

# WLS DESIGN OF ARMA GRAPH FILTERS USING ITERATIVE SECOND-ORDER CONE PROGRAMMING

Darukeesan Pakiyarajah\*, Chamira U. S. Edussooriya<sup>†,‡</sup>

\*Dept. of Electrical and Electronic Engineering, University of Jaffna, Sri Lanka

<sup>†</sup>Dept. of Electronic and Telecommunication Engineering, University of Moratuwa, Sri Lanka

<sup>‡</sup>Dept. of Electrical and Computer Engineering, Florida International University, FL, USA

## ABSTRACT

We propose a weighted least-square (WLS) method to design autoregressive moving average (ARMA) graph filters. We first express the WLS design problem as a numerically-stable optimization problem using Chebyshev polynomial bases. We then formulate the optimization problem with a nonconvex objective function and linear constraints for stability. We employ a relaxation technique and convert the nonconvex optimization problem into an iterative second-order cone programming problem. Experimental results confirm that ARMA graph filters designed using the proposed WLS method have significantly improved frequency responses compared to those designed using previously proposed WLS design methods.

**Index Terms**— Graph filters, ARMA filters, WLS design, Chebyshev polynomials, second-order cone programming.

## 1. INTRODUCTION

Graph signals can represent data associated with irregular structures, for example, in social networks, sensor networks, power grids and transportation networks [1–3]. The introduction of the graph Fourier transform [2] has led to the processing of graph signals in the graph frequency domain, where graph filters constitute a key class of graph systems. A graph filter shapes the spectral content of a graph signal and are employed in applications such as signal denoising and smoothing [4, 5], classification [6], clustering [7], analog network coding [8], and signal reconstruction [9].

Graph filters are realized as finite-extent impulse response (FIR) filters with polynomial frequency responses, and autoregressive moving average (ARMA) filters with a rational frequency responses and infinite-extent impulse responses (IIRs). In the case of the time-varying graph signals, ARMA filters have the ability to filter graph signals in both the graph frequency domain and regular temporal frequency domain [10]. Several methods have been proposed to design ARMA graph filters including rational Butterworth filter [11], rational Chebyshev filter [12], spectral-transformation-based methods [13], and optimization methods [14, 15].

The optimization methods, unlike other methods, lead to graph filters having frequency responses optimal in either weighted least-square (WLS) or minimax sense. There are two fundamental challenges associated with the optimization methods for ARMA graph filters: the optimization problem is nonconvex and highly susceptible to numerical instability. The WLS method proposed in [14] convert the nonconvex optimization problem to a convex optimization problem by modifying the objective function. Therefore, ARMA graph filters designed using this method are not necessarily optimal in the WLS sense [16]. In [15], an iterative scheme is employed without any modification to the nonconvex objective function. However, this WLS method does not incorporate stability constraints, possibly leading to unstable designs, and does not monotonically converge for some designs.

In this paper, we propose a WLS method to design ARMA graph filters with a *nonconvex objective function and linear constraints for stability*. We achieve our formulation by converting the filter design problem into a numerically-stable optimization problem using Chebyshev polynomial bases and employing a relaxation technique. We employ an iterative second-order cone programming (SOCP) scheme to solve the optimization problem. Experimental results confirm that ARMA graph filter designed using the proposed WLS method has *significantly improved* frequency response compared to those designed using previously proposed WLS design methods, in particular, 88% reduction in the passband ripple and 17 dB reduction in the sum-of-square-error (SSE) are achieved compared to the state-of-the-art WLS method [14].

## 2. PROPOSED WLS DESIGN METHOD

### 2.1. Problem Formulation

The frequency response  $h(\lambda)$  of an ARMA graph filter of order  $(P, Q)$  can be expressed as [10]

$$h(\lambda) = \frac{\sum_{p=0}^P b_p \lambda^p}{1 + \sum_{q=1}^Q a_q \lambda^q}, \quad (1)$$

where  $\lambda$  is the graph spectral frequency. In the design of the ARMA graph filter, we need to determine the coefficients  $b_0, b_1, \dots, b_P$ , and  $a_1, a_2, \dots, a_Q$  such that the spectral response  $h(\lambda)$  approximates a given ideal spectral response  $h_d(\lambda)$  as closely as possible. Because the denominator and numerator polynomials of  $h(\lambda)$  are linear combination of monomial basis  $\{1, \lambda, \lambda^2, \dots, \lambda^M\}$  and  $\lambda \in [0, 2]$ , the conventional optimization techniques employed in digital IIR filter designs are highly susceptible to numerical instability for the designs of higher-order ARMA filters of the form given in (1). Due to the limited space, we do not present definitions of graph signals and graph filters, and the reader is referred to [2] and [3].

To convert the filter design problem into a numerically stable problem, without loss of generality, we express  $h(\lambda)$  in the form

$$h(\lambda) = \frac{\sum_{p=0}^P \beta_p T_p(1-\lambda)}{1 + \sum_{q=1}^Q \alpha_q T_q(1-\lambda)}, \quad (2)$$

where  $T_n(x)$  is the Chebyshev polynomial of order  $n$ , which satisfies the recursive relation  $T_n(x) = 2xT_{n-1}(x) - T_{n-2}(x)$  for  $n \geq 2$  with  $T_0(x) = 1$  and  $T_1(x) = x$ . We note that the use of Chebyshev polynomials for the design of IIR graph filters is inspired from previous works on the Chebyshev polynomial based designs of FIR graph filters [17, 18]. Now, we express  $h(\lambda)$  as  $h(\lambda) = \frac{c_P^T(\lambda)\beta}{1 + c_Q^T(\lambda)\alpha}$ , where  $\alpha = [\alpha_1, \alpha_2, \dots, \alpha_Q]^T$ ,  $\beta = [\beta_0, \beta_1, \dots, \beta_P]^T$ ,  $c_P(\lambda) = [T_0(1-\lambda), T_1(1-\lambda), \dots, T_P(1-\lambda)]^T$ , and  $c_Q(\lambda) = [T_1(1-\lambda), T_2(1-\lambda), \dots, T_Q(1-\lambda)]^T$ . Here,  $c_p$  and  $c_Q$  are the orthogonal shifted Chebyshev polynomial bases. The WLS design of the ARMA graph filter can then be expressed as the minimization problem given by

$$\underset{\alpha, \beta}{\text{minimize}} \quad J(\alpha, \beta) \quad (3a)$$

$$\text{subject to: } h(\lambda) \text{ is stable for } \lambda \in [0, 2]. \quad (3b)$$

The objective function  $J(\alpha, \beta)$  is defined by  $J(\alpha, \beta) = \int_0^2 W(\lambda) [h(\lambda) - h_d(\lambda)]^2 d\lambda$ , where  $W(\lambda)$  is a nonnegative weighting function. Note that the properties  $|1 - \lambda| \leq 1$  and  $|T_n(x)| \leq 1$  for  $|x| \leq 1$  of the selected orthogonal Chebyshev polynomial bases *significantly improve* the numerical stability of the optimization problem in (3a) and (3b) compared to an optimization problem corresponding to the monomial basis [19, ch. 7.3.4]. Thus, we can employ the conventional optimization techniques to design the ARMA filters of the form given in (2) for reasonably large values of  $P$  and  $Q$ .

## 2.2. Stability Condition

A causal and stable ARMA filter can be obtained when  $h(\lambda)$  does not have any pole in the range  $\lambda \in [0, 2]$  [15], or equivalently, when  $1 + c_Q^T(\lambda)\alpha \neq 0$  for  $\lambda \in [0, 2]$ . A practical way

to incorporate this condition in optimization problems is by slightly modifying this constraint as [16]

$$1 + c_Q^T(\lambda)\alpha \geq \epsilon \quad \text{for } \lambda \in [0, 2], \quad (4)$$

where  $\epsilon$  is a small positive quantity. Now, we can express the constraint in (4) equivalently as a linear constraint given by

$$-c_Q^T(\lambda)\alpha \leq 1 - \epsilon \quad \text{for } \lambda \in [0, 2]. \quad (5)$$

## 2.3. Iterative Scheme

In this subsection, we present the iterative scheme required to solve the optimization problem in (3a) and (3b). Using (2), we express  $J(\alpha, \beta)$  as

$$\begin{aligned} J(\alpha, \beta) &= \int_0^2 W(\lambda) \left[ \frac{c_P^T(\lambda)\beta}{1 + c_Q^T(\lambda)\alpha} - h_d(\lambda) \right]^2 d\lambda \\ &= \int_0^2 \frac{W(\lambda)}{(1 + c_Q^T(\lambda)\alpha)^2} \left[ c_P^T(\lambda)\beta \right. \\ &\quad \left. - h_d(\lambda)c_Q^T(\lambda)\alpha - h_d(\lambda) \right]^2 d\lambda, \end{aligned} \quad (6)$$

which is a *nonconvex* function. We note that the WLS method proposed in [14] neglects the term  $(1 + c_Q^T(\lambda)\alpha)^2$  beneath  $W(\lambda)$  and minimizes the so-called modified error given by

$$\hat{J}(\alpha, \beta) = \int_0^2 W(\lambda) \left[ c_P^T(\lambda)\beta - h_d(\lambda)c_Q^T(\lambda)\alpha - h_d(\lambda) \right]^2 d\lambda. \quad (7)$$

In this case,  $\alpha$  and  $\beta$  that minimize  $\hat{J}(\alpha, \beta)$  do not necessarily minimize (6). In general, the solution obtained by minimizing (7) is not optimal in the WLS sense [16]. To incorporate the term  $(1 + c_Q^T(\lambda)\alpha)^2$  into the optimization problem, we adopt an iterative scheme given by

$$J(\alpha_k, \beta_k) = \int_0^2 W_k(\lambda) \left[ c_P^T(\lambda)\beta_k - h_d(\lambda)c_Q^T(\lambda)\alpha_k - h_d(\lambda) \right]^2 d\lambda, \quad (8)$$

$$W_k(\lambda) = \frac{W(\lambda)}{(1 + c_Q^T(\lambda)\alpha_{k-1})^2}, \quad (9)$$

where  $k = 0, 1, 2, \dots$ , and  $\alpha_k, \beta_k$  are variables to be determined in the  $k^{\text{th}}$  iteration. The initial value  $\alpha_0$  required to compute  $W_1(\lambda)$  should be taken such that it satisfies the condition in (5). For instance,  $\alpha_0 = [0, 0, \dots, 0]^T$  is a suitable initial value.

The nonconvex objective function  $J(\alpha, \beta)$  in (6) generally has multiple minima. Therefore, the solution obtained by iterative scheme in (8) and (9) can be expected to be only a local minimum of  $J(\alpha, \beta)$  [16]. The systematic procedure required to solve the iterative scheme in (8) and (9), subject to the stability constraint in (5), is presented in Sec. 2.6.

## 2.4. SOCP Approach

In this subsection, we convert the optimization problem of minimizing  $J(\alpha_k, \beta_k)$  subject to (5) into an SOCP problem. To this end, we express  $J(\alpha_k, \beta_k)$  as,

$$J(\mathbf{h}_k) = \int_0^2 W_k(\lambda) [\mathbf{d}(\lambda)^T \mathbf{h}_k - h_d(\lambda)]^2 d\lambda, \quad (10)$$

where  $\mathbf{d}(\lambda) = [c_P^T(\lambda) \quad -h_d(\lambda)c_Q^T(\lambda)]^T$ , and  $\mathbf{h}_k = [\beta_k^T \quad \alpha_k^T]^T$ . The expression in (10) can be expanded as

$$J(\mathbf{h}_k) = \mathbf{h}_k^T \mathbf{Q}_k \mathbf{h}_k - 2\mathbf{q}_k^T \mathbf{h}_k + p_k, \quad (11)$$

where  $\mathbf{Q}_k = \int_0^2 W_k(\lambda) \mathbf{d}(\lambda) \mathbf{d}^T(\lambda) d\lambda$ ,  $\mathbf{q}_k = \int_0^2 W_k(\lambda) h_d(\lambda) \mathbf{d}(\lambda) d\lambda$ , and  $p_k = \int_0^2 W_k(\lambda) h_d^2(\lambda) d\lambda$ . We then express (11) as

$$\begin{aligned} J(\mathbf{h}_k) &= \mathbf{h}_k^T \mathbf{Q}_k^{1/2} \mathbf{Q}_k^{1/2} \mathbf{h}_k - 2\mathbf{h}_k^T \mathbf{Q}_k^{1/2} \mathbf{Q}_k^{-1/2} \mathbf{q}_k + p_k \\ &= \|\hat{\mathbf{Q}}_k \mathbf{h}_k - \hat{\mathbf{q}}_k\|_2^2 - \hat{\mathbf{q}}_k^T \hat{\mathbf{q}}_k + p_k, \end{aligned} \quad (12)$$

where  $\hat{\mathbf{Q}}_k = \mathbf{Q}_k^{1/2}$ ,  $\hat{\mathbf{q}}_k = \mathbf{Q}_k^{-1/2} \mathbf{q}_k$ , and  $\|\cdot\|_2$  is the 2-norm of a vector. Because  $p - \hat{\mathbf{q}}_k^T \hat{\mathbf{q}}_k$  is a constant term for a given  $k$ , we can equivalently formulate the optimization problem of minimizing  $J(\mathbf{h}_k)$  subject to (5) as

$$\underset{\mathbf{h}_k}{\text{minimize}} \quad \eta_k \quad (13a)$$

$$\text{subject to:} \quad \|\hat{\mathbf{Q}}_k \mathbf{h}_k - \hat{\mathbf{q}}_k\|_2 \leq \eta_k, \quad (13b)$$

$$\mathbf{g}^T(\lambda) \mathbf{h}_k \leq 1 - \epsilon \quad \text{for } \lambda \in [0, 2], \quad (13c)$$

where  $\eta_k$  is an upper bound on  $\|\hat{\mathbf{Q}}_k \mathbf{h}_k - \hat{\mathbf{q}}_k\|_2$ ,  $\mathbf{g}(\lambda) = [\mathbf{O}_{1 \times (P+1)} \quad -c_Q^T(\lambda)]^T$ , and  $\mathbf{O}_{M \times N}$  is an  $M \times N$  zero matrix.

Next, we consider the discretized version of the optimization problem in (13) with a dense set of values for  $\lambda$ , i.e.,  $\Lambda = \{\lambda_{d,0}, \lambda_{d,1}, \lambda_{d,2}, \dots, \lambda_{d,L}\} \subseteq [0, 2]$ . The straightforward numerical approximations for  $\mathbf{Q}_k$  and  $\mathbf{q}_k$  are

$$\mathbf{Q}_k \approx \sum_{j=0}^L W_k(\lambda_{d,j}) \mathbf{d}(\lambda_{d,j}) \mathbf{d}^T(\lambda_{d,j}) \quad (14)$$

$$\mathbf{q}_k \approx \sum_{j=0}^L W_k(\lambda_{d,j}) h_d(\lambda_{d,j}) \mathbf{d}(\lambda_{d,j}), \quad (15)$$

respectively [20, 21]. Then we can express the optimization problem in (13) as an SOCP problem, in the standard form, as [22, ch. 14.7]

$$\underset{\mathbf{x}_k}{\text{minimize}} \quad \mathbf{f}^T \mathbf{x}_k \quad (16a)$$

$$\text{subject to:} \quad \|\mathbf{A}_k^T \mathbf{x}_k + \mathbf{b}_k\|_2 \leq \mathbf{f}^T \mathbf{x}_k, \quad (16b)$$

$$\mathbf{B}^T \mathbf{x}_k \leq (1 - \epsilon) \mathbf{e}_{L+1}, \quad (16c)$$

where  $\mathbf{f} = [1 \ 0 \ 0 \ \dots \ 0]^T_{1 \times (P+Q+2)}$ ,  $\mathbf{x}_k = [\eta_k \ \mathbf{h}_k^T]^T$ ,  $\mathbf{A}_k = [\mathbf{O}_{(P+Q+1) \times 1} \quad \hat{\mathbf{Q}}_k]^T$ ,  $\mathbf{b}_k = -\hat{\mathbf{q}}_k$ ,  $\mathbf{B} = [\mathbf{O}_{(L+1) \times 1} \quad [\mathbf{g}(\lambda_{d,0}) \ \mathbf{g}(\lambda_{d,1}) \ \dots \ \mathbf{g}(\lambda_{d,L})]^T]^T$ , and  $\mathbf{e}_{L+1} = [1 \ 1 \ \dots \ 1]^T_{1 \times (L+1)}$ . In MATLAB, the SOCP problem in (16) can be effectively solved using convex optimization toolboxes such as CVX [23, 24].

## 2.5. Convergence of the Iterative Scheme

We now present a modification required to ensure the convergence of the iterative scheme proposed in Sec. 2.3. Because the original optimization problem in (3a) and (3b) is nonconvex, the proposed iterative technique is not necessarily guaranteed to converge for all filter parameters. To handle this issue, we adopt the relaxation technique proposed in [16], and slightly modify the iterative scheme. Let  $\Phi$  be the operator that maps the initial point  $\mathbf{x}_{k-1}$  to the solution of the SOCP problem in (16). Then, the iterative scheme proposed in Sec. 2.3 can be expressed as  $\mathbf{x}_k = \Phi(\mathbf{x}_{k-1})$ . Now, we modify this to

$$\mathbf{x}_k = \gamma \Phi(\mathbf{x}_{k-1}) + (1 - \gamma) \mathbf{x}_{k-1}, \quad (17)$$

where  $\gamma \in [0, 1]$  is a relaxation constant. Equivalently, we can state that  $\mathbf{x}_k$  is obtained by combining the solution of (16) with the initial point  $\mathbf{x}_{k-1}$  used in (16). There is no approach to theoretically determine appropriate  $\gamma$  required for the convergence of the algorithm. However, we can experimentally obtain a feasible range of values for  $\gamma$  [16].

## 2.6. Modified Iterative Algorithm

We now summarize the iterative algorithm described through Secs. 2.1 to 2.5 in this subsection, with a terminating conditions for the iterative scheme. The iteration begins by choosing an initial value for  $\mathbf{x}_0$ . At each iteration, matrices  $\mathbf{A}_k$  and  $\mathbf{b}_k$  are updated, the optimization problem in (16) is solved, and  $\mathbf{x}_k$  is updated according to (17). The iteration continues until  $\|\mathbf{x}_k - \mathbf{x}_{k-1}\|_\infty$  is less than a prescribed tolerance  $\delta_t$  or the number of iterations reach a predefined maximum iteration  $k_{max}$ . Here,  $\|\cdot\|_\infty$  is the infinity-norm of a vector. At the end of this iterative process, the convergence is claimed, and  $\mathbf{x}_k$  is considered as the optimal solution  $\mathbf{x}_{opt}$  of the WLS optimization problem in (16). This modified algorithm is presented in Alg. 1.

## 3. EXPERIMENTAL RESULTS

In this section, we present experimental results obtained for an example design of a lowpass ARMA graph filter, and we compare the performance of the proposed method with the inverse solution and peak-error constrained WLS design in [14], the iterative scheme in [15], and the least-square

---

**Algorithm 1** Modified Iterative Algorithm

---

**Input:**  $P, Q, L, \Lambda, h_d(\lambda_{d,i}), W(\lambda_{d,i}), \epsilon, \gamma, \delta_t, k_{max}$ **Initialize:**  $k \leftarrow 1, \mathbf{x}_0$ **Define:**  $\mathbf{f}, \mathbf{B}, \mathbf{e}_{L+1}$ **Iteration:**

- 1: Compute  $\mathbf{A}_k, \mathbf{b}_k$
- 2: Solve the SOCP problem in (16) for  $\mathbf{x}_k$
- 3:  $\mathbf{x}_k \leftarrow \gamma \mathbf{x}_k + (1 - \gamma) \mathbf{x}_{k-1}$
- 4: **if**  $\|\mathbf{x}_k - \mathbf{x}_{k-1}\|_\infty > \delta_t$  and  $k < k_{max}$  **then**
- 5:      $k \leftarrow k + 1$
- 6:     Go to step 1
- 7: **end if**

**Output:**  $\mathbf{x}_{opt} \leftarrow \mathbf{x}_k$ 

---

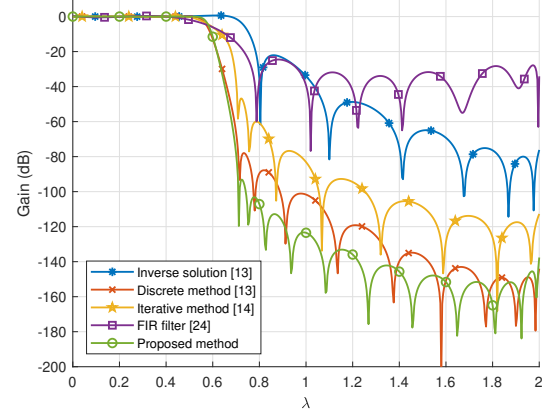
FIR filter in [25] to confirm the effectiveness of the proposed method. We design all the filters using MATLAB and CVX [23, 24] as the optimization toolbox.

We define the ideal frequency response of the filter as,

$$h_d(\lambda) = \begin{cases} 1, & 0 \leq \lambda \leq \lambda_p \\ \text{don't care}, & \lambda_p < \lambda < \lambda_s \\ 0, & \lambda_s \leq \lambda \leq 2 \end{cases} \quad (18)$$

where  $\lambda_p$  and  $\lambda_s$  are the passband and stopband edges of the lowpass filter, respectively. We select  $\lambda_p = 0.5$  and  $\lambda_s = 0.7$ , respectively, which are the specifications employed in the design example considered in [14]. Furthermore, we select the weight function  $W(\lambda)$  as 1, 0 and 1 for  $0 \leq \lambda \leq \lambda_p$ ,  $\lambda_p < \lambda < \lambda_s$ , and  $\lambda_s \leq \lambda \leq 2$ , respectively. We select the order of the filter as  $P = Q = 11$  (the same as [14]) and the discretized values of  $\lambda$  as  $\lambda_{d,i} = \frac{2i}{L}$ , where  $L = 500$ . The total number of points in the passband and stopband is 401. Furthermore, we select the parameters  $\epsilon = 10^{-5}$  (see (5)),  $k_{max} = 25$ , and  $\delta_t = 2 \times 10^{-8}$  (see Alg. 1). We use the initial value of  $\mathbf{x}_0 = \mathbf{O}_{(P+Q+2) \times 1}$  in this design. From experiments, we found that  $\gamma \in [0.1, 0.5]$  ensures the convergence of the iterative algorithm, and we use  $\gamma = 0.25$  for this design. In this case, the algorithm converges after 7 iterations.

Next, we employ the previously proposed ARMA graph filter design methods in [14]: discrete and inverse solution (see (25) in [14]), the iterative method in [15], and the WLS FIR graph filter design method in [25] to design graph filters for the same specifications. Here, we set the regularization parameter as 0.005 for the inverse solution method, and the peak error constraint as  $10^{-5}$  for the discrete method. Furthermore, we set norm penalty constant as  $5.5 \times 10^{-6}$  for the FIR filter design [25]. The orders of the ARMA graph filter is 11 and that of the FIR graph filter is 22. The iterative method in [15] does not monotonically converge, and we consider the best filter design achieved in the 7<sup>th</sup> iteration among the first 30 iterations. We present the magnitude responses of the designed filters in Fig. 1. We observe that the magnitude response of the ARMA graph filter designed using



**Fig. 1:** The magnitude response (in dB) of the lowpass graph filter designed using different WLS methods.

**Table 1:** Parameters of the magnitude responses achieved with the different designs of the lowpass graph filter.

Method	Parameter		
	$\delta_p$ (dB)	$\delta_s$ (dB)	SSE (dB)
Inverse solution [14]	0.1909	1.5997	7.5223
Discrete [14]	0.0134	70.4361	-50.5811
Iterative [15]	0.2284	43.7050	-28.3991
FIR filter [25]	1.5305	15.2755	-1.1277
Proposed	<b>0.0016</b>	<b>77.5931</b>	<b>-67.7455</b>

the proposed method well approximates the desired magnitude response. Furthermore, we compare the maximum passband ripple ( $\delta_p$ ), maximum stopband attenuation ( $\delta_s$ ), and the SSE of the designed filters in Table 1. It is evident that the proposed design method *significantly outperforms* those proposed in [14], [15] and [25] with *all the three parameters*. In particular,  $\delta_p$  and SSE are reduced by 88% and 17 dB, respectively, compared to [14].

#### 4. CONCLUSION AND FUTURE WORK

We propose a WLS design method for ARMA graph filters by employing Chebyshev polynomials and iterative SOCP scheme. In contrast to previously proposed WLS design methods, we consider the optimization problem without any modifications to the nonconvex objective function. Experimental results confirm that ARMA graph filters designed using the proposed WLS method have significantly improved frequency response compared to those designed with previously proposed WLS design methods. Future work includes the design of ARMA graph filters optimal in the minimax sense.

#### 5. ACKNOWLEDGMENT

Authors thank the University of Moratuwa for financial support.

## 6. 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 Process. Mag.*, vol. 30, no. 3, pp. 83–98, May 2013.
- [2] A. Sandryhaila and J. M. Moura, "Discrete signal processing on graphs," *IEEE Trans. Signal Process.*, vol. 61, no. 7, pp. 1644–1656, Apr. 2013.
- [3] A. Ortega, P. Frossard, J. Kovačević, J. M. Moura, and P. Vandergheynst, "Graph signal processing: Overview, challenges, and applications," *Proc. IEEE*, vol. 106, no. 5, pp. 808–828, May 2018.
- [4] F. Zhang and E. Hancock, "Graph spectral image smoothing using the heat kernel," *Pattern Recognition*, vol. 41, pp. 3328–3342, Nov. 2008.
- [5] E. Isufi and G. Leus, "Distributed sparsified graph filters for denoising and diffusion tasks," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2017, pp. 5865–5869.
- [6] J. Ma, W. Huang, S. Segarra, and A. Ribeiro, "Diffusion filtering for graph signals and its use in recommendation systems," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Mar. 2016.
- [7] N. Tremblay, G. Puy, R. Gribonval, and P. Vandergheynst, "Compressive spectral clustering," in *Proc. Int. Conf. Machine Learning*, vol. 48, 2016, pp. 1002–1011.
- [8] S. Segarra, A. G. Marques, and A. Ribeiro, "Optimal graph-filter design and applications to distributed linear network operators," *IEEE Trans. Sig. Process.*, vol. 65, no. 15, pp. 4117–4131, Aug. 2017.
- [9] E. Isufi, P. Di Lorenzo, P. Banelli, and G. Leus, "Distributed Wiener-based reconstruction of graph signals," in *Proc. IEEE Stat. Sig. Process. Workshop*, 2018, pp. 21–25.
- [10] E. Isufi, A. Loukas, A. Simonetto, and G. Leus, "Autoregressive moving average graph filtering," *IEEE Trans. Sig. Process.*, vol. 65, no. 2, pp. 274–288, 2017.
- [11] X. Shi, H. Feng, M. Zhai, T. Yang, and B. Hu, "Infinite impulse response graph filters in wireless sensor networks," *IEEE Sig. Process. Letters*, vol. 22, no. 8, pp. 1113–1117, 2015.
- [12] O. Rimleascaia and E. Isufi, "Rational Chebyshev graph filters," in *2020 54th Asilomar Conf. Signals, Systems, Computers*, 2020, pp. 736–740.
- [13] C.-C. Tseng, "Rational graph filter design using spectral transformation and IIR digital filter," in *2020 IEEE Region 10 Conf. (TENCON)*, 2020, pp. 247–250.
- [14] T. Aittomaki and G. Leus, "Graph filter design using sum-of-squares representation," in *Proc. European Signal Process. Conf.*, 2019, pp. 1–5.
- [15] J. Liu, E. Isufi, and G. Leus, "Filter design for autoregressive moving average graph filters," *IEEE Trans. Signal, Info. Process. Netw.*, vol. 5, no. 1, pp. 47–60, 2019.
- [16] W.-S. Lu, S.-C. Pei, and C.-C. Tseng, "A weighted least-squares method for the design of stable 1-D and 2-D IIR digital filters," *IEEE Trans. Sig. Process.*, vol. 46, no. 1, pp. 1–10, Oct. 1998.
- [17] D. I. Shuman, P. Vandergheynst, D. Kressner, and P. Frossard, "Distributed signal processing via Chebyshev polynomial approximation," *IEEE Trans. Signal, Info. Process. Netw.*, vol. 4, no. 4, pp. 736–751, 2018.
- [18] C.-C. Tseng and S.-L. Lee, "Minimax design of graph filter using Chebyshev polynomial approximation," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 68, no. 5, pp. 1630–1634, May 2021.
- [19] M. Heath and E. Munson, *Scientific Computing: An Introductory Survey*, 2nd ed. Philadelphia: SIAM, 2018.
- [20] W.-S. Lu and T. Hinamoto, "Two-dimensional digital filters with sparse coefficients," *Multidim. Syst. Signal Process.*, vol. 22, no. 1-3, pp. 173–189, Mar. 2011.
- [21] D. Pakiyarajah, S. S. Jayaweera, C. U. S. Edussooriya, C. Wijenayake, and A. Madanayake, "WLS design of M-D complex-coefficient FIR filters with low group delay using second-order cone programming," in *Proc. IEEE Int. Symp. Circuits Syst.*, 2021, pp. 1–5.
- [22] A. Antoniou and W.-S. Lu, *Practical Optimization: Algorithms and Engineering Applications*. New York: Springer, 2007.
- [23] M. Grant and S. Boyd, "CVX: Matlab software for disciplined convex programming, version 2.1," <http://cvxr.com/cvx>, Mar. 2014.
- [24] —, "Graph implementations for nonsmooth convex programs." London: Springer-Verlag, 2008, pp. 95–110.
- [25] C.-C. Tseng and S.-L. Lee, "Design of graph filter using least-squares method with parameter norm penalty," in *Proc. 2020 IEEE Int. Conf. Consumer Electronics - Taiwan (ICCE-Taiwan)*, 2020, pp. 1–2.