Universal Graph Filter Design Based on Butterworth, Chebyshev, and Elliptic Functions

Zirui Ge · Haiyan Guo · Tingting Wang · Zhen Yang

Received: 12 January 2022 / Revised: 30 July 2022 / Accepted: 30 July 2022 / Published online: 19 August 2022
© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2022

Abstract
Graph filters are crucial tools in processing the spectrum of graph signals. In this paper, we propose to design universal IIR graph filters with low computational complexity by using three kinds of functions, which are Butterworth, Chebyshev, and elliptic functions, respectively. Specifically, inspired by the classical analog filter design method, we first derive the zeros and poles of graph frequency responses. With these zeros and poles, we construct the conjugate graph filters to design the Butterworth high-pass graph filter, Chebyshev high-pass graph filter, and elliptic high-pass graph filter, respectively. On this basis, we further propose to construct a desired graph filter of low pass, band pass, and band stop by mapping the parameters of the desired graph filter to those of the equivalent high-pass graph filter. Furthermore, we propose to set the graph filter order given the maximum passband attenuation and the minimum stopband attenuation. Our numerical results show that the proposed graph filter design methods realize the desired frequency response more accurately than the autoregressive moving average graph filter design method, the linear least-squares fitting-based graph filter design method, and the Chebyshev FIR polynomial graph filter design method.

Keywords Graph filters · Butterworth functions · Chebyshev functions · Elliptic functions

Zhen Yang yangz@njupt.edu.cn
Zirui Ge 1019010430@njupt.edu.cn
Haiyan Guo guohy@njupt.edu.cn
Tingting Wang 2018010215@njupt.edu.cn

1 School of Communication and Information Engineering, Nanjing University of Posts and Telecommunications, Nanjing 210023, China
1 Introduction

Graph signal processing (GSP) extends classical digital signal processing (DSP) technologies to signals that reside on irregular structures [13, 18]. Graph filter design is an important research area of GSP. It aims to reserve the desired graph frequencies and attenuate the others. Due to its wide range of applications, graph filter design has attracted increasing attention and shown its advantage in graph wavelets [2, 22], graph signal denoising [1, 4, 11], graph signal recovery [10], speech enhancement [20, 23], and others.

There are generally two types of graph filters, which are finite impulse response (FIR) graph filters [3, 14, 16, 19] and infinite impulse response (IIR) graph filters [5, 7, 8, 17]. FIR graph filters are generally modeled as the polynomials of the graph shift operator. In [14], a linear least-squares fitting (LLS)-based method was proposed to obtain the coefficients of FIR graph filters. In [19], Shuman et al. proposed a method to approximate the graph multipliers by Chebyshev polynomials. In [16], Segarra et al. proposed to design optimal graph filters by using arbitrary linear transformations between graph signals. In [3], Contino et al. proposed to extend distributed FIR graph filters to their edge-variant (EV) versions. It is worth noting that a high-pass filter order is required so that the designed FIR graph filter can realize a sharp transition. However, these can be trapped in an ill condition if it meets a high order.

By contrast, IIR graph filters are generally modeled as rational functions, which outperform polynomials and have low degrees in fitting desired graph filters. In [8, 17], the authors first proposed the concept of the IIR graph filter. In [5, 7], the authors proposed the concept of the universal graph filter which makes graph filters applicable for any graph structure and further designed autoregressive moving average (ARMA) graph filters by utilizing classical temporal ARMA filters. In [12], the authors proposed to construct the Chebyshev I graph filter by using closed-form filter coefficients, which avoids the high computational complexity led by solving complex optimization problems.

In this context, we investigate the universal graph filter design based on the undirected graph. Different from [17] and [12], we propose to construct the desired Butterworth/Chebyshev/elliptic IIR graph filter by a pair of conjugate graph filters. Specifically, we first obtain the zeros and poles of graph frequency responses based on Butterworth, Chebyshev, and elliptic functions, respectively, and further construct the conjugate graph filters to design three types of desired graph filters.

The main contributions of this paper are summarized as follows:

1) Inspired by the classical Butterworth, Chebyshev, and elliptic functions, we derive the zeros and poles of the corresponding graph frequency responses and further build the zero-pole equations which are used for the high-pass universal IIR graph filter design. Furthermore, we propose to design low-pass, band-pass, and band-stop graph filters mapped by the obtained high-pass graph filters.

2) For the integrity of the graph filter design, we also propose to set the order of IIR graph filters with the given requirements.

3) Our simulation results show that our designed IIR filters perform better than the ARMA graph filter, LLS graph filter, and Chebyshev FIR polynomial graph filter.
in terms of graph filter responses. Additionally, the proposed elliptic graph filter has the best performance in approximating the step graph frequency response.

The outline of the paper is as follows. Section 2 introduces the basic concepts of signal processing on graphs and classical filters. Section 3 shows the construction of high-pass, low-pass, band-pass, and band-stop graph filters. In Sect. 4, we show how to set the order of IIR graph filters. Our simulation results are provided in Sect. 5, while Sect. 6 concludes the paper.

2 Related Work

2.1 Graph Filters

Consider an undirected graph \( G = (V, E) \) with \( N \) vertices and \( M \) edges, where \( V = [v_1 \ v_2 \ \cdots v_N] \) denotes the set of \( N \) vertices and \( E \) denotes the set of \( M \) edges. The underlying structure of \( G \) is described by the adjacency matrix \( A \in \mathbb{R}^{N \times N} \) or by Laplacian matrix \( L = D - A \), where \( D \) is the diagonal degree matrix. It is worth noting that the corresponding normalized Laplacian matrix is \( L_n = D^{-1/2}LD^{-1/2} \).

Since \( G \) is undirected, the edge between \( v_i \) and \( v_j \) is the same as that in \( v_j \) and \( v_i \). The Laplacian matrix \( L \) is symmetric, and its eigendecomposition can be further expressed as \( L = U\Lambda U^T \), where \( U \) is the eigenvector matrix and \( \Lambda \) is the diagonal matrix.

In \([12, 21]\), the authors proved that the eigenvalues of the normalized Laplacian matrix are in the interval \([0, 2]\). In \([15]\), the authors defined the eigenvector matrix \( U \) as the graph Fourier transformation basis and the diagonal elements of the diagonal matrix \( \Lambda \) as the graph frequencies \( \lambda_1, \lambda_2, \ldots, \lambda_N \). In \([14]\), the authors proposed to sort the graph frequencies by using the total variation, so that small eigenvalues represent high frequencies and large eigenvalues represent low frequencies. Therefore, a high-pass graph filter aims to retain small eigenvalues and discard large eigenvalues, while a low-pass graph filter aims to retain large eigenvalues and discard small eigenvalues.

By taking the normalized Laplacian matrix as the graph shift operator, the FIR graph filter is expressed as

\[
H = \sum_{i=0}^{N-1} h_i L_n^i, \quad (1)
\]

and the IIR graph filter \([17]\) is

\[
H = \left( \sum_{i=0}^{q} h_i L_n^i \right)^{-1} \sum_{i=0}^{p} h_i L_n^i, \quad \sum_{i=0}^{q} h_i L_n^i \text{ is invertible}. \quad (2)
\]

Both the graph filters’ graph frequency responses are written as

\[
H(\lambda) = \sum_{i=0}^{N-1} h_i \lambda^i, \quad (3)
\]
and
\[
H(\lambda) = \frac{p_n(\lambda)}{p_d(\lambda)} = \frac{\sum_{p=0}^{P} a_p \lambda^p}{\sum_{q=0}^{Q} b_q \lambda^q},
\]
respectively, where \( P, Q \) are the highest order in the numerator and denominator. It is noted that \( \sum_{q=0}^{Q} b_q L_q^n \) is invertible to guarantee a bounded output given a bounded input. That is the stability of a graph filter.

### 2.2 Classical Analog IIR Filters

The magnitude response of an analog IIR filter is given by
\[
|H(\Omega)|^2 = \frac{1}{1 + \varepsilon_p^2 F_N^2(\omega)}, \quad \omega = \frac{\Omega}{\Omega_p},
\]
where \( N \) is the filter order, \( \varepsilon_p \) is the passband ripple parameters, \( \Omega \) is the analog frequency, \( \Omega_p \) is the passband cutoff frequency, and \( F_N(\omega) \) is a function of the normalized frequency \( \omega \).

In classical analog IIR filters, Butterworth, Chebyshev, and elliptic filter design methods are well established and generally applied in different fields [9].

For the Butterworth filter, \( F_B(\omega) = \omega^N \). The Butterworth filter keeps a flat frequency response in the pass band and stop band.

For the Chebyshev I filter,
\[
F_{CI}(\omega) = C_N(\omega) = \begin{cases} 
\cos(N \cos^{-1} \omega), & \omega \leq 1 \\
\text{ch}(N \text{ch}^{-1} \omega), & \omega > 1
\end{cases},
\]
where \( \text{ch}(\cdot) \) is the hyperbolic cosine function.

For the Chebyshev II filter, \( F_{CI1}(\omega) = \left[k_1 C_N \left(k^{-1} \omega^{-1}\right)\right]^{-1}, \) where \( k = \frac{\Omega_p}{\Omega_s}, \ k_1 = \frac{\varepsilon_p}{\varepsilon_s}, \) and \( \varepsilon_s \) is the stopband ripple parameter. The main difference between the graph frequency responses of the Chebyshev I filter and Chebyshev II filter is that the former keeps equal ripple in the passband, while the latter in the stopband.

For the elliptic filter,
\[
F_E(\omega) = \text{cd}(NuK_1, k_1), \quad \omega = \text{cd}(uK, k),
\]
where \( \text{cd}(uK, k) \) is the Jacobian elliptic function with the modulus \( k \) [6], \( u \) is a real number in the interval \([0,1]\), \( K \) and \( K_1 \) are the complete integral of modulus \( k \) and \( k_1 \), respectively.

It is noted that \( F(\omega) \) is normalized for all the four classical analog IIR filters stated above, that is, \( F(1) = 1 \). Additionally, to guarantee the equal-ripple characteristic in
the elliptic case, the function of $F(\omega)$ satisfies the identity

$$F(\omega) = \frac{1}{k_1 F(k^{-1} \omega^{-1})}.$$  \hspace{1cm} (8)

The key to designing analog filters is to obtain the poles and zeros of the magnitude responses. With the obtained zeros and poles, analog filters can be constructed as rational functions.

3 Graph Filter Design Based on Three Functions

In this section, we first investigate the Butterworth graph filter design, the Chebyshev graph filter design, and the elliptic high-pass graph filter design. Then, we propose to design low-pass, band-pass, and band-stop graph filters based on the high-pass graph filter design method above.

3.1 The Construction of High-Pass Graph Filters

3.1.1 The Butterworth High-Pass Graph Filter

By replacing $\Omega$ in (5) with graph frequency $\lambda$, we can express the frequency response of the Butterworth graph filter as

$$H_B(\lambda) = \frac{1}{1 + \varepsilon^2 \omega^{2N}}, \quad \omega = \frac{\lambda}{\lambda_p},$$  \hspace{1cm} (9)

where $\lambda_p$ is the passband cutoff graph frequency. We can obtain the poles of $H_B(\lambda)$ as

$$\lambda = \lambda_p e^{-1/N} e^{j \frac{2m-1}{2N} \pi}, \quad m = 1, 2, \cdots, 2N.$$  \hspace{1cm} (10)

We select the first $N$ poles above the real axis, i.e., $m = 1, 2, \cdots, N$, to construct a graph filter expressed as

$$G_B(\lambda) = \frac{1}{\prod_{i=1}^{N} \left(1 - \lambda / \lambda_{p_i}\right)},$$  \hspace{1cm} (11)

where $G_B(\lambda)$ can attain the desired magnitude response, but its complex-valued frequency response only works in complex signal space. We name another filter $G_B^*(\lambda)$ constructed by the other $N$ points below the real axis, where $G_B^*(\lambda)$ is the conjugate function with $G_B(\lambda)$.

Thus, to eliminate the disallowed phases, we compose the conjugated filters to construct the final IIR graph filter whose frequency response is real-valued. That is different from the classical filters, which only keep the poles of the left complex plane.
We can obtain frequency response of the Butterworth graph filter as

\[ H_B(\lambda) = G_B(\lambda) \cdot \frac{G^*_B(\lambda)}{H_0} = \frac{1}{\prod_{i=1}^{\lceil N/2 \rceil} \left( 1 + 2\lambda^2 \text{imag} \left( \frac{1}{\lambda_{pi}} \right) + \lambda^4 \text{abs} \left( \frac{1}{\lambda_{pi}} \right) \right)}, \]  \tag{12}

where

\[ H_0 = \begin{cases} 
1 - 2\lambda \text{real} \left( \frac{1}{\lambda_{p[N/2]}} \right) + \lambda^2 \text{abs} \left( \frac{1}{\lambda_{p[N/2]}} \right) & \text{if } N \text{ is odd} \\
1 & \text{if } N \text{ is even} 
\end{cases}, \]  \tag{13}

and we indicate by \text{abs}(\cdot), \text{imag}(\cdot), \text{and real}(\cdot) the absolutely value, imaginary part, and real part of a complex value. We can use the convolution operation to compute polynomial multiplication in (12). For a large \( N (N \geq 64) \), we can use the fast Fourier transform (FFT) to further simplify the complexity.

This graph filter (12) is the cascade form proposed in [17]. In the following two cases, we also take the cascade form to construct the graph filter.

Figure 1 illustrates the graph Butterworth filter’s poles for \( N = 16 \), where Fig. 1a, b shows the locations of all the \( 2N \) poles and those of the first \( N \) poles, respectively.

Observe from Eq. (10) and Fig. 1 that none of the poles reside on the real axis. Thus, \( \prod_{i=1}^{N} \left( 1 - \lambda / \lambda_{pi} \right) \neq 0 \) is always satisfied, and the Butterworth graph filter can always provide a bounded output given a bounded input. That is, stability is guaranteed.

3.1.2 The Chebyshev Graph Filter Design

Similar to the Chebyshev I analog filter and the Chebyshev II analog filter, we can express the frequency response of the Chebyshev I graph filter and that of the Chebyshev II graph filter as:
Fig. 2 Locations of Chebyshev I graph filter’s poles. a The locations of all the $2N$ poles; b the locations of the first $N$ poles

Chebyshev II graph filter as

$$H_{CI}(\lambda) = \frac{1}{1 + \varepsilon_p^2 C_N^2(\gamma)}, \quad \gamma = \frac{\lambda}{\lambda_p},$$

and

$$H_{CII}(\lambda) = \frac{1}{1 + \varepsilon_p^2 \left[k_1^2 C_N^2(k^{-1}\gamma^{-1})\right]^{-1}}, \quad \gamma = \frac{\lambda}{\lambda_p},$$

respectively.

We can obtain the poles of $H_{CI}(\lambda)$ as

$$\lambda_{CI} = \lambda_p \cos\left(\frac{2m - 1}{2N} \pi\right) \cdot \text{ch}\left(\frac{1}{N} \text{sh}^{-1}\frac{1}{\varepsilon_p}\right)$$

$$\pm j\lambda_p \sin\left(\frac{2m - 1}{2N} \pi\right) \cdot \text{sh}\left(\frac{1}{N} \text{sh}^{-1}\frac{1}{\varepsilon_p}\right), \quad m = 1, 2, \ldots, 2N.$$  

Figure 2 illustrates Chebyshev I graph filter’s poles in the polar coordinate for $N = 6$. Similar to the Butterworth graph filter, all the $2N$ poles are selected. Observe from Fig. 2 that the poles do not reside on the real axis. The reason for this is that $\text{sh}\left(\frac{1}{N} \text{sh}^{-1}\frac{1}{\varepsilon_p}\right) \neq 0$ and $\sin\left(\frac{2m - 1}{2N} \pi\right) \neq 0$, $m = 1, 2, \ldots, 2N$. Thus, the designed Chebyshev I graph filter satisfies the stability. Given any bounded input, the designed Chebyshev I graph filter always provides a limited output. Different from [12], which directly uses the poles in classical filters, we utilize the conjugate graph filters to obtain the desired graph filter in a simpler form. The construction of the Chebyshev I graph filter is the same as equation (12).
Let us now study the zeros and poles of $H_{CII}(\lambda)$. We can obtain the poles of $H_{CII}(\lambda)$ as

$$
\lambda_{CII} = \lambda_s \frac{c_m \pm j d_m}{c_m^2 + d_m^2}, \quad m = 1, 2, \ldots, 2N,
$$

where $c_m = \cos\left(\frac{2m-1}{2N} \pi\right) \cdot \text{ch}\left(\frac{1}{N} \text{sh}^{-1} \varepsilon_s\right)$, $d_m = \sin\left(\frac{2m-1}{2N} \pi\right) \cdot \text{sh}\left(\frac{1}{N} \text{sh}^{-1} \varepsilon_s\right)$.

The zeros of $H_{CII}(\lambda)$ are

$$
z_{CII} = \frac{\lambda_s}{\cos\left(\frac{2m-1}{2N} \pi\right)}, \quad m = 1, 2, \ldots, 2N.
$$

It is worth noting that in the case where $N$ is odd, there are two zeros with infinite values since $\cos\left(\frac{2m-1}{2N} \pi\right) = 0$, when $m = \lceil \frac{N}{2} \rceil$ and $m = \lceil \frac{N}{2} \rceil + N$, which should be discarded in the construction of graph filters.

Figure 3 illustrates Chebyshev II graph filter’s poles in the polar coordinate for $N = 10$.

The frequency response of the Chebyshev II graph high-pass filter is expressed as

$$
H_{CII}(\lambda) = G_{CII}(\lambda) G_{CII}^c(\lambda) = \frac{H_0 \prod_{i=1}^{\lceil N/2 \rceil} 1 - 2\lambda^2 / \lambda_{zi}^2 + \lambda^4 / \lambda_{zi}^4}{\prod_{i=1}^{\lceil N/2 \rceil} 1 + 2\lambda^2 \text{imag} \left(1/\lambda_{zi}\right)^2 + \lambda^4 \text{abs} \left(1/\lambda_{zi}^4\right)},
$$

where $H_0$ is the same as equation (13).
3.1.3 The Elliptic Graph Filter Design

We can obtain the zeros and poles of elliptic graph filter as
\[ z_E = \text{cd}(u_m K, k), \quad u_m = \frac{2m - 1}{N}, \quad m = 1, 2, \ldots, 2N, \]  
and
\[ \lambda_E = \lambda_p \text{cd} \left( (u_m - jv_0) K, k \right), \quad m = 1, 2, \ldots, 2N, \]
where
\[ v_0 = \frac{j}{NK_1} \text{sn}^{-1} \left( \frac{j}{\varepsilon_p}, k_1 \right). \]  

By combining (21) and (22), we can obtain that \( \lambda_E \) is complex-valued. Hence, no poles reside on the real axis so that the elliptic graph filter is stable.

Figure 4 illustrates the elliptic graph filter’s poles in the polar coordinate for \( N = 8 \).

It is noted that the frequency response of the designed elliptic high-pass graph filter also has two forms, which are similar to that of the designed Chebyshev II graph high-pass filter. Additionally, when \( N \) is odd, the infinite zero points are also discarded.

3.2 Low-Pass, Band-Pass, and Band-Stop Graph Filters

We first investigate mapping the desired low-pass graph filter parameters to the equivalent high-pass graph filter parameters. Let us denote the required pass band cutoff
frequency and stop band cutoff frequency as $\lambda_{lp}$ and $\lambda_{ls}$, respectively. Following the parameter transformation methods in classical analog filter design [9], we can obtain the graph frequency $\lambda$, the $\lambda_p$, and the $\lambda_s$ of the equivalent high-pass graph filter as

$$\lambda = \frac{1}{\lambda_{lp}}, \quad \lambda_p = \frac{1}{\lambda_{lp}'}, \quad \lambda_s = \frac{1}{\lambda_{ls}'.} \quad (23)$$

Let us now map the parameters of the desired band-pass graph filter to those of the equivalent high-pass graph filter. Upon denoting the stop band cutoff frequencies and the pass band cutoff frequencies of the desired band-pass graph filter as $\lambda_{bp1}$, $\lambda_{bp2}$, $\lambda_{bs1}$, and $\lambda_{bs2}$, respectively, we can obtain the $\lambda$, $\lambda_p$, and $\lambda_s$ of the equivalent high-pass graph filter as

$$\lambda = \lambda_{bp} - \lambda_{bp1} \lambda_{bp2} \lambda_{bp}, \quad \lambda_p = \lambda_{bp2} - \lambda_{bp1}, \quad \lambda_s = \min \left( \left| \lambda_{bs1} - \frac{\lambda_{bp1} \lambda_{bp2}}{\lambda_{bp}} \right|, \left| \lambda_{bs2} - \frac{\lambda_{bp1} \lambda_{bp2}}{\lambda_{bp}} \right| \right). \quad (24)$$

Now, we map the parameters of the desired band-stop graph filters to those of the equivalent high-pass graph filters. We denote the required pass band cutoff frequency and stop band cutoff frequency as $\lambda_{bs1}$, $\lambda_{bs2}$, $\lambda_{bs1}$, and $\lambda_{bs2}$, respectively, where $\lambda_{bs1} < \lambda_{bs2} < \lambda_{bs1} < \lambda_{bs2} < \lambda_{bs2}$. We can obtain the $\lambda$, $\lambda_p$, and $\lambda_s$ of the equivalent high-pass graph filter as

$$\lambda = \frac{1}{\lambda_{bs} - \frac{\lambda_{bs1} \lambda_{bs2}}{\lambda_{bs}}}, \quad \lambda_p = \frac{1}{\lambda_{bp} - \frac{\lambda_{bs1} \lambda_{bs2}}{\lambda_{bs}}} \left( \left| \lambda_{bs1} - \frac{\lambda_{bs} \lambda_{bs1}}{\lambda_{bp}} \right|, \left| \lambda_{bs2} - \frac{\lambda_{bs} \lambda_{bs2}}{\lambda_{bp}} \right| \right), \quad \lambda_s = \frac{1}{\lambda_{bs2} - \lambda_{bs1}}. \quad (25)$$

We show an example and illustrate the correspondence between these three types of graph filters and their equivalent high-pass parameters in Table 1.

| Table 1 | Desired graph filter parameters and their equivalent high-pass parameters |
|---------|---------------------------------------------------------------------------------|
| Low pass | Band pass | Stop pass |
| Filter parameters | $\lambda_{lp} = 0.5 \lambda_{lp}' = 0.5$ | $\lambda_{bp} = 0.69, \lambda_{bp1} = 0.7, \lambda_{bp2} = 1.2, \lambda_{bs1} = 0.7, \lambda_{bs2} = 1.7$ | $\lambda_{bs} = 0.29, \lambda_{bs1} = 0.3, \lambda_{bs2} = 1.7, \lambda_{bs} = 1.71$ |
| Equivalent high-pass parameters | $\lambda_p = 0.662 \lambda_s = 0.667$ | $\lambda_p = 0.5, \lambda_s = 0.51$ | $\lambda_p = 0.708, \lambda_s = 0.714$ |
4 The Graph Filter Order

In this section, we study how to set the order of the Butterworth, the Chebyshev I, the Chebyshev II, and the elliptic high-pass graph filter, respectively.

Similar to analog filters, given the constraints of the passband $\lambda_p$ and the stopband $\lambda_s$, the maximum passband attenuation $R_p$ and the minimum stopband attenuation $A_s$ satisfy

$$-20 \log \left( \frac{H(\lambda_p)}{H(0)} \right) \leq R_p,$$

and

$$-20 \log \left( \frac{H(\lambda_s)}{H(0)} \right) \geq A_s,$$

respectively. For the Butterworth graph high-pass filter, by combining (9), (26), and (27), we can obtain that

$$N \geq \log \left( \frac{10^{A_s/20} - 1}{10^{R_p/20} - 1} \right) / \log \left( \frac{\lambda_s}{\lambda_p} \right).$$

By similar analysis, we can conclude that for the Chebyshev I graph high-pass filters, and $N$ satisfies

$$N \geq ch^{-1} \left( \frac{10^{A_s/20} - 1}{10^{R_p/20} - 1} \right) / ch^{-1} \left( \frac{\lambda_s}{\lambda_p} \right).$$

It is noted that for the Chebyshev II graph filter, $k$ should be adjusted to guarantee that the graph filter is equal-ripple. The adjusted $k$ is given by

$$k = 1/ch \left( ch^{-1} \left( \frac{1}{k_1} \right) / N \right).$$

Similarly, for the high-pass elliptic graph filter,

$$N \geq \frac{K \cdot K_1'}{K' \cdot K_1}, \quad k_1 = \sqrt{\frac{10^{A_s/20} - 1}{10^{R_p/20} - 1}}.$$

It is worth noting that $K_1'$ is the complete integral with $k_1'$ and $k_1' = \sqrt{1 - k_1^2}$. Additionally, the parameter $k$ also needs to be adjusted as [6]

$$k = \sqrt{1 - (k')^2},$$

and $k' = (k_1')^N \prod_{i=1}^{L} sn^4 \left( u_i K_1', k_1' \right)$, where $L = \lfloor N/2 \rfloor$. 

Birkhäuser
5 Simulation Results

In this section, we present our numerical results of the proposed methods and compare them with the other graph filters. The ARMA graph filter in [5, 7], the LLS graph filter in [14], as well as the FIR Chebyshev graph filter in [19] are used as the benchmarks. Without special explanation, the passband cutoff frequency is set to $\lambda_p = 1$ and the stopband cutoff frequency is set to $\lambda_s = 1.2$. The maximum attenuation of the passband and minimum attenuation of the stopband are set to $R_p = 1$dB and $A_s = 30$dB, respectively.

In Fig. 5, we show the graph frequency responses versus the graph frequency of the proposed Butterworth high-pass graph filter, the proposed Chebyshev I high-pass graph filter, the proposed Chebyshev II high-pass elliptic graph filter, the ARMA high-pass graph filter, the LLS high-pass graph filter, and the FIR Chebyshev high-pass graph filter. Observe from Fig. 5b, d, f that our graph filters can obtain the required frequency responses, while ARMA graph filters fail in the stopband attenuation and LLS cannot achieve the desired transition passband and stopband attenuation. Additionally, in Fig. 5e, we can see that the needed order of the Chebyshev FIR graph filter is higher than those of our graph filters, and its fluctuations at cutoff frequency remain constant with an increasing order.

Figure 6 shows graph frequency responses versus graph frequency and graph magnitude responses versus graph frequency of the proposed Butterworth graph filter, the proposed Chebyshev I graph filter, the proposed Chebyshev II graph filter, and the
proposed elliptic graph filter, for $\lambda_p = 1, \lambda_s = 1.01, R_p=0.1$dB, and $A_s=40$dB. From Fig. 6, we can see that the proposed IIR graph filters can approximate ideal graph filters if the filter order is high enough. Observe from Fig. 6 that the proposed Butterworth graph filter, the proposed Chebyshev I graph filter, the proposed Chebyshev II graph filter, and the proposed elliptic graph filter are approximate ideal graph filters when the filter order is 910, 74, 74, and 18, respectively. The proposed elliptic graph filter outperforms the other three filters since it can realize the desired response with the minimum order.

Figure 7 shows the graph frequency responses versus the graph frequency of low-pass elliptic graph filter, band-pass elliptic graph filter, and band-stop elliptic graph filter. The parameters of three graph filters and those of their equivalent high-pass graph filters are shown in Table 1. Observe from Fig. 7 that the desired low-pass elliptic graph filter, band-pass elliptic graph filter, and band-stop elliptic graph filter are successfully obtained by their corresponding equivalent high-pass graph filter.
Fig. 7 Graph frequency responses versus the graph frequency of the low-pass, band-pass, and band-stop graph filter

6 Conclusion

In this paper, we use Butterworth, Chebyshev, and elliptic functions to design graph filters. We derive the corresponding zero-pole equations on graph frequency responses. By using these zeros and poles, we construct a couple of conjugate graph filters that achieve desired magnitude graph responses to design the universal Butterworth high-pass IIR graph filters, Chebyshev high-pass IIR graph filters, and elliptic high-pass IIR graph filters, respectively. We also propose to design the low-pass graph filter, band-pass graph filter, and band-stop graph filter by mapping the parameters to those of the equivalent high-pass graph filters. Furthermore, we propose to set the graph filter order given the corresponding requirements. Our simulation results show that the proposed graph filter design methods outperform the compared graph filter design methods in realizing desired frequency responses. Moreover, our methods can obtain arbitrary precision for step graph spectral responses. Additionally, the elliptic graph filter has the minimum order for the same given requirements.

Acknowledgements This work was supported by the National Natural Science Foundation of China under Grant 62071242.

Data Availability The datasets generated during and/or analyzed during the current study are available from the corresponding author on reasonable request.
Declarations

Conflict of interest  The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

1. S. Chen, A. Sandryhaila, J. Moura et al., Signal denoising on graphs via graph filtering, in 2014 IEEE Global Conference on Signal and Information Processing (GlobalSIP) (IEEE. 2014), pp. 872–876. https://doi.org/10.1109/GlobalSIP.2014.7032244
2. M. Cheung, J. Shi, O. Wright et al., Graph signal processing and deep learning: convolution, pooling, and topology. IEEE Signal Process. Mag. 37(6), 139–149 (2020). https://doi.org/10.1109/MSP.2020.3014594
3. M. Contino, E. Isufi, G. Leus, Distributed edge-variant graph filters, in 2017 IEEE 7th International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP) (IEEE. 2017), pp. 1–5. https://doi.org/10.1109/CAMSAP.2017.8313105
4. S. Deutsch, A. Ortega, G. Medioni, Manifold denoising based on spectral graph wavelets, in 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (IEEE 2016), pp. 4673–4677. https://doi.org/10.1109/ICASSP.2016.7472563
5. E. Isufi, A. Loukas, A. Simonetto et al., Autoregressive moving average graph filtering. IEEE Trans. Signal Process. 65(2), 274–288 (2017). https://doi.org/10.1109/TSP.2016.2614793
6. D. Lawden, Elliptic functions and applications, vol 80 (2013)
7. J. Liu, E. Isufi, L. Geert, Autoregressive moving average graph filtering. IEEE Trans. Signal Inf. Process. Netw. 5(1), 47–60 (2019). https://doi.org/10.1109/TISP.2018.2854627
8. A. Loukas, M. Cattani, M. Zuniga et al., Graph scale-space theory for distributed peak and pit identification. Association for Computing Machinery, New York, NY, USA, IPSN ’15, pp. 118–129 (2015). https://doi.org/10.1145/2737095.2737101
9. M. Lutovac, D. Toseic, B. Evans, Filter design for signal processing using matlab and mathematica (2000)
10. A. Miraki, H. Saeedi-Sourck, Spline graph filter bank with spectral sampling. Circ. Syst. Signal Process. 40, 5744–5758 (2021). https://doi.org/10.1007/s00034-021-01729-2
11. M. Onuki, S. Ono, M. Yamagishi et al., Graph signal denoising via trilateral filter on graph spectral domain. IEEE Trans. Signal Inf. Process. Netw. 2(2), 137–148 (2016). https://doi.org/10.1109/TISP.2016.2532464
12. O. Rimleanscaia, E. Isufi, Rational chebyshev graph filters, in 2020 54th Asilomar Conference on Signals, Systems, and Computers (ACSSC) (IEEE 2020), pp. 736–740. https://doi.org/10.1109/ACSSC51394.2020.9443317
13. A. Sandryhaila, J. Moura, Big data analysis with signal processing on graphs: Representation and processing of massive data sets with irregular structure. IEEE Signal Process. Mag. 31(5), 80–90 (2014). https://doi.org/10.1109/MSP.2014.2329213
14. A. Sandryhaila, J. Moura, Discrete signal processing on graphs: frequency analysis. IEEE Trans. Signal Process. 62(12), 3042–3054 (2014). https://doi.org/10.1109/TSP.2014.2321121
15. A. Sandryhaila, J. Moura, Discrete signal processing on graphs. IEEE Trans. Signal Process. 61(7), 1644–1656 (2013). https://doi.org/10.1109/TSP.2013.2238935
16. S. Segarra, A. Marques, A. Ribeiro, Optimal graph-filter design and applications to distributed linear network operators. IEEE Trans. Signal Process. 65(15), 4117–4131 (2017). https://doi.org/10.1109/TSP.2017.2703660
17. X. Shi, H. Feng, M. Zhai et al., Infinite impulse response graph filters in wireless sensor networks. IEEE Signal Process. Lett. 22(8), 1113–1117 (2015). https://doi.org/10.1109/LSP.2014.2387204
18. D. Shuman, S. Narang, P. Frossard et al., The emerging field of signal processing on graphs: extending high-dimensional data analysis to networks and other irregular domains. IEEE Signal Process. Mag. 30(3), 83–98 (2013). https://doi.org/10.1109/MSP.2012.2235192
19. D. Shuman, P. Vanderheyst, D. Kressner et al., Distributed signal processing via Chebyshev polynomial approximation. IEEE Trans. Signal Inf. Process. Netw. 4(4), 736–751 (2018). https://doi.org/10.1109/TISP.2018.2824239
20. T. Wang, H. Guo, B. Lyu et al., Speech signal processing on graphs: graph topology, graph frequency analysis and denoising. Chin. J. Electron. 29(5), 926–936 (2020). https://doi.org/10.1049/cje.2020.08.008

21. F. Wu, A. Souza, T. Zhang et al., Simplifying graph convolutional networks, in International conference on machine learning (PMLR) (2019), pp. 6861–6871. https://doi.org/10.1007/s40747-021-00567-8

22. Z. Wu, S. Pan, F. Chen et al., A comprehensive survey on graph neural networks. IEEE Trans. Neural Netw. Learn Syst. 32(1), 4–24 (2020). https://doi.org/10.1109/TNNLS.2020.2978386

23. X. Yan, Z. Yang, T. Wang et al., An iterative graph spectral subtraction method for speech enhancement. Speech Commun. 123, 35–42 (2020). https://doi.org/10.1016/j.specom.2020.06.005

**Publisher’s Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.