Graph Filter Design Using Sum-of-squares Representation

*§Tuomas Aittomäki
*Department of Signal Processing and Acoustics,
Aalto University
PO Box 15400, FI-00076 Aalto, Finland

§Geert Leus
§Department of Microelectronics,
TU Delft,
Mekelweg 4, 2628 CD Delft, The Netherlands

***Abstract*—Graph filters are an essential part of signal processing on graphs enabling one to modify the spectral content of the graph signals. This paper proposes a graph filter optimization method with an exact control of the ripple on the passband and the stopband of the filter. The proposed filter design method is based on the sum-of-squares representation of positive polynomials. The optimization of both FIR and ARMA graph filters is convex with the proposed method.**

***Index Terms*—Graph filters, filter design, convex optimization**

I. Introduction

Graph signal processing is a field that extends classical discrete-time signal processing methods to the irregular domain of graphs [1]. Signals on graphs can arise in e.g. network analysis, biomedical signal processing, and machine learning [2]. A rudimentary signal processing method is filtering, and several studies have extended classical filters to the graph domain [3]–[9].

The typical task for a filter is to stop some of the signal components whereas letting others pass as unaltered as possible. The key parameter for the design is not only the cutoff, the point between the passband and stopband, but also the distortion allowed to the passed components and the residual power of the components that need to be attenuated. The former is controlled by the passband ripple, whereas the latter is determined by the stopband ripple.

There are various methods to obtain a graph filter with an approximation of the desired response, including least-squares fitting [4], [5] and Chebyshev polynomial fitting [10]. The drawback of these methods is that there is no control over the ripple on the passband or the stopband. In this paper, we propose a graph filter design method based on the sum-of-squares representation of polynomials. This formulation results in a convex optimization problem allowing one to control the ripple exactly for both finite impulse response (FIR) and autoregressive moving average (ARMA) filters. Furthermore, the least-squres methods can only minimize the so-called modified error of the ARMA filters, whereas the proposed method allows one to control the true error.

This paper is organized as follows. The use of the sum-of-squares representation of polynomials is briefly explained in Section II. The proposed design method for FIR filters is provided in Section III, while the ARMA filters are covered in Section IV. Section V discusses a discretization approximation to the proposed method. Numerical examples are provided in Section VI, and conclusions are given in Section VII.

II. Sum-of-squares Representation

In many optimization problems, we often encounter constraints of the type

$$p(x) \geq 0, \tag{1}$$

where p is a polynomial of a continuous variable x. The typical method to deal with this type of constraints is to discretize the domain of x, but then the solution will no longer be optimal. Moreover, discretization is not always possible (e.g. the domain is not bounded). Using the sum-of-squares representation, however, it is possible to handle this type of constraints without discretization.

The basic idea of the the sum-of-squares representation is to write $p(x)$ as a sum of squares of other polynomials, i.e.

$$p(x) = \sum_i q_i^2(x). \tag{2}$$

If this can be done, then necessarily $p(x) \geq 0$. It turns out that for univariate polynomials, a polynomial is non-negative if and only if it can be written as a sum of squares [11].

Requiring $p(x)$ to be a sum of squares can be further converted into linear matrix inequalities. The idea is the following. Suppose that the degree of $p(x)$ is $2K$. Collecting the monomials $1, x, x^2, \ldots, x^K$ into a vector $\mathbf{m}(x)$, it is possible to write

$$p(x) = \mathbf{m}^T(x)\mathbf{G}\mathbf{m}(x), \tag{3}$$

where the coefficients of $p(x)$ have been arranged into the $(K+1) \times (K+1)$ matrix $\mathbf{G}$. The (non-unique) matrix $\mathbf{G}$ is called the Gram matrix. If $\mathbf{G}$ is positive-semidefinite (PSD), then $p(x)$ expressed as the quadratic form in (3) is clearly non-negative for any x. Furthermore, $p(x)$ can be expressed as a sum of squares after applying the Cholesky factorization on $\mathbf{G}$.

It can in fact be shown that $p(x)$ is non-negative on the real axis if and only if there exists a PSD Gram matrix for the polynomial [11]. Satisfying the constraint (1) is thus equivalent to finding a positive-semidefinite matrix $\mathbf{G}$ such that (3) holds. In other words, defining $\mathbf{Y}_n$ to be an elementary Hankel matrix

978-9-0827-9703-9/19/$31.00 ©2019 IEEE

with ones on the nth anti-diagonal and zeroes elsewhere (the zeroth anti-diagonal being in the upper left corner), $p(x) \geq 0$ if and only if there exists a PSD matrix $\mathbf{G}$ such that

$$a_k = \text{trace}[\mathbf{Y}_k \mathbf{G}], k = 0, \ldots, 2K \tag{4}$$

where a_k is the kth degree coefficient of p [11].

Often the constraint $p(x) \geq 0$ does not need to hold for any x on the real axis, but only on some subset of it. The sum-of-squares representation can be used for this type of constraints as well.

Define $\mathcal{X}$ as the set on which polynomials $R_i(x)$ are non-negative, $i = 1, \ldots, L$, i.e.

$$\mathcal{X} = \{x \in \mathbb{R} | R_i(x) \geq 0, i = 1, \ldots, L\}. \tag{5}$$

The polynomial coefficients of R_i are denoted by $r_{i,k}$. The polynomial $p(x)$ is a sum of squares (non-negative) on $\mathcal{X}$ if and only if there are PSD matrices $\mathbf{G}_i$, $i = 0, \ldots, L$ such that

$$a_k = \text{trace}\left[\mathbf{Y}_k \mathbf{G}_0 + \sum_{i=1}^{L} \sum_{m+n=k} r_{i,m} \mathbf{Y}_n \mathbf{G}_i\right], n = 0, \ldots, 2K, \tag{6}$$

see [11] for details. With this formulation, the polynomial positivity constraints on intervals can be converted into PSD constraints.

III. FIR Filter Design

The graph FIR filter of the order K is given by

$$\mathbf{H} = \sum_{k=0}^{K} h_k \mathbf{L}^k, \tag{7}$$

where $\mathbf{L}$ is the Laplacian of the graph and h_k are the filter coefficients [4]. The output of the filter depends on the eigenvalues λ of $\mathbf{L}$. We wish to design the filter so as to approximate a desired response $H_d(\lambda)$. One approach to do this is to choose the filter coefficients to minimize an error criterion

$$\int_\lambda \Big| \sum_{k=0}^{K} h_k \lambda^k - H_d(\lambda) \Big|^2 w(\lambda) \mathrm{d}\lambda, \tag{8}$$

where $w(\lambda)$ is a weighting function. Denoting

$$\mathbf{h} = \begin{bmatrix} h_0 & h_1 & h_2 & \ldots & h_K \end{bmatrix}^T$$
$$\boldsymbol{\lambda} = \begin{bmatrix} 1 & \lambda & \lambda^2 & \ldots & \lambda^K \end{bmatrix}^T,$$

we have

$$\begin{aligned}
&\int_\lambda \Big| \sum_{k=0}^{K} h_k \lambda^k - H_d(\lambda) \Big|^2 w(\lambda) \mathrm{d}\lambda \\
=& \int_\lambda \Big| \mathbf{h}^T \boldsymbol{\lambda} - H_d(\lambda) \Big|^2 w(\lambda) \mathrm{d}\lambda \\
=& \int_\lambda (\mathbf{h}^T \boldsymbol{\lambda} \boldsymbol{\lambda}^T \mathbf{h} - 2 H_d(\lambda) \mathbf{h}^T \boldsymbol{\lambda} \mathrm{d}\lambda + H_d(\lambda)^2) w(\lambda) \mathrm{d}\lambda \\
=& \mathbf{h}^T \int_\lambda \boldsymbol{\lambda} \boldsymbol{\lambda}^T w(\lambda) \mathrm{d}\lambda \mathbf{h} - 2 \mathbf{h}^T \int_\lambda H_d(\lambda) \boldsymbol{\lambda} w(\lambda) \mathrm{d}\lambda \\
&+ \int_\lambda H_d(\lambda)^2 w(\lambda) \mathrm{d}\lambda \\
=& \mathbf{h}^T \mathbf{A}_0 \mathbf{h} - 2 \mathbf{c}^T \mathbf{h} + d.
\end{aligned} \tag{9}$$

If the desired response and the weighting function are piecewise constants or polynomials, the integrals can be done in a closed form. We now see that the coefficients minimizing the error are given by

$$\mathbf{h} = \mathbf{A}_0^{-1} \mathbf{c}. \tag{10}$$

It may be useful to add a regularization term $\gamma \|\mathbf{h}\|^2$ to the error criterion to penalize for coefficients with a large absolute value, as the noise power of the filter output is proportional to $\|\mathbf{h}\|^2$. In this case, the solution is simply

$$\mathbf{h} = (\mathbf{A}_0 + \gamma \mathbf{I})^{-1} \mathbf{c}. \tag{11}$$

If $\mathbf{A}_0$ is close to a singular matrix, the regularization will also help to stabilize the solution numerically.

The problem with minimizing only the integrated error is that there is no control over the peak error. We might also want to limit the maximum error, or limit the ripple on the passband and the stopband. Such peak constraints can be formulated as

$$\max_{\lambda \in \Lambda_i} \Big| \sum_{k=0}^{K} h_k \lambda^k - H_d(\lambda) \Big| \leq \epsilon_i, \tag{12}$$

where Λ_i is a particular domain, in this case an interval of λ i.e. the passband, the stopband, or the transition band. This type of constraints can be converted into positivity constraints of polynomials. More specifically, (12) is equal to

$$\Big| \sum_{k=0}^{K} h_k \lambda^k - H_d(\lambda) \Big| \leq \epsilon_i, \quad \forall \lambda \in \Lambda_i \tag{13}$$

from which one obtains two polynomial constraints

$$\sum_{k=0}^{K} h_k \lambda^k - (H_d(\lambda) - \epsilon_i) \geq 0, \quad \forall \lambda \in \Lambda_i \tag{14}$$

$$-\sum_{k=0}^{K} h_k \lambda^k + H_d(\lambda) + \epsilon_i \geq 0, \quad \forall \lambda \in \Lambda_i. \tag{15}$$

Using the sum-of-squares representation, such constraints can be formulated as linear matrix inequalities. Thus, we obtain a filter design problem of the form

$$\min_{\mathbf{h}} \ \mathbf{h}^T \mathbf{A}_0 \mathbf{h} - 2 \mathbf{c}^T \mathbf{h} + \gamma \|\mathbf{h}\|^2 \tag{16a}$$
$$\text{s.t.} \ \mathbf{h}^T \boldsymbol{\lambda} - (H_d(\lambda) - \epsilon_i) \geq 0, \quad \forall \lambda \in \Lambda_i \tag{16b}$$
$$-\mathbf{h}^T \boldsymbol{\lambda} + H_d(\lambda) + \epsilon_i \geq 0, \quad \forall \lambda \in \Lambda_i, \tag{16c}$$

which is convex.

IV. ARMA Filter Design

The response of a graph ARMA filter of order K is given by [5]

$$H(\lambda) = \frac{f(\lambda)}{g(\lambda)} = \frac{\sum_{k=0}^{K} b_k \lambda^k}{\sum_{k=0}^{K} a_k \lambda^k}. \tag{17}$$

The square error $|H(\lambda) - H_d(\lambda)|^2$ for the ARMA filter is not directly integrable, but one may integrate the so-called modified error

$$\int_\lambda \Big| \sum_{k=0}^{K} b_k \lambda^k - H_d(\lambda) \sum_{k=0}^{K} a_k \lambda^k \Big|^2 w(\lambda) \mathrm{d}\lambda \tag{18}$$

instead. Defining

$$\mathbf{A}_i = \int_\lambda H_d^i(\lambda)\boldsymbol{\lambda}\boldsymbol{\lambda}^T w(\lambda)\mathrm{d}\lambda \tag{19}$$

and using the same procedure as for the FIR case, the modified error can be written as

$$\mathbf{b}^T\mathbf{A}_0\mathbf{b} - 2\mathbf{b}^T\mathbf{A}_1\mathbf{a} + \mathbf{a}^T\mathbf{A}_2\mathbf{a} = \tilde{\mathbf{h}}^T\tilde{\mathbf{A}}\tilde{\mathbf{h}}, \tag{20}$$

where

$$\tilde{\mathbf{h}} = \begin{bmatrix}\mathbf{b}^T & \mathbf{a}^T\end{bmatrix}^T \tag{21}$$

and

$$\tilde{\mathbf{A}} = \begin{bmatrix}\mathbf{A}_0 & -\mathbf{A}_1 \\ -\mathbf{A}_1 & \mathbf{A}_2\end{bmatrix} \tag{22}$$

In the case H_d and w are either zero or one for any λ, we have $\mathbf{A}_1 = \mathbf{A}_2$.

In order to avoid the trivial solution, a constraint is needed for $\mathbf{a}$. A logical one is to choose $a_0 = 1$. Defining $\mathbf{u}_i$ as the unit vector with the ith element equal to one, the problem can be now written as

$$\min_{\tilde{\mathbf{h}}} \mathbf{h}^T\tilde{\mathbf{A}}\tilde{\mathbf{h}} + \gamma\|\tilde{\mathbf{h}}\|^2 \tag{23}$$

$$\text{s.t. } \tilde{\mathbf{h}}^T\mathbf{u}_{K+2} = 1 \tag{24}$$

with a well-known solution

$$\tilde{\mathbf{h}} = \frac{(\tilde{\mathbf{A}} + \gamma\mathbf{I})^{-1}\mathbf{u}_{K+2}}{\mathbf{u}_{K+2}^T(\tilde{\mathbf{A}} + \gamma\mathbf{I})^{-1}\mathbf{u}_{K+2}} \tag{25}$$

The regularization does not have a direct connection with the output noise power, but it stabilizes the solution numerically in this case as well.

As for the FIR case, this design has the problem that it does not control the peak values, which is especially important as the ARMA filter might have poles.

Although the exact error is not integrable, it is possible to constrain the exact peak values using the sum-of-squares representation with an additional constraint. The peak error constraint on the interval Λ_i is given by

$$\max_{\lambda\in\Lambda_i}\left|\frac{f(\lambda)}{g(\lambda)} - H_d(\lambda)\right| \le \epsilon_i. \tag{26}$$

With the additional constraint that $g(\lambda) \ge 0$, one obtains

$$-\epsilon_i g(\lambda) \le f(\lambda) - H_d(\lambda)g(\lambda) \le \epsilon_i g(\lambda) \quad \forall\lambda \in \Lambda_i, \tag{27}$$

and then finally the polynomial constraints

$$f(\lambda) - [H_d(\lambda) - \epsilon_i]g(\lambda) \ge 0, \quad \forall\lambda \in \Lambda_i \tag{28}$$

$$-f(\lambda) + [H_d(\lambda) + \epsilon_i]g(\lambda) \ge 0, \quad \forall\lambda \in \Lambda_i \tag{29}$$

It turns out that the non-negativity of the denominator $g(\lambda)$ is in fact a mild constraint. The filter is designed for an undirected graphs, so the eigevalues of the graph Laplacian are positive [2]. Consequently, we need to look at the behaviour of $g(\lambda)$ only on the positive real axis. Since $g(\lambda)$ is continuous, if we are not to have any poles on Λ_i, $g(\lambda)$ has to be either positive or negative. Thus, we are merely fixing the sign (if $0 \in \Lambda_i$, then the sign is already fixed by the constraint $a_0 = 1$).

The ARMA graph filter optimization problem can now be formulated as

$$\min_{\mathbf{a},\mathbf{b}} \tilde{\mathbf{h}}^T\tilde{\mathbf{A}}\tilde{\mathbf{h}} + \gamma\|\tilde{\mathbf{h}}\|^2 \tag{30a}$$

$$\text{s.t. } \tilde{\mathbf{h}} = [\mathbf{b}^T \ \mathbf{a}^T]^T \tag{30b}$$

$$\mathbf{a}^T\mathbf{u}_1 = 1 \tag{30c}$$

$$\mathbf{a}^T\boldsymbol{\lambda} \ge 0, \quad \forall\lambda \tag{30d}$$

$$\mathbf{b}^T\boldsymbol{\lambda} - [H_d(\lambda) - \epsilon_i]\mathbf{a}^T\boldsymbol{\lambda} \ge 0, \quad \forall\lambda \in \Lambda_i \tag{30e}$$

$$-\mathbf{b}^T\boldsymbol{\lambda} + [H_d(\lambda)\mathbf{a}^T\boldsymbol{\lambda} + \epsilon_i]\mathbf{a}^T\boldsymbol{\lambda} \ge 0, \quad \forall\lambda \in \Lambda_i, \tag{30f}$$

which is again a convex optimization problem.

V. Discretization

It might be necessary for reasons of computational complexity or numerical problems to solve the discretized version of the graph filter design problem, in which the eigenvalue interval is discretized into a set of points λ_n. For the graph FIR filter, the discretized optimization is

$$\min_{\mathbf{h}} \epsilon_{\text{tot}} + \gamma\|\mathbf{h}\|^2 \tag{31a}$$

$$\text{s.t. } \sum_n \left|\mathbf{h}^T\boldsymbol{\lambda}_n - H_d(\lambda_n)\right|^2 w(\lambda_n) \le \epsilon_{\text{tot}} \tag{31b}$$

$$|\mathbf{h}^T\boldsymbol{\lambda}_n - H_d(\lambda_n)| \le \epsilon_i, \quad \forall\lambda_n \in \Lambda_i. \tag{31c}$$

For the ARMA filter, a similar additional constraint $g(\lambda_n) \ge 0$ as in the sum-of-squares polynomial optimization is needed. A discrete version of the ARMA filter optimization can be formulated as

$$\min_{\mathbf{a},\mathbf{b}} \epsilon_{\text{tot}} + \gamma(\|\mathbf{a}\|^2 + \|\mathbf{b}\|^2) \tag{32a}$$

$$\text{s.t. } \mathbf{a}^T\boldsymbol{\lambda}_n \ge 0, \quad \forall\lambda_n \tag{32b}$$

$$\sum_n \left|\mathbf{b}^T\boldsymbol{\lambda}_n - H_d(\lambda_n)\mathbf{a}^T\boldsymbol{\lambda}_n\right|^2 w(\lambda_n) \le \epsilon_{\text{tot}} \tag{32c}$$

$$\left|\mathbf{b}^T\boldsymbol{\lambda}_n - H_d(\lambda_n)\mathbf{a}^T\boldsymbol{\lambda}_n\right| \le \epsilon_i\mathbf{a}^T\boldsymbol{\lambda}_n, \quad \forall\lambda_n \in \Lambda_i, \tag{32d}$$

where the last constraint is a second-order cone constraint [12].

Although these discrete problems are convex, it should be noted that unlike in the polynomial optimization, the ripple constraints are not guaranteed for all values of λ.

VI. Examples

In this section, we show the results of applying the proposed filter design method. The desired filter was a low-pass filter with a cutoff at 0.5. The stopband was the interval [0.7, 2], so the transient band was between 0.5 and 0.7. The regularization parameter γ was chosen to be zero.

Eleventh order FIR and ARMA filters were designed with the proposed method and also with the least squares method for comparison. The optimization of the proposed method was carried out using CVX [13], [14].

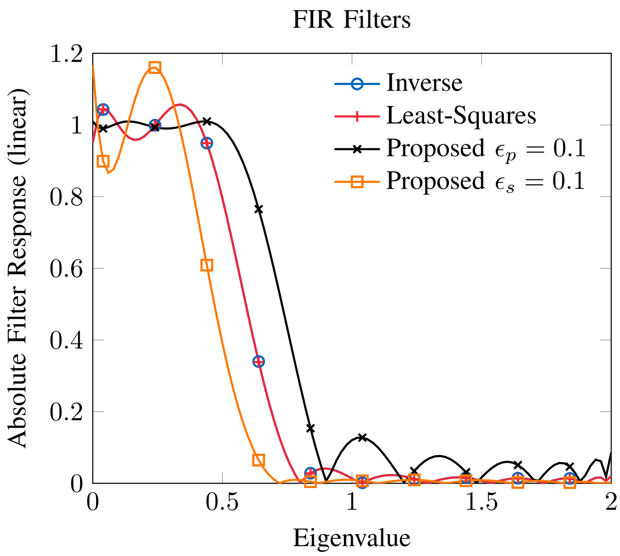

Fig. 1. FIR filters. The least-squares and inverse solutions are very similar, but neither can control the ripple levels. With the proposed method, is is possible to achieve low ripple on the passband or the stopband.

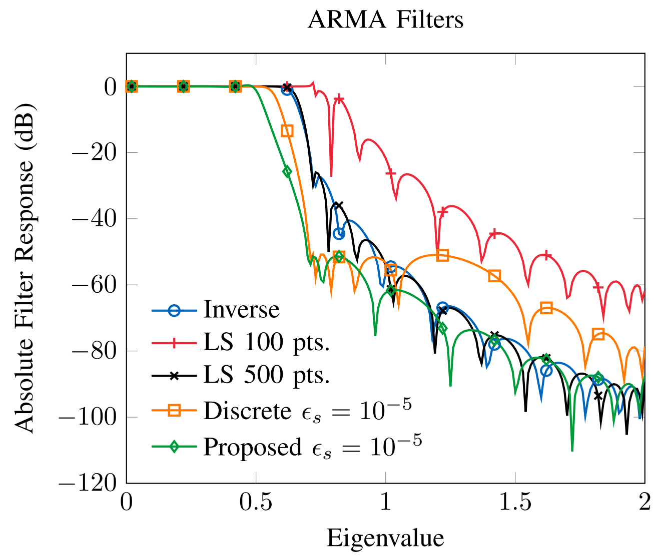

Fig. 2. ARMA filters. The least-squares approximation produces a bad result with 100 points, and 500 points are required to bring to the level of the inverse solution. Both the polynomial method and its discrete version produce a filter with the desired characteristics, but the proposed polynomial one has overall higher attenuation on the stopband and lower approximation error.

Fig.1 shows the responses of the optimized graph FIR filters. The inverse solution of (11) and the least-squares fit are close as expected. In addition, two filters were designed with the proposed method. The first one had the passpand ripple ϵ_p constrained to 0.1. This way, a FIR filter with a much lower passband ripple compared to the inverse and the LS solutions is achieved. The drawback is that the ripple on the stopband has increased significantly.

For the other filter optimized with the proposed method, the stopband ripple ϵ_s was constrained to 0.1. Consequently, a filter with a low ripple on the stopband was achieved, naturally at the cost of an increased passband ripple.

The ARMA filter responses are shown in Fig.2. This figure shows the responses for two least-squares solutions, one that uses 100 points and the other using 500 points. The former one has resulted in a poor filter, whereas the 500 points are already sufficient to bring the filter close to the inverse solution of (25).

Fig.2 also shows the responses of the proposed sum-of-squares formulation (30) and the discretized version (32). The stopband ripple was constrained to 10^{-5} for both methods. Both design problems are feasible, so the desired stopband ripple is achieved, but the proposed solution has overall a better attenuation on the stopband.

The CPU used by the solver to get the filter coefficients is approximately 0.5 seconds for the discretized approach and 0.9 seconds for the polynomial approach, so the discretization is a faster way to solve the filter design problem. However, the interated square error of the filter response is 0.092 for the discretized approach and 0.006 for the polynomial aproach. Therefore, a better filter can be obtained using the polynomial optimization.

VII. Conclusions

In this paper, we have proposed a graph filter design method based on the sum-of-squares representation of polynomials that can be used for optimization of both FIR and ARMA filters. Contrary to the discretized methods, the proposed optimization method allows one to control the ripple amplitude on the passband and the stopband exactly, also for ARMA filters. The proposed approach leads to a convex optimization problem so that the globally optimal filter can be found efficiently.

The numerical examples demonstrated that unlike with the least-squares approximation, filters with the desired stopband and passband ripples could be achieved. The polynomial method also yielded a better filter than the discretized approximation. It was seen that the passband and the stopband ripple are competing design goals, but if the filter with the desired ripple levels is feasible, it can be found with the proposed approach.

The proposed filter design method can be applied to undirected graphs. This method will be extended to directed graphs in a future work.

References

[1] A. Sandryhaila and J. M. F. Moura, "Discrete signal processing on graphs," *IEEE Transactions on Signal Processing*, vol. 61, no. 7, pp. 1644–1656, Apr. 2013.

[2] 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, May 2013.

[3] A. Sandryhaila and J. M. F. Moura, "Discrete signal processing on graphs: Frequency analysis," *IEEE Transactions on Signal Processing*, vol. 62, no. 12, pp. 3042–3054, June 2014.

[4] A. Loukas, A. Simonetto, and G. Leus, "Distributed autoregressive moving average graph filters," *IEEE Signal Processing Letters*, vol. 22, no. 11, pp. 1931–1935, Nov. 2015.

[5] 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, Jan. 2017.

[6] S. Segarra, A. G. 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, Aug. 2017.

[7] A. G. Marques, S. Segarra, G. Leus, and A. Ribeiro, "Stationary graph processes and spectral estimation," *IEEE Transactions on Signal Processing*, vol. 65, no. 22, pp. 5911–5926, Nov. 2017.

[8] S. Segarra, A. G. Marques, G. R. Arce, and A. Ribeiro, "Design of weighted median graph filters," in *2017 IEEE 7th International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*, Dec. 2017, pp. 1–5.

[9] O. Teke and P. P. Vaidyanathan, "Extending classical multirate signal processing theory to graphs—part i: Fundamentals," *IEEE Transactions on Signal Processing*, vol. 65, no. 2, pp. 409–422, Jan. 2017.

[10] 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, Dec. 2018.

[11] Bogdan Alexandru Dumitrescu, *Positive Trigonometric Polynomials and Signal Processing Applications*, Springer Netherlands, 2007.

[12] Stephen Boyd and Lieven Vandenberghe, *Convex Optimization*, Cambridge University Press, 2004.

[13] M. Grant and S. Boyd, "Graph implementations for nonsmooth convex programs," in *Recent Advances in Learning and Control*, V. Blondel, S. Boyd, and H. Kimura, Eds., Lecture Notes in Control and Information Sciences, pp. 95–110. Springer-Verlag Limited, 2008.

[14] Inc. CVX Research, "CVX: Matlab software for disciplined convex programming, version 2.1," http://cvxr.com/cvx, June 2015.