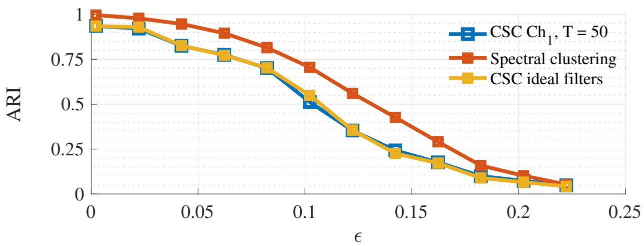

Fig. 4: Recovery performance of different clustering algorithms as a function of the SBM graph connectivity ϵ. The figure compares the conventional spectral clustering with the CSC using the proposed rational Ch_1 filter and with the CSC that uses ideal filters.

to step 8. Second, to meet the performance of the spectral clustering, an effort in CSC should be put in the sampling part of $k-$means.

6. CONCLUSION

This paper introduced a design method for rational graph filters based on Chebyshev filters of the first kind. The key to this design is based on a transformation of variables between angular frequencies and graph frequencies. Compared with state-of-the-art designs of rational graph filters, the proposed approach identifies the minimum filter order that meets the desired requirements and provides the filter coefficients in closed-form. The proposed rational Chebyshev graph filters are useful when the desired frequency response has an ideal step-function such as the ideal low-pass and high-pass. The latter are of interest in graph filter banks and compressive spectral clustering. Numerical results confirm the proposed design leads to sharper filters which improve the clustering performance by up to 4% without significantly affecting the computational cost. The transformation of variables used in this work for Chebyshev filters of the first kind can be directly used to design graph filters that follow the principles of Elliptic or Bessel filters, which performance evaluation is left for future work.

7. REFERENCES

[1] A. Ortega, P. Frossard, J. Kovacevic, J. M. Moura, and P. Vandergheynst, "Graph signal processing: Overview, challenges, and applications," *Proceedings of the IEEE*, vol. 106, no. 5, pp. 808–828, 2018.

[2] F. Gama, E. Isufi, G. Leus, and A. Ribeiro, "From graph filters to graph neural networks," *IEEE Signal Processing Magazine*, 2020.

[3] 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.

[4] E. Isufi, F. Gama, and A. Ribeiro, "Edgenets: edge varying graph neural networks," *arXiv preprint arXiv:2001.07620*, 2020.

[5] M. Coutino, E. Isufi, and G. Leus, "Advances in distributed graph filtering," *IEEE Transactions on Signal Processing*, vol. 67, no. 9, pp. 2320–2333, 2019.

[6] O. Teke and P. P. Vaidyanathan, "IIR filtering on graphs with random node-asynchronous updates," *IEEE Transactions on Signal Processing*, vol. 68, pp. 3945–3960, 2020.

[7] S. K. Narang and A. Ortega, "Perfect reconstruction two-channel wavelet filter banks for graph structured data," IEEE *Trans. Signal Process*, vol. 60, no. 6, pp. 2786–2799, 2012.

[8] 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.

[9] N. Tremblay and P. Borgnat, "Subgraph-based filterbanks for graph signals," *IEEE Transactions on Signal Processing*, vol. 64, no. 15, pp. 3827–3840, 2016.

[10] 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.

[11] A. Y. Ng, M. I. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," in *Advances in neural information processing systems*, 2002, pp. 849–856.

[12] N. Tremblay, G. Puy, R. Gribonval, and P. Vandergheynst, "Compressive spectral clustering," in *Int. Conf. on Machine Learning*, 2016, pp. 1002–1011.

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

[14] D. I. Shuman, P. Vandergheynst, D. Kressner, and P. Frossard, "Distributed signal processing via chebyshev polynomial approximation," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 4, no. 4, pp. 736–751, 2018.

[15] 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.

[16] X. Shi, H. Feng, M. Zhai, T. Yang, and B. Hu, "Infinite Impulse Response Graph Filters in Wireless Sensor Networks," *IEEE Signal Processing Letters*, Jan 2015.

[17] J. Liu, E. Isufi, and G. Leus, "Filter design for autoregressive moving average graph filters," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 5, no. 1, pp. 47–60, 2018.

[18] 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.

[19] T. Aittomäki and G. Leus, "Graph filter design using sum-of-squares representation," in *2019 27th European Signal Processing Conference (EUSIPCO)*. IEEE, 2019, pp. 1–5.

[20] J. G. Proakis and D. G. Manolakis, "Digital signal processing," 1996.

[21] D. B. Tay and J. Zhang, "Techniques for constructing biorthogonal bipartite graph filter banks," *IEEE Transactions on Signal Processing*, vol. 63, no. 21, pp. 5772–5783, 2015.

[22] S. Segarra, A. Marques, and A. Ribeiro, "Optimal graph-filter design and applications to distributed linear network operators," IEEE *Trans. Signal Process*, 2017.

[23] 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.

[24] L. Hubert and P. Arabie, "Comparing partitions," *Journal of classification*, vol. 2, no. 1, pp. 193–218, 1985.