

Qucs

Report

Qucsactivefilter — Active filter synthesis subsystem of Qucs

Vadim Kuznetsov

Copyright © 2014 Vadim Kuznetsov <ra3xdh@gmail.com>

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.1 or any later version published by the Free Software Foundation. A copy of the license is included in the section entitled "GNU Free Documentation License".

Contents

1	Introduction	3
2	Filter transfer function	4
2.1	Frequency domain filter response	4
2.2	Transfer function general form	5
2.3	Poles and Zeros	6
2.4	Time domain parameters	6
2.5	Transfer function approximations	7
2.6	Physical active filter transfer function	8
2.7	Physical active filter poles and zeros	9
3	Low-pass filters transfer function approximations	11
3.1	Butterworth	11
3.1.1	Transfer function	11
3.1.2	Minimum order estimation	12
3.2	Chebyshev Type-I	12
3.2.1	Transfer function	12
3.2.2	Minimum order estimation	14
3.3	Chebyshev Type-II	14
3.3.1	Transfer function	14
3.3.2	Minimum order estimation	15
3.4	Cauer (Elliptic)	16
3.4.1	Transfer function and magnitude response	16
3.4.2	Minimum order estimation	16
3.4.3	Poles and zeros	17
3.5	Legendre	19
3.6	Bessel	20
4	Other filter types	22
4.1	High-pass filters	22
4.2	Band-pass filters	22

4.3	Band-stop filters	24
5	Filter topologies	25
5.1	Active filter schematic synthesis algorithm	25
5.2	First-order section	26
5.2.1	Low pass filter	26
5.2.2	High pass filters	27
5.3	Sallen-Key	28
5.3.1	Low pass filter	28
5.3.2	High pass filters	29
5.3.3	Band pass filter	30
5.4	Multifeedback	33
5.4.1	Low pass filter	33
5.4.2	High pass filters	34
5.4.3	Band pass filter	35
5.5	Cauer and Chebyshev Type-II filters	35
5.5.1	Low pass filter	35
5.5.2	High pass filters	37
5.5.3	Band pass filters	37
5.6	Band stop filters	39
6	Conclusion	41

Chapter 1

Introduction

The purpose of this report is description of background mathematics used in Qucsactivefilter.

Qucsactivefilter is powerful active filter design tool. It could be called from *Tools->Active Filters* menu. It allows you to build active filter circuit and simulate it with Qucs. Qucsactivefilter builds active filters circuits based on RC-components and operational amplifier (opamp).

It is need to define following four groups of parameters to calculate active filter:

1. Frequency response approximation type. Butterworth, Chebyshev, Inverse Chebyshev, Cauer (Elliptic) and Bessel filters are available.
2. Frequency response parameters: filter gain and bandwidth.
3. Filter topology. Sallen-Key, Mutifeedback (MFB) and Cauer topologies are available.
4. Filter type. Low-pass, high-pass, band-pass and band-stop filters are available.

Filter synthesis method used by Qucsactivefilter is based on filter transfer function poles and zeros analysis in frequency domain.

Chapter 2

Filter transfer function

2.1 Frequency domain filter response

There are 4 main types frequency domain filter responses:

1. Low-pass filter (LPF)
2. High-pass filter (HPF)
3. Band-pass filter (BPF)
4. Band-stop filter (BSF)

Magnitude responses $|H(j\omega)|$ of ideal filters are shown in the Figure 2.1.

By ω_c denote the *cutoff frequency* of the filter. LPF passes all frequencies below ω_c and rejects all frequencies upper ω_c . HPF operates contrariwise. Magnitude responses of ideal filters have rectangular form. Magnitude responses of physical filters have smoothed curves form. Frequency response of HPF, BPF and BSF could be normalized to low-pass prototype. For this reason we consider low-pass active filter further. It will be shown how to transform HPF, BPF, and BSF to low-pass prototype filter.

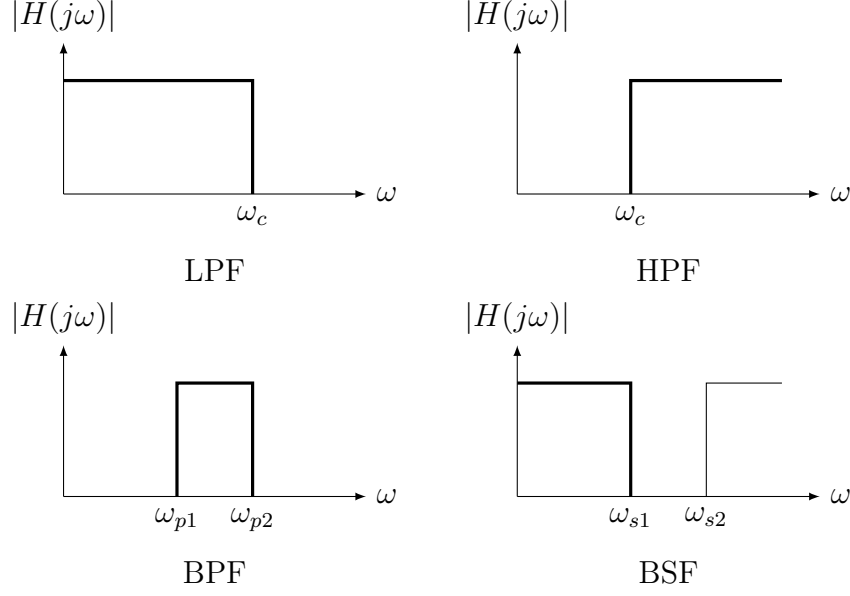


Figure 2.1: Magnitude responses of ideal filters

2.2 Transfer function general form

Active filters are characterized by transfer function in frequency domain. Common form of the filter transfer function is given here:

$$H(s) = \frac{b_m s^m + b_{m-1} s^{m-1} + \dots + b_2 s^2 + b_1 s + b_0}{a_n s^n + a_{n-1} s^{n-1} + \dots + a_2 s^2 + a_1 s + a_0} \quad (2.1)$$

The filter order N is:

$$N = \max(m, n) \quad (2.2)$$

Filter order determines the number of filter sections and filter circuit complexity. Active filter consists of $k = \lfloor N/2 \rfloor$ 2-nd order section and $k = \lceil N \bmod 2 \rceil$ 1-st order sections.

Zeros of the transfer function are the roots of numerator. Poles are the roots of denominator. We need to know filter transfer function to determine components (resistors and capacitors — RC) values of the active filter circuit.

We can obtain magnitude $A(\omega)$ and phase $\theta(\omega)$ responses using transfer function:

$$A(\omega) = |H(j\omega)| \quad (2.3)$$

$$\theta(\omega) = \arg(H(j\omega)) \quad (2.4)$$

Qucsactivefilter uses filter design algorithms provided by [1].

2.3 Poles and Zeros

Transfer function could be represented as a ratio of numerator $P(s)$ and denominator $Q(s)$ polynomials. Each of these polynomials could be factorized:

$$H(s) = \frac{P(s)}{Q(s)} = H_0 \frac{(s - z_1)(s - z_2)(s - z_3) \dots (s - z_m)}{(s - p_1)(s - p_2)(s - p_3) \dots (s - p_n)} \quad (2.5)$$

The roots of the numerator $z_1, z_2, z_3, \dots, z_m$ are called zeros of the transfer function. The roots of the denominator $p_1, p_2, p_3, \dots, p_n$ are called poles of the transfer function. The n -th order polynomial has n roots. H_0 is DC gain of the filter.

If $P(s)$ and $Q(s)$ are polynomials with real coefficients, each pole or zero has its complex-conjugated pair. For example for n -th order:

$$p_i = p_{n-i} = \sigma_{pi} \pm j\omega_{pi} \quad (2.6)$$

$$z_i = z_{n-i} = \sigma_{zi} \pm j\omega_{zi} \quad (2.7)$$

σ_i is real part of the pole or zero:

$$\sigma_{pi} = \Re[p_i] \quad \omega_{pi} = \Im[p_i] \quad (2.8)$$

ω_i is imaginary part of the pole or zero:

$$\sigma_{zi} = \Re[z_i] \quad \omega_{zi} = \Im[z_i] \quad (2.9)$$

2.4 Time domain parameters

The impulse response $h(t)$ is the filter output signal when *Dirac delta impulse* is applied to its input. Impulse response is inverse Laplace transform of the filter transfer function:

$$h(t) = \mathcal{L}^{-1}[H(s)] \quad (2.10)$$

The step response $g(t)$ is the filter output signal when *unit step* is applied to filter input. Step response could be expressed via inverse Laplace transform:

$$g(t) = \mathcal{L}^{-1} \left[\frac{H(s)}{s} \right] \quad (2.11)$$

Impulse response is derivative of step response

$$h(t) = \frac{d}{dt}g(t) \quad (2.12)$$

The *phase delay* $\tau_p(\omega)$ of a system is defined using phase response $\theta(\omega)$

$$\tau_p(\omega) = \frac{-\theta(\omega)}{\omega} \quad (2.13)$$

Phase delay is time delay of the sinusoidal signal of frequency ω passing through the filter.

The *group delay* is defined as

$$\tau_g(\omega) = -\frac{d}{d\omega}\theta(\omega) \quad (2.14)$$

The group delay is the measure of modulated signal distortion. Group delay is important for high-quality audio signals amplification.

2.5 Transfer function approximations

Magnitude and phase response type depends on transfer function numerator and denominator polynomials coefficients a_i and b_i . Substituting different sets of a_i and b_i we can implement different filters: low-pass, high-pass, band-pass and band-stop. The following polynomials are the most frequently used for the active filters design purposes :

1. Butterworth
2. Chebyshev — Type I
3. Chebyshev — Type II (Inverse Chebyshev)
4. Cauer
5. Bessel
6. Legendre

Polynomials coefficients are calculated using filter approximation. Every transfer function approximation has its own set of poles and zeros. It is need to note that Butterworth, Chebyshev Type I and Bessel filters have no zeros.

`Qucsactivefilter` evaluates poles and zeros for given approximation and then evaluates RC-elements values for each section of active filter.

You can define a_i and b_i coefficients of the transfer function (2.1) manually with `Qucsactivefilter`. This method is suitable for unknown or new approximation. Then `Qucsactivefilter` evaluates poles and zeros and builds filter circuit. See `Filter::calcUserTrFunc()` in `filter.cpp` for details.

2.6 Physical active filter transfer function

Physical active filter of N -th order consists of $N_2 = \lfloor N/2 \rfloor$ 2-nd order sections and $N \bmod 2$ 1-st order sections. Physical filter consists of

$$N_{sec} = \lfloor N/2 \rfloor + N \bmod 2 \quad (2.15)$$

total 2-nd order and 1-st order sections.

So, transfer function can be represented as product of the each filter section transfer functions.

For i -th 2-nd order section which have transfer function zeros (Cauer and Chebyshev Type II) we have following section transfer function:

$$H_2(s) = H_0^1 \frac{s^2 + A_i \omega_c^2}{s^2 + B_i \omega_c s + C_i \omega_c^2} \quad (2.16)$$

And for i -th 2-nd order section without zeros (Butterworth, Chebyshev Type I) and Bessel:

$$H_2(s) = H_0^1 \frac{C_i \omega_c^2}{s^2 + B_i \omega_c s + C_i \omega_c^2} \quad (2.17)$$

For $N - th$ 1-st order section:

$$H_1(s) = \frac{H_0^1}{s + C_N \omega_c} \quad (2.18)$$

where A, B, C — are determined by poles and zeros location; ω_c is filter cutoff frequency.

H_0^1 is DC gain of the i -th section:

$$H_0^1 = (H_0)^{1/N_{sec}} \quad (2.19)$$

Let's consider normalized form of the filter frequency response. For normalized frequency response we assume $\omega_c = 1$.

Common form of the filter transfer function could be factorized as below. For odd order filters without transfer function zeros (Butterworth, Chebyshev Type-I and Bessel):

$$H(s) = H_1(s) \prod_{i=0}^{N_2} H_2(s) = H_0 \frac{1}{s + C_N} \prod_{i=0}^{N_2} \frac{C_i}{s^2 + B_i s + C_i} \quad (2.20)$$

For odd order filter with transfer function zeros (Cauer and Chebyshev-Type-II)

$$H(s) = H_1(s) \prod_{i=0}^{N_2} H_2(s) = H_0 \frac{1}{s + C_N} \prod_{i=0}^{N_2} \frac{s^2 + A_i}{s^2 + B_i s + C_i} \quad (2.21)$$

For even order filter without transfer function zeros:

$$H(s) = \prod_{i=0}^{N_2} H_2(s) = H_0 \prod_{i=0}^{N_2} \frac{C_i}{s^2 + B_i s + C_i} \quad (2.22)$$

For even order filter with transfer function zeros:

$$H(s) = \prod_{i=0}^{N_2} H_2(s) = H_0 \prod_{i=0}^{N_2} \frac{s^2 + A_i}{s^2 + B_i s + C_i} \quad (2.23)$$

Every 2-nd order factor matches one pair of complex conjugated poles and one pair of complex conjugated zeros. First-order factor matches one real pole.

Filter synthesis method proposed in this paper uses A_i , B_i , C_i coefficients to evaluate active filter RC-elements values. We should find A_i , B_i , C_i coefficients using poles and zeros location to find RC-elements values.

2.7 Physical active filter poles and zeros

Even N -th order physical active filter transfer function has no zeros or $N/2$ complex conjugated pairs of zeros and $N/2$ complex conjugated pairs of poles. Odd order physical active filter transfer function has additional real pole.

We need to solve numerator of i -th factor (2.16) equals zeros to find i -th zero pair location:

$$s^2 + A_i = 0 \quad (2.24)$$

$$z_{i,N-i} = \pm j\sqrt{A_i} \quad (2.25)$$

And we can find coefficient A_i by known i -th zero pair imaginary part 2.7:

$$z_{i,N-i} = \pm j\omega_{zi} \quad (2.26)$$

$$A_i = \omega_{zi}^2 \quad (2.27)$$

Zeros of the physical active filter transfer function have no real part. Using this equation (2.27) we can find A_i coefficients by known zeros location. Zeros location (and real and imaginary part) could be found using transfer function approximation (Butterworth, Chebyshev, etc.).

We need to solve the following quadratic equation to find i -th pole location (denominator of i -th 2-nd order factor equals zero):

$$s^2 + B_i s + C_i = 0 \quad (2.28)$$

Solution of this equation yields complex conjugated pair of poles:

$$p_{i,N-i} = \sigma_{pi} \pm j\omega_{pi} = -\frac{B_i}{2} \pm \frac{\sqrt{-B_i^2 + 4C_i}}{2} \quad (2.29)$$

We need to solve the next system of equations find B_i and C_i by known poles location:

$$\omega_{pi} = \frac{\sqrt{-B_i^2 + 4C_i}}{2} \quad (2.30)$$

$$\sigma_{pi} = -\frac{B_i}{2} \quad (2.31)$$

Solution of this system yields:

$$B_i = -2\sigma_{pi} \quad (2.32)$$

$$C_i = \sigma_{pi}^2 + \omega_{pi}^2 \quad (2.33)$$

Odd order N -th filters have one N -th real pole $p_N = \sigma_N \pm j \cdot 0$. To find this pole we need to solve the following equation (denominator of the (2.18) equals zero):

$$s + C_N = 0 \quad (2.34)$$

Solution gives:

$$p_N = \sigma_N = -C_N \quad (2.35)$$

And we can find C_N coefficient by known real part of N -th real pole:

$$C_N = -\sigma_N \quad (2.36)$$

Using poles and zeros location and equation (2.27), (2.32) and (2.33) we can find A_i , B_i , C_i coefficients, factorize transfer function to (2.21) form and find filter RC-elements values.

Chapter 3

Low-pass filters transfer function approximations

3.1 Butterworth

3.1.1 Transfer function

Butterworth filter implements magnitude response as flat as possible in pass band. Magnitude response monolitically decays outside passband.

Normalized magnitude response of N -th order the Butterworth filter has the form:

$$|H(j\omega)| = \sqrt{\frac{1}{1 + (\omega/\omega_c)^{2N}}} \quad (3.1)$$

Transfer function of N -th order Butterworth filter could be factorized as following:

$$H(s) = \frac{1}{\prod_{i=1}^N (s - p_i)} = \frac{1}{(s - p_1)(s - p_2) \dots (s - p_N)} \quad (3.2)$$

where p_i are transfer function poles. Butterworth filter has no zeros. Poles location could be determined as following [2]:

$$p_i = e^{j\pi[(2i+N-1)/2N]} = \cos\left(\frac{2i + N - 1}{2N}\right) + j \sin\left(\frac{2i + N - 1}{2N}\right) \quad (3.3)$$

Filter::calcButterworth() function implements such poles location evaluation according equation (3.3). For sources see `filter.cpp`.

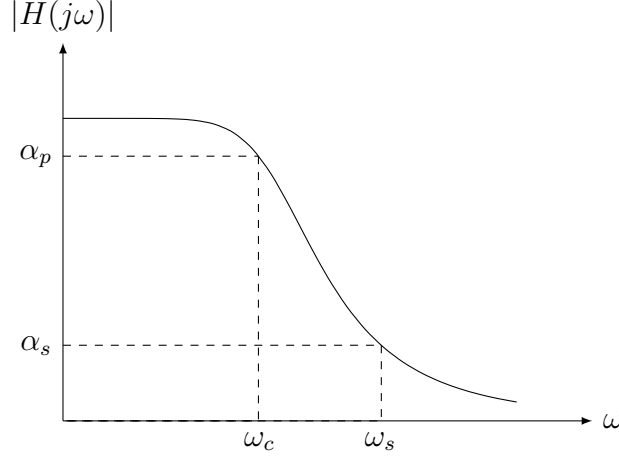


Figure 3.1: Typical 4-th order Butterworth filter magnitude response

3.1.2 Minimum order estimation

We need to know maximum passband attenuation A_p and minimum stopband attenuation A_s to estimate minimum Butterworth order. These attenuations often are measured in decibels (dB).

These values are determined by absolute attenuation α_p at cutoff frequency ω_c and attenuation α_s at stopband frequency ω_s .

$$A_p = 20 \log \alpha_p \quad (3.4)$$

$$A_s = 20 \log \alpha_s \quad (3.5)$$

The next equation determines the minimum order of the Butterworth filter.

$$N = \frac{1}{2 \log(\omega_s/\omega_c)} \cdot \frac{\log(10^{0.1A_s} - 1)}{\log(10^{0.1A_p} - 1)} \quad (3.6)$$

This filter has A_p attenuation value at cutoff frequency ω_c and at least A_s attenuation at stopband frequency ω_s .

3.2 Chebyshev Type-I

3.2.1 Transfer function

Chebyshev filters provide more sharp transition from pass band to stop band than Butterworth filter. Passband magnitude response of the Chebyshev filter is not flat. It has ripple in passband R_p up to 3dB.

Magnitude response of the N – th order Chebyshev Type-I filter is determined by equation:

$$H(j\omega) = \frac{1}{\sqrt{1 + \varepsilon^2 T_N^2(\omega/\omega_c)}} \quad (3.7)$$

where $T_N^2(\omega)$ is N -th order Chebyshev polynomial.

$$T_N^2(x) = \cos(N \arccos(x)) \quad (3.8)$$

ε is the ripple coefficient:

$$\varepsilon = \sqrt{10^{0.1R_p} - 1} \quad (3.9)$$

For $R_p=3\text{dB}$ we obtain $\varepsilon = 1$. For $R_p=0\text{dB}$ (no ripple) we obtain $\varepsilon = 0$.

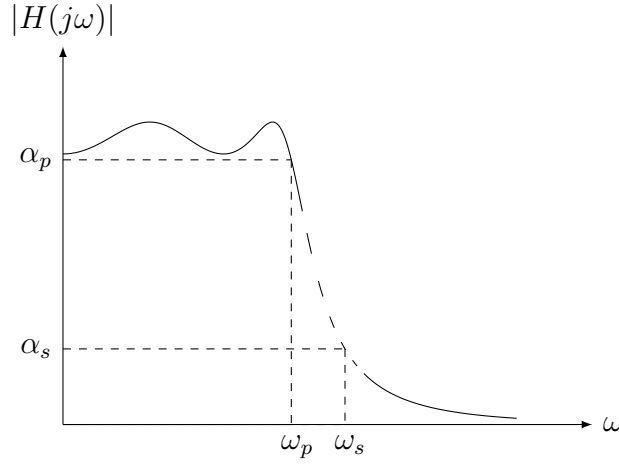


Figure 3.2: Typical 4-th order Chebyshev filter magnitude response

The poles p_i of the Chebyshev Type-I filter could be evaluated as following [2]. The poles are the roots of the N -th order Chebyshev polynomial:

$$p_i = \sigma_i + j\omega_i \quad (3.10)$$

$$\sigma_i = -\sin\left(\frac{(2i-1)\pi}{2N}\right) \sinh\left[\frac{1}{N} \sinh^{-1}\left(\frac{1}{\varepsilon}\right)\right] \quad (3.11)$$

$$\omega_i = \cos\left(\frac{(2i-1)\pi}{2N}\right) \cosh\left[\frac{1}{N} \sinh^{-1}\left(\frac{1}{\varepsilon}\right)\right] \quad (3.12)$$

Chebyshev filters have no zeros. We should know filter order N and acceptable bandpass ripple R_p to find filter poles. `Filter::calcChebyshev` function evaluates Chebyshev filter poles using equations (3.11) and (3.12) (see `filter.cpp`).

3.2.2 Minimum order estimation

The minimum order N of the Chebyshev filter can be estimated using following equation. We assume $A_p = R_p$.

$$N = \frac{\cosh^{-1}(\sqrt{(10^{0.1A_s} - 1)/\varepsilon^2})}{\cosh^{-1}(\omega_s/\omega_c)} \quad (3.13)$$

where ε could be evaluated from equation (3.9). It's need to know stopband attenuation A_s (dB), passband ripple R_p (dB), cutoff frequency ω_c , and stopband frequency ω_s . `Filter::calcChebyshev()` function performs such estimation. See `filter.cpp` for source code.

Chebyshev Type-I and Butterworth filters are the most frequently used ones.

3.3 Chebyshev Type-II

3.3.1 Transfer function

Magnitude response of the N -th order Chebyshev Type-I (or inverse Chebyshev) filter is determined by equation:

$$H(j\omega) = \frac{1}{\sqrt{1 + \frac{1}{\varepsilon^2 T_N^2(\omega_c/\omega)}}} \quad (3.14)$$

where $T_N(\omega)$ is Chebyshev polynomial (3.8).

Stopband ripple R_s is determined by ε

$$\varepsilon = \frac{1}{\sqrt{10^{0.1R_s} - 1}} \quad (3.15)$$

Typical magnitude response of the 4-th order Chebyshev Type-II filter is shown in the Figure 3.3.

You can see from this figure that this filter has flat response at passband and ripple at stopband.

N -th order Chebyshev Type-II filter has N imaginary zeros. The location of the i -th zero z_i could be estimated using following equations:

$$z_i = j\omega_{zi} \quad (3.16)$$

$$\omega_{zi} = -\frac{1}{\cos\left(\frac{(2i-1)\pi}{2N}\right)} \quad (3.17)$$

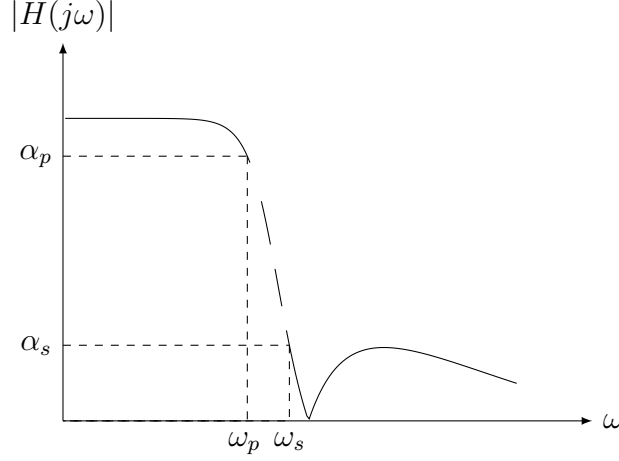


Figure 3.3: Typical 4-th order Chebyshev Type-II filter magnitude response

Also N -th order Chebyshev Type-II filter has N imaginary poles. The location of the i -th pole p_i is determined by following equations. The poles are inverse to poles of the Chebyshev Type-I filter. And Chebyshev Type-II filters are known as inverse Chebyshev filters.

$$p_i = \frac{1}{\sigma_{pi} + j\omega_{pi}} \quad (3.18)$$

$$\sigma_{pi} = -\sin\left(\frac{(2i-1)\pi}{2N}\right) \sinh\left[\frac{1}{N} \sinh^{-1}\left(\frac{1}{\varepsilon}\right)\right] \quad (3.19)$$

$$\omega_{pi} = \cos\left(\frac{(2i-1)\pi}{2N}\right) \cosh\left[\frac{1}{N} \sinh^{-1}\left(\frac{1}{\varepsilon}\right)\right] \quad (3.20)$$

3.3.2 Minimum order estimation

The minimum order N of the Chebyshev filter can be estimated using following equation. We assume $A_s = R_s$.

$$N = \frac{\cosh^{-1}(\sqrt{(10^{0.1A_s} - 1)})}{\cosh^{-1}(\omega_s/\omega_c)} \quad (3.21)$$

It's need to know stopband attenuation A_s (dB), cutoff frequency ω_c , and stopband frequency ω_s . `Filter::calcInvChebyshev()` function performs such estimation. See `filter.cpp` for source code.

3.4 Cauer (Elliptic)

3.4.1 Transfer function and magnitude response

Cauer or Elliptic filters have ripple in pass band and in stop band. These filters have the sharpest magnitude frequency response. Magnitude response is determined by following equation:

$$|H(j\omega)| = \frac{1}{\sqrt{1 + \varepsilon^2 R_N^2(\omega/\omega_c, L)}} \quad (3.22)$$

where $R_N(\omega, L)$ is N -th order elliptic rational function with ripple parameter L . Typical magnitude response is shown in the Figure 3.4.

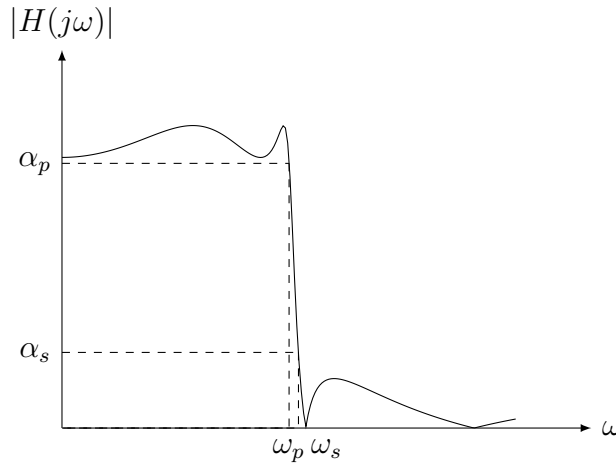


Figure 3.4: Typical 4-th order Cauer(Elliptic) filter magnitude response

You can see from this figure that Cauer filter has the most sharpest magnitude response. But this filter has both bandpass ripple and bandstop ripple.

3.4.2 Minimum order estimation

We need to know cutoff frequency ω_c , stopband frequency ω_s , passband ripple R_p , stopband attenuation A_s . We assume $A_p = R_p$ and $R_s = A_s$. Minimum order estimation method follows Handbook [3], page 95.

Minimum filter order could be estimated using following steps:

1. Determine selectivity factor k

$$k = \omega_c/\omega_s \quad (3.23)$$

2. Compute the modular constant q

$$q = u + 2u^5 + 15u^9 + 150u^{13} \quad (3.24)$$

$$u = \frac{1 - \sqrt[4]{1 - k^2}}{2(1 + \sqrt[4]{1 - k^2})} \quad (3.25)$$

3. Compute the discrimination factor D

$$D = \frac{10^{0.1A_s} - 1}{10^{0.1A_p - 1}} \quad (3.26)$$

4. Estimate the minimum required order of the Cauer filter

$$N = \left\lceil \frac{\log 16D}{\log(1/q)} \right\rceil \quad (3.27)$$

`Filter::cauerOrderEstim()` function implements this algorithm. See `filter.cpp`.

3.4.3 Poles and zeros

Elliptic rational functions are expressed via Jacobi elliptic cosine functions. By this reason you should use polynomial approximations to find poles and zeros of the transfer function. Qucsactivefilter uses algorithm based on *Digital Filter Designer's Handbook* [3].

This algorithm consists of three steps:

1. Estimate the minimum order of the Cauer filter using magnitude response parameters
2. Evaluate coefficients A_i , B_i , C_i for transfer function factorization (2.21) using algorithm from pages 95 – 97 of the Handbook [3].
3. Find poles and zeros as roots of quadratic equation using equations (2.29) and (2.27). You also need to find N -th real pole p_N for odd filter order.

Use following steps to calculate A_i , B_i , C_i coefficient for every section of the N -th order Cauer filter.

1. Determine selectivity factor k (3.23) and modular constant q (3.24)
2. Compute V as

$$V = \frac{1}{2N} \ln \left(\frac{10^{A_p/20} + 1}{10^{A_p/20} - 1} \right) \quad (3.28)$$

3. Compute p_0 as

$$p_0 = \left| \frac{q^{1/4} \sum_{m=0}^{\infty} (-1)^m q^{m(m+1)} \sinh[(2m+1)V]}{0.5 + \sum_{m=0}^{\infty} (-1)^m q^{m^2} \cosh(2mV)} \right| \quad (3.29)$$

4. Compute W as

$$W = \left[\left(1 + \frac{p_0^2}{k} \right) (1 + kp_0^2) \right]^{1/2} \quad (3.30)$$

5. Determine the number r of 2-nd order sections

$$r = \lfloor \frac{N}{2} \rfloor \quad (3.31)$$

6. For each i -th 2-nd order section $i = 1, 2, \dots, r$ compute X_i as

$$X_i = \frac{2q^{1/4} \sum_{m=0}^{\infty} (-1)^m q^{m(m+1)} \sin[(2m+1)\mu\pi V/N]}{1 + 2 \sum_{m=0}^{\infty} (-1)^m q^{m^2} \cos(2m\mu\pi/N)} \quad (3.32)$$

where

$$\mu = \begin{cases} i, & N \text{ odd} \\ i - 1/2, & N \text{ even} \end{cases} \quad (3.33)$$

7. For each i -th 2-nd order section $i = 1, 2, \dots, r$ compute Y_i as

$$Y_i = \left[\left(1 - \frac{X_i^2}{k} \right) (1 - kX_i^2) \right] \quad (3.34)$$

8. Compute coefficients A_i, B_i, C_i using W, X_i, Y_i

$$A_i = \frac{1}{X_i^2} \quad (3.35)$$

$$B_i = \frac{2p_0 Y_i}{1 + p_0^2 X_i^2} \quad (3.36)$$

$$C_i = \frac{(p_0 Y_i)^2 + (X_i W)^2}{(1 + p_0^2 X_i^2)^2} \quad (3.37)$$

9. For odd filter order compute N -th real pole p_N

$$p_N = -p_0 \quad (3.38)$$

`Filter::cauerOrderEstim()` and `Filter::calcCauer()` implement this algorithm. These functions are called together. See `filter.cpp`.

3.5 Legendre

Legendre filter (or "L" optimum filter) provides the maximum possible magnitude response rolloff and has no ripple in passband. The magnitude response of Legendre filter for unity cutoff frequency is defined as following:

$$|H(j\omega)| = \frac{1}{\sqrt{1 + L_n(\omega^2)}} \quad (3.39)$$

where $L_n(\omega^2)$ is n -th order optimal polynomial, which is derived from Legendre polynomials [4]. The first 10 Legendre polynomials are shown in the Table 3.1. An example of typical magnitude response for Legendre filter is shown in the Figure 3.5.

Table 3.1: The first "L" polynomials

n	$L_n(\omega^2)$
1	ω^2
2	ω^4
3	$\omega^2 - 3\omega^4 + 3\omega^6$
4	$3\omega^4 - 8\omega^6 + 6\omega^8$
5	$\omega^2 - 8\omega^4 + 28\omega^6 - 40\omega^8 + 20\omega^{10}$
6	$6\omega^4 - 40\omega^6 + 105\omega^8 - 120\omega^{10} + 50\omega^{12}$
7	$\omega^2 - 15\omega^4 + 105\omega^6 - 355\omega^8 + 615\omega^{10} + 525\omega^{12} + 175\omega^{14}$
8	$10\omega^4 - 120\omega^6 + 615\omega^8 - 1624\omega^{10} + 2310\omega^{12} - 1680\omega^{14} + 490\omega^{16}$
9	$\omega^2 - 24\omega^4 + 276\omega^6 - 1624\omega^8 + 5376\omega^{10} - 10416\omega^{12} + 11704\omega^{14} - 7056\omega^{16} + 1764\omega^{18}$
10	$15\omega^4 - 280\omega^6 + 2310\omega^8 - 10416\omega^{10} + 27860\omega^{12} - 45360\omega^{14} + 44100\omega^{16} - 23520\omega^{18} + 5292\omega^{20}$

The following sequence is used to synthesize Legendre filter in Qucsactivefilter application. Input parameters are cutoff frequency f_c and filter order n .

1. Select appropriate Legendre polynomial from the table 3.1
2. Obtain squared magnitude response $H^2(\omega)$. For example, for 4-th order filter:

$$H(\omega) = \frac{1}{1 + \omega^2 - 3\omega^4 + 3\omega^6} \quad (3.40)$$

3. Substitute $-s^2 = \omega^2$. For 4-th order filter we obtain:

$$H(s^2) = \frac{1}{1 - s^2 - 3s^4 - 3s^6} \quad (3.41)$$

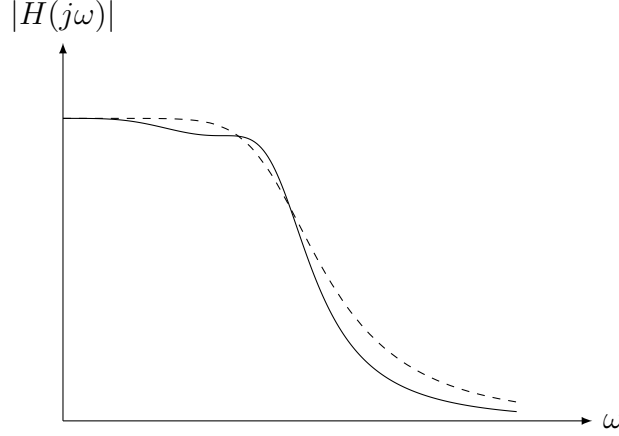


Figure 3.5: Typical 4-th Legendre optimum filter magnitude response (solid curve) compared to Butterworth 4-th order filter magnitude response (dashed curve)

4. Calculate roots of denominator. Roots of denominator are the poles of transfer function.
5. Synthesize filter circuitry by known transfer function poles.

Qucsactivefilter application uses predefined poles table of Legendre filter. The `legendre.h` header file contains poles table. This table is computed in Octave using steps 1 – 4 of the algorithm. Qucsactivefilter application takes pole value from the table and perform step 5 to obtain filter circuitry.

3.6 Bessel

For all considered filter transfer approximations magnitude response was considered. Phase response was not taken into account. Bessel filter belongs to filters with normalized phase delay τ_p (2.13). Phase response of such filter is linear-dependent on frequency

$$\theta(\omega) = -\omega\tau_p \quad (3.42)$$

Phase response of Bessel filters approaches to such dependency in some frequency range.

Transfer function of the Bessel filter has following form:

$$H(s) = \frac{\theta_n(0)}{\theta_n(s/\omega_c)} \quad (3.43)$$

where $\theta_n(s)$ is n -th order reverse Bessel polynomial

$$\theta_n(s) = \sum_{k=0}^n a_k s^k \quad (3.44)$$

$$a_k = \frac{(2n-k)!}{2^{n-k} k! (n-k)!} \quad k = 0, 1, \dots, n \quad (3.45)$$

For example for 5-th order Bessel polynomial:

$$H(s) = \frac{945}{s^5 + 15s^4 + 105s^3 + 420s^2 + 945s + 945} \quad (3.46)$$

Poles of Bessel transfer function could not be evaluated symbolically. Qucsactive-filter uses precalculated poles tables for Bessel filters up to 20-th order. See header **bessel.h** and Octave script **bessel-poles.m**

Bessel filter transfer function has no zeros.

Minimal order of the Bessel filter could not be evaluated too. You should define it manually. See **Filter::calcBessel()** and **filter.cpp** for source code.

Chapter 4

Other filter types

4.1 High-pass filters

High-pass filters calculation uses low-pass filter prototype. Then you can use low-pass prototype to determine filter order and poles/zeros.

High-pass filter transfer function $H(s)$ could be mapped to low-pass prototype filter transfer function $H_{LPF}(s')$. The following transform should be used:

$$H(s) = H_{LPF}(s') \quad (4.1)$$

$$s = \frac{1}{s'} \quad (4.2)$$

For cutoff frequency and stopband frequency the following transformations are valid:

$$\omega'_c = \frac{1}{\omega_c} \quad (4.3)$$

$$\omega'_s = \frac{1}{\omega_s} \quad (4.4)$$

We can map cutoff and stopband frequencies of high-pass filter to cutoff and stopband frequencies of low-pass prototype using these equations. Then we can use common method of low-pass filters synthesis. We can obtain poles and zeros of transfer function and determine RC-elements values.

4.2 Band-pass filters

Band-pass filter requires another transformations. By Ω_0 denote the center frequency. By $\Delta\Omega$ denote the bandwidth.

$$\Omega_0 = \sqrt{\omega_{p1}\omega_{p2}} \quad (4.5)$$

$$\Delta\Omega = \omega_{p2} - \omega_{p1} \quad (4.6)$$

where ω_{p1} is lower cutoff frequency; ω_{p2} is upper cutoff frequency. Denote by TW transient bandwidth. Upper ω_{s2} and lower ω_{s1} cutoff frequencies are determined as following:

$$\omega_{s2} = \omega_{p2} - TW \quad (4.7)$$

$$\omega_{s1} = \omega_{p1} + TW \quad (4.8)$$

Quality factor of band-pass filter is:

$$Q = \frac{\Omega_c}{\Delta\Omega} \quad (4.9)$$

We get band-pass filter transfer function:

$$H(s) = H_{LPF}(s') \quad (4.10)$$

$$s' = s + \frac{\Omega_0^2}{s} \quad (4.11)$$

Low-pass prototype cutoff frequency ω'_c and stopband frequency ω'_s yield:

$$\omega'_c = \Delta\Omega \quad (4.12)$$

$$\omega'_s = \min(|\omega'_{s1}|, |\omega'_{s2}|) \quad (4.13)$$

$$\omega'_{s1} = \omega_{s1} - \frac{\Omega_0^2}{\omega_{s1}} \quad (4.14)$$

$$\omega'_{s2} = \omega_{s2} - \frac{\Omega_0^2}{\omega_{s2}} \quad (4.15)$$

Using these equation we can obtain low-pass filter cutoff and stopband frequencies. Magnitude response ripple parameters are the same. Then we can obtain filter order and poles/zeros.

Two 2-nd order sections match one complex conjugated pole/zero pair of low-pass prototype. Band-pass filter has always even order.

4.3 Band-stop filters

Band stop filters transfer function could be transformed into low-pass filter transfer function using the presented substitution.

$$H(s) = H_{LPF}(s') \quad (4.16)$$

$$s' = \frac{1}{s + \frac{\Omega_0^2}{s}} \quad (4.17)$$

Center frequencies Ω_0 and bandwidth $\Delta\Omega$ are the same as for band-pass filter. Cutoff frequency and bandwidth of the low-pass prototype could be estimated using approach from the previous section.

Chapter 5

Filter topologies

5.1 Active filter schematic synthesis algorithm

Active filter synthesis algorithm contains the following steps. For low-pass and high-pass filters you can use this algorithm directly without any adaptations. For band-pass and band-stop filters you should evaluate low-pass prototype cutoff frequency and stopband frequency first. `Filter::calcFilter()` function implements these steps. See `filter.cpp` for sources. Additional information about used filter topologies are provided in [1, 5, 6]

- **Step 1:** Select desired filter type (low-pass, high-pass, band-pass, band-stop), filter topology, frequency response approximation type and following frequency response parameters:
 1. Cutoff frequency
 2. Stopband frequency
 3. Passband attenuation
 4. Stopband attenuation
 5. Passband ripple (for Chebyshev and Cauer filters only)
 6. Passband gain
- **Step 1a:** For all filters except low-pass. Determine cutoff frequency, and stopband frequency for low-pass prototype using methods from section 4.
- **Step 2:** Estimate filter order N using equation (3.6) for Butterworth filters, equation (3.13) for Chebyshev Type-I filters, equation (3.21) for Chebyshev Type-II filter, equation (3.27) for Cauer filter.

- **Step 3:** Evaluate poles and zeros of the transfer function using order value from previous step. Qucsactivefilter stores poles and zeros as complex numbers using `std::complex` class. You can access human-readable poles and zeros list using `Filter::createPolesZerosList()`.
- **Step 4:** Calculate number of 2-nd order sections and 1-st order sections using equation (2.15). Even order filters contain only 2-nd order sections. Odd order filters contain one 1-st order section too.
- **Step 5:** Evaluate A_i , B_i , C_i coefficients for each 2-nd order section using poles/zeros complex conjugated pairs obtained at step 3. For Chebyshev Type-II and Causer filters use equation (2.27), (2.32), and (2.33). Butterworth, Chebyshev Type-I, and Bessel filters have no zeros. You should calculate only B_i and C_i for these filters. Use equations (2.32) and (2.33) for this purpose.
- **Step 5a:** For odd order filters only evaluate first order section transfer function coefficient C_N using equation (2.36).
- **Step 6:** Select desired filter topology and determine RC-elements values for each 2-nd order filter section. You can use any of Sallen-Key (S-K) and Multifeedback(MFB) topologies for Butterworth, Chebyshev Type-I, and Bessel filters. For Causer and Chebyshev Type-II you should use only special Causer filter topology. RC-elements calculation algorithm is based on A_i , B_i , C_i coefficients. These coefficients are obtained at previous step.
- **Step 6a:** For odd order filters only determine RC-elements values of the 1-st order section.

We obtain active filter RC-elements values list after these steps are performed. Qucsactivefilter assumes ideal opamps for active filters. Now we can build filter circuit. Qucsactivefilter builds filter circuit automatically. You can simple copy-paste it into Qucs. Also you can access human-readable RC-elements list via `Filter::createPartList()`.

The next sections contain description of active filters circuitry. Such circuitry is used in Qucsactivefilter.

5.2 First-order section

5.2.1 Low pass filter

First order section of low-pass filters circuit is shown in the Figure 5.1

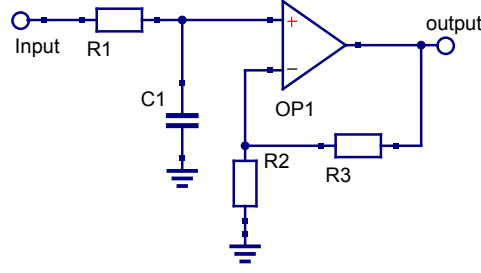


Figure 5.1: Low-pass first order active filter section

The transfer function of this section is following:

$$H(s) = \frac{K_v C \omega_c}{s + C \omega_c} \quad (5.1)$$

where K_v is passband voltage gain of filter section, and C is transfer function coefficient (2.36).

Denote by f_c (Hz) the cutoff frequency:

$$f_c = \frac{\omega_c}{2\pi} \quad (5.2)$$

RC-elements values could be determined as following:

$$C_1 = \frac{10}{f_c}, \quad [\mu\text{F}] \quad (5.3)$$

$$R_1 = \frac{1}{\omega_c C_1 C} \quad (5.4)$$

$$R_2 = \frac{K_v R_1}{K_v - 1} \quad (5.5)$$

$$R_3 = K_v R_1 \quad (5.6)$$

For unity gain $K_v = 1$ we have $R_3 = 0$ and $R_2 = \infty$. R_2 can be removed and R_3 can be shorted to obtain unity gain.

`Filter::calcFirstOrder` implements these evaluations. See `filter.cpp`.

5.2.2 High pass filters

First order section of high-pass filters circuit is shown in the Figure 5.1

Transfer function is:

$$H(s) = \frac{K_v s}{s + \omega_c / C} \quad (5.7)$$

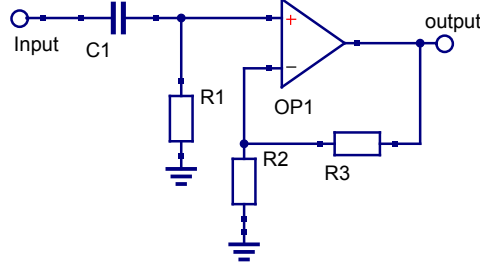


Figure 5.2: High-pass first order active filter section

where C is transfer function coefficient (2.36).

RC-elements values could be determined as following:

$$C_1 = \frac{10}{f_c}, \quad [\mu\text{F}] \quad (5.8)$$

$$R_1 = \frac{C}{\omega_c C_1} \quad (5.9)$$

R_2 and R_3 values could be determined using equations (5.5) and (5.6). For unity gain $K_v = 1$ R_2 should be removed and R_3 should be shorted.

`Filter::calcFirstOrder` implements these evaluations. This method determines RC-elements values for both low-pass and high-pass first order sections. See `filter.cpp` for source code.

5.3 Sallen-Key

5.3.1 Low pass filter

Sallen-Key topology of low-pass filter is shown in the Figure 5.3. This is second order section.

Transfer function of Sallen-Key section has form:

$$H(s) = \frac{K_v C \omega_c^2}{s^2 + B \omega_c s + C \omega_c^2} \quad (5.10)$$

where B and C are normalized coefficient for unity cutoff frequencies $\omega_c = 1$ (2.32), (2.33). These coefficient could be evaluated using poles and zeros location.

Consider Sallen-Key filter design procedure. C_2 capacitance could be estimated as following:

$$C_2 = \frac{10}{f_c}, \quad [\mu\text{F}] \quad (5.11)$$

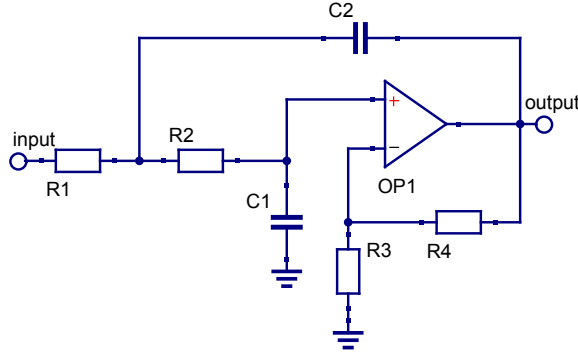


Figure 5.3: Low-pass Sallen-Key active filter section

Then C_1 capacitance could be estimated:

$$C_1 \leq \frac{[B^2 + 4C(K_v - 1)]C_2}{4C} \quad (5.12)$$

Then we can evaluate resistors values:

$$R1 = \frac{2}{\omega_c[BC_2 + \sqrt{[B^2 + 4C(K_v - 1)]C_2^2 - 4CC_1C_2}]} \quad (5.13)$$

$$R_2 = \frac{1}{CC_1C_2R_1\omega_c^2} \quad (5.14)$$

$$R_3 = \frac{K_v(R_1 + R_2)}{K_v - 1} \quad (5.15)$$

$$R_4 = K_v(R_1 + R_2) \quad (5.16)$$

For unity gain $K_v = 1$, R_3 should be removed and R_4 should be shorted. `SallenKey` class implements these evaluations and builds Sallen-Key circuit. See `sallenkey.cpp`.

5.3.2 High pass filters

Sallen-Key topology of high-pass filter is shown in the Figure 5.4. This is second order section.

Transfer function of this section has the following form:

$$H(s) = \frac{K_v s^2}{s^2 + (B\omega_c/C)s + \omega_c^2/C} \quad (5.17)$$

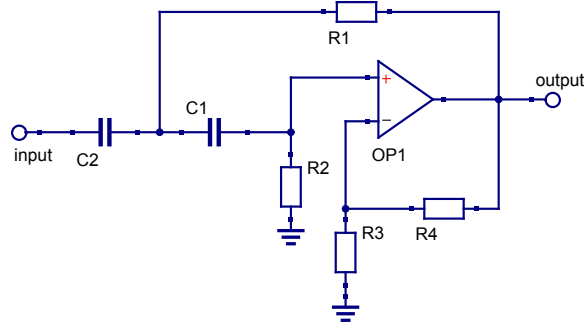


Figure 5.4: High-pass Sallen-Key active filter section

where B and C are normalized coefficients of the low-pass prototype filter. These coefficients could be evaluated using poles and zeros location of the low-pass prototype.

RC-elements values evaluation methods are similar to low-pass section. $C1$ and $C2$ capacitors values are equals:

$$C_1 = C_2 = \frac{10}{f_c}, \quad [\mu\text{F}] \quad (5.18)$$

Then we can evaluate resistors values:

$$R_2 = \frac{4C}{[B + \sqrt{B^2 + 8C(K_v - 1)}]\omega_c C_1} \quad (5.19)$$

$$R_1 = \frac{C}{\omega_c^2 C_1^2 R_2} \quad (5.20)$$

$$R_3 = \frac{K_v R_2}{k_v - 1} \quad (5.21)$$

$$R_4 = K_v R_2 \quad (5.22)$$

For unity gain $K_v = 1$, $R3$ should be removed and $R4$ should be shorted. `SallenKey` class implements these evaluations and builds Sallen-Key circuit. See `sallenkey.cpp`.

5.3.3 Band pass filter

Sallen-Key topology of band-pass filter is shown in the Figure 5.5. Two such section should be connected in series to implement 2-nd order section of band-pass filter.

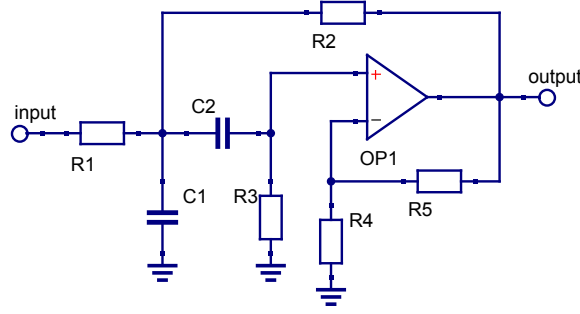


Figure 5.5: Band pass Sallen-Key active filter section

Transfer function of this topology is:

$$H(s) = \frac{\rho\omega_0}{s^2 + \beta\omega_0s + \gamma\omega_0^2} \quad (5.23)$$

where ρ , β , γ are some coefficients. These coefficients depend on RC-elements values.

Using this topology we can implement three types of filter sections:

1. Second order band-pass filter
2. Second-order (Chebyshev, Butterworth, or Cauer) band-pass filter section that corresponds second-order section of an LPF prototype.
3. Second-order band-pass filter section that corresponds first-order section of an LPF prototype.

To calculate such section at first we should determine ρ , β , γ coefficients. The will be following.

- For second order band-pass filter

$$\rho = \frac{K_v}{Q} \quad \beta = \frac{1}{Q} \quad \gamma = 1.0 \quad (5.24)$$

- For second-order (Chebyshev, Butterworth, or Cauer) band-pass filter section that corresponds second-order section of an LPF prototype. At first coefficients B_i and C_i should be evaluated from poles and zeros set of low-pass prototype using (2.32) and (2.33). Then supplementary coefficient D , E , F , H should be calculated for every section of the filter.

$$H = C + 4Q^2 \quad (5.25)$$

$$E = \frac{1}{B} \sqrt{\frac{(H + \sqrt{H^2 - 4B^2Q^2})}{2}} \quad (5.26)$$

$$F = \frac{BE}{Q} \quad (5.27)$$

$$D = \frac{F + \sqrt{F^2 - 4}}{2} \quad (5.28)$$

After this evaluation is performed we can evaluate ρ , β , γ coefficients. Second order section of band-pass filter consists of two sections. For first section coefficients are:

$$\rho = \frac{K_v \sqrt{C}}{Q} \quad (5.29)$$

$$\beta = D/E \quad (5.30)$$

$$\gamma = D^2 \quad (5.31)$$

And for second section:

$$\rho = \frac{K_v \sqrt{C}}{Q} \quad (5.32)$$

$$\beta = \frac{1}{DE} \quad (5.33)$$

$$\gamma = \frac{1}{D^2} \quad (5.34)$$

- For second-order band-pass filter section that corresponds first-order section of an LPF prototype. We need to calculate coefficient C_i from filter poles. Coefficients are:

$$\rho = \frac{K_v C}{Q} \quad \beta = \frac{C}{Q} \quad \gamma = 1.0 \quad (5.35)$$

Having ρ , β , γ coefficients we can calculate RC-elements values:

$$C_1 = C_2 = \frac{10.0}{2\pi\omega_0} \quad [\mu\text{F}] \quad (5.36)$$

$$R_1 = \frac{2}{\rho\omega_0 C_1} \quad (5.37)$$

$$R_2 = \frac{2}{-\beta + \sqrt{[(\rho - \beta)^2 + 8\gamma]\omega_0 C_1}} \quad (5.38)$$

$$R_3 = \frac{1}{\gamma\omega_0^2 C_1^2} \cdot \left(\frac{1}{R_1} + \frac{1}{R_2} \right) \quad (5.39)$$

$$R_4 = 2R_3 \quad (5.40)$$

Now we can substitute RC-elements values and build filter circuit. `SallenKey::calcBandPass()` implements these calculations. See `sallenkey.cpp`

5.4 Multifeedback

5.4.1 Low pass filter

Multifeedback (MFB) circuit of low-pass filter is shown in the Figure 5.6.

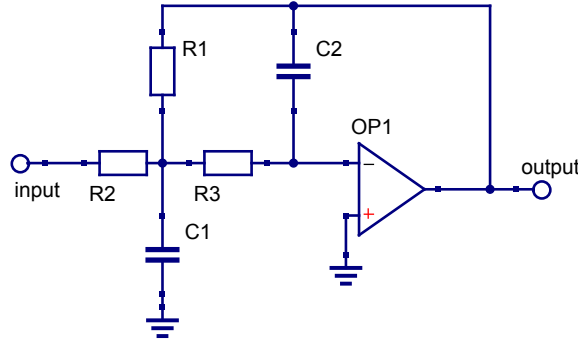


Figure 5.6: Low-pass multifeedback active filter section

Transfer function of this section has form (5.10). Having normalized transfer function coefficients B and C we can determine RC-elements values. At first, select C_2 using equation (5.11). Then we can determine C_1 value

$$C_1 \leq \frac{B^2 C_2}{4C(K_v + 1)} \quad (5.41)$$

Using C_1 and C_2 we can determine resistors values

$$R_2 = \frac{2(K_v + 1)}{[BC_2 + \sqrt{B^2C_2^2 - 4CC_1C_2(K_v + 1)}]\omega_c} \quad (5.42)$$

$$R_1 = \frac{R_2}{K_v} \quad (5.43)$$

$$R_3 = \frac{1}{CC_1C_2\omega_c^2R_2} \quad (5.44)$$

`MFBfilter` class is responsible for multifeedback filter circuit calculation and building. `MFBfilter::calcLowPass()` method evaluates RC-elements values.

`MFBfilter::createLowPassSchematic()` method builds low-pass filter schematic for Qucs. See `mfbbfilter.cpp` for source code.

5.4.2 High pass filters

Multifeedback (MFB) circuit of low-pass filter is shown in the Figure 5.7.

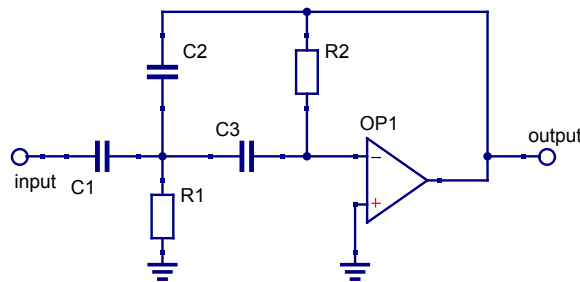


Figure 5.7: High-pass multifeedback active filter section

Transfer function of this section has form (5.17).

RC-elements value could be evaluated as following:

$$C_1 = C_3 = \frac{10}{f_c} \quad (5.45)$$

$$C_2 = \frac{C_1}{K_v} \quad (5.46)$$

$$R_1 = \frac{B}{\omega_c(2C_1 + C_2)} \quad (5.47)$$

$$R_2 = \frac{(2C_1 + C_2)C}{BC_1C_2\omega_c} \quad (5.48)$$

`MFBfilter::calcHighPass()` and `MFBfilter::createHighPassSchematic()` methods are responsible for multifeedback high-pass filter design. See `mfbbfilter.cpp`

5.4.3 Band pass filter

MFB band-pass filter section topology is shown in the Figure 5.8.

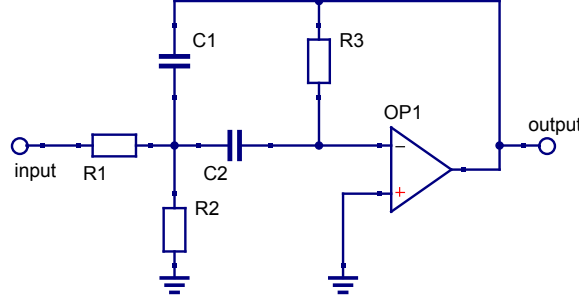


Figure 5.8: Band pass multifeedback active filter section

This section has transfer function of the form (5.23). It is need to determine ρ , β , γ coefficients set using method presented in previous section. Having these coefficients we can evaluate RC-elements values.

$$C_1 = \frac{10.0}{2\pi\omega_0} \quad [\mu\text{F}] \quad (5.49)$$

$$C_2 = \frac{C_1(\rho\beta - \gamma)}{\gamma} \quad (5.50)$$

If C_2 is less than zero ($C_2 < 0$) we should put $C_2 = C_1$.

$$R_1 = \frac{1}{\rho\omega_0} \quad (5.51)$$

$$R_2 = \frac{\beta}{[C_1(\gamma - \rho\beta) + \gamma C_2]\omega_0} \quad (5.52)$$

$$R_3 = \frac{1}{\beta\omega_0} \left(\frac{1}{C_1} + \frac{1}{C_2} \right) \quad (5.53)$$

Now we can substitute RC-elements values and build filter circuit. `MFBfilter::calcBandPass()` implements these calculations. See `mfbfilter.cpp`

5.5 Cauer and Chebyshev Type-II filters

5.5.1 Low pass filter

Low-pass Cauer filter schematic is shown in the Figure 5.9. High-pass section of Cauer filter has the same topology.

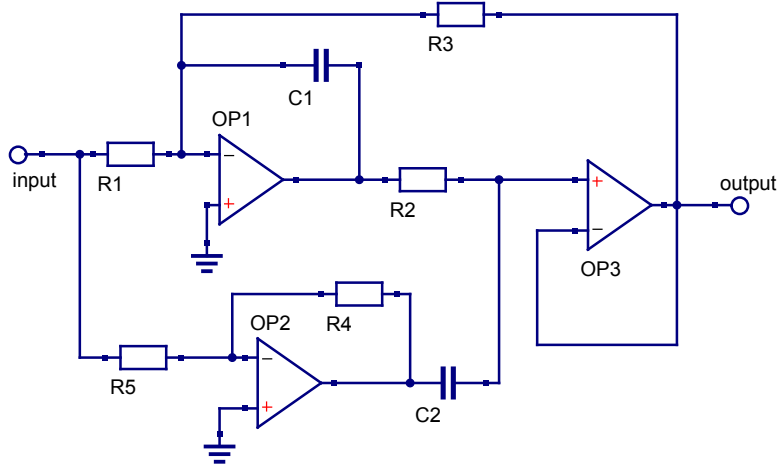


Figure 5.9: Low-pass and high-pass Cauer active filter section

We should know A_i , B_i , and C_i transfer function coefficients to evaluate RC-elements values of the Cauer filter.

Capacitors values are:

$$C_1 = C_2 = \frac{10}{f_c} \quad (\mu\text{F}) \quad (5.54)$$

Resistors values are:

$$R_5 = \frac{1}{\omega_c C_1} \quad (5.55)$$

$$R_1 = \frac{B R_5}{K_v C} \quad (5.56)$$

$$R_2 = \frac{R_5}{B} \quad (5.57)$$

$$R_3 = \frac{B R_5}{C} \quad (5.58)$$

$$R_4 = \frac{K_v C R_5}{A} \quad (5.59)$$

`SchCauer::calcLowPass()` and `SchCauer::createLowPassSchematic()` methods implement these evaluation. See `schcauer.cpp`

5.5.2 High pass filters

High-pass section of Cauer filters use the same topology as low-pass section (5.9). Capacitors values could be determined using equation (5.54). R5 resistors value using (5.55)

$$R_1 = \frac{ABR_5}{K_v C} \quad (5.60)$$

$$R_2 = \frac{CR_5}{B} \quad (5.61)$$

$$R_3 = BR_5 \quad (5.62)$$

$$R_4 = K_v R_5 \quad (5.63)$$

SchCauer::calcHighPass() and SchCauer::createHighPassSchematic() methods implement these evaluation. See schcauer.cpp

5.5.3 Band pass filters

The topology of Cauer band-pass filter differs from LPF and HPF topology. Schematic is presented in the Figure 5.10. There are two additional resistors R6 and R7.

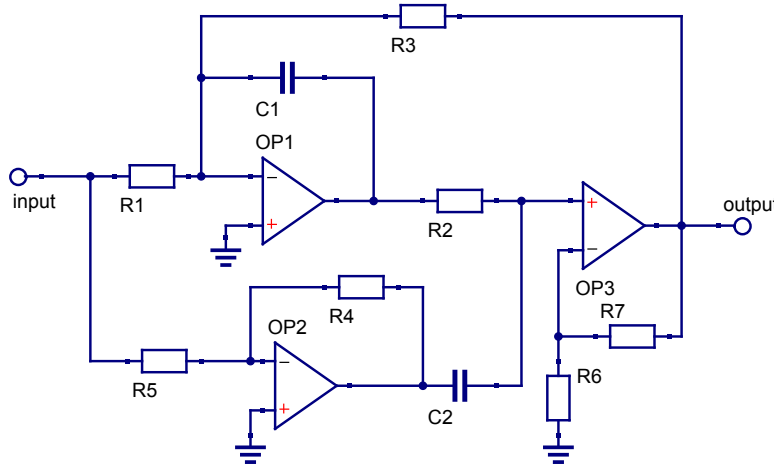


Figure 5.10: Low-pass and high-pass Cauer active filter section

Transfer function of such section has the following form:

$$H(s) = \frac{\rho(s^2 + \alpha\omega_0^2)}{s^2 + \beta\omega_0 s + \gamma\omega_0^2} \quad (5.64)$$

At first we need to determine A_i , B_i , C_i coefficients set (2.27), (2.32), and (2.33) for every section of the filter. Qucsactivefilter build schematic only for even order prototype of Cauer band-pass filters. If low-pass prototype has odd order, then order is expanded to nearest even order.

It's need to evaluate the following coefficients for every i -th section of the low-pass prototype:

$$A_1 = 1 + \frac{A + \sqrt{A^2 + 4AQ^2}}{2Q^2} \quad (5.65)$$

$$H = C + 4Q^2 \quad (5.66)$$

$$E = \frac{1}{B} \sqrt{\frac{H + \sqrt{H^2 - 4B^2Q^2}}{2}} \quad (5.67)$$

$$F = \frac{BE}{Q} \quad (5.68)$$

$$D = \frac{F + \sqrt{F^2 - 4}}{2} \quad (5.69)$$

Let μ be $\mu = 2.0$ [1].

Second order section of low-pass prototype corresponds two second-order sections of band-pass filter. Capacitors values for these sections are equals.

$$C1 = C2 = \frac{20\pi}{\omega_0} \quad (uF) \quad (5.70)$$

Resistors values for the first section:

$$R_1 = \frac{\mu D}{K_v A_1 E \omega_0 C_1} \cdot \sqrt{\frac{A}{C}} \quad (5.71)$$

$$R_2 = \frac{E}{DE \omega_0 C_2} \quad (5.72)$$

$$R_3 = \frac{\mu}{DE \omega_0 C_1} \quad (5.73)$$

For the second section:

$$R_1 = \frac{\mu A_1}{K_v DE \omega_0 C_1} \cdot \sqrt{\frac{A}{C}} \quad (5.74)$$

$$R_2 = \frac{DE}{\omega_0 C_2} \quad (5.75)$$

$$R_3 = \frac{\mu D}{E \omega_0 C_1} \quad (5.76)$$

For both sections:

$$R_5 = R_3 \quad (5.77)$$

$$R_4 = \frac{K_v R_5}{\mu} \cdot \sqrt{\frac{C}{A}} \quad (5.78)$$

$$R_6 = \frac{\mu R_2}{\mu - 1} \quad (5.79)$$

$$R_7 = \mu R_2 \quad (5.80)$$

Now we can substitute RC-elements values and build filter circuit. `SchCauer::calcBandPass()` implements these calculations. See `schcauer.cpp`

5.6 Band stop filters

Band stop filter can be implemented using Cauer active filter section topology for band-pass filters (Figure 5.10).

Transfer function of such section has the following form:

$$H(s) = \frac{\rho(s^2 + \alpha\omega_0^2)}{s^2 + \beta\omega_0 s + \gamma\omega_0^2} \quad (5.81)$$

At first, we need to evaluate some coefficients for every section of low-pass prototype. These coefficients should be found using A_i , B_i , C_i coefficients, that could be found from poles and zeros of low-pass prototype transfer function.

For Cauer or Chebyshev Type-II coefficient A_2 :

$$A_2 = 1 + \frac{1 + \sqrt{1 + 4AQ^2}}{2AQ^2} \quad (5.82)$$

For Butterworth, Chebyshev Type-I, or Bessel filters we should put $A_2 = 1$. Let μ be $\mu = 2.0$.

Other coefficients:

$$H = 1 + 4CQ^2 \quad (5.83)$$

$$E_1 = \frac{1}{B} \sqrt{\frac{C(H + \sqrt{H^2 - 4B^2Q^2})}{2}} \quad (5.84)$$

$$G = \frac{BE_1}{QC} \quad (5.85)$$

$$D_1 = \frac{G + \sqrt{G^2 - 4}}{2} \quad (5.86)$$

For the first section:

$$\alpha = A_2 \quad \beta = \frac{D_1}{E_1} \quad \gamma = G^2 \quad (5.87)$$

For the second section:

$$\alpha = \frac{1}{A_2} \quad \beta = \frac{1}{D_1 E_1} \quad \gamma = \frac{1}{G^2} \quad (5.88)$$

Capacitors values should be evaluated using (5.70). Resistors values for first section are:

$$R_1 = \frac{\mu\beta}{K_v\alpha\omega_0 C_1} \quad (5.89)$$

$$R_2 = \frac{1}{\beta\omega_0 C_2} \quad (5.90)$$

$$R_3 = \frac{K_v\alpha R_1}{\gamma} \quad (5.91)$$

$$R_5 = \frac{1}{\omega_0 C_1} \quad (5.92)$$

$$R_4 = \frac{K_v R_5}{\mu} \quad (5.93)$$

$$R_6 = R_7 = \frac{\mu R_2}{\mu - 1} \quad (5.94)$$

Now we can substitute RC-elements values and build filter circuit. `SchCauer::calcBandStop()` implements these calculations. See `schcauer.cpp`

Chapter 6

Conclusion

Qucsactivefilter is powerful tool that allows you to build filter circuits. The following filter topologies are implemented:

1. Sallen-Key low-pass, high-pass, and band-pass filters. Butterworth, Chebyshev Type-I, and Bessel approximations.
2. Multifeedback low-pass, high-pass, and band-pass filters. Butterworth, Chebyshev Type-I, and Bessel approximations.
3. Cauer low-pass, high-pass, and band-pass filters. Chebyshev Type-II and Cauer approximations.
4. Cauer band-stop filters. Butterworth, Chebyshev Type-I, Bessel, and Cauer approximations.
5. User defined transform function approximation — all types of filter topologies.

Background mathematics filter synthesis methods were considered. These methods are used in Qucsactivefilter sources.

You can implement physical filters after filter circuit is built and simulated with Qucs. It's need to note that real RC-elements values have tolerances. You should use RC-elements with at at least less than 1% tolerance. Filter circuit may require RC-elements values trimming. For trimming methods see [1] and [7].

Bibliography

- [1] D. Johnson, J. Johnson, and H. Moore, *A handbook of active filters*. Prentice-Hall, Inc, Engewood Cliffs, 1980.
- [2] R. W. Hamming, *Digital filters; 1st ed.*, ser. Prentice-Hall signal processing. Englewood Cliffs, NJ: Prentice-Hall, 1977.
- [3] C. B. Rorabaugh, *Digital Filter Designer's Handbook: Featuring C Routines*. New York, NY, USA: McGraw-Hill, Inc., 1992.
- [4] M. Biegert. The papoulis filter (aka optimum "l" filter). [Online]. Available: <http://mathscinotes.com/2011/06/the-papoulis-filter-aka-optimum-l-filter/>
- [5] U. Tietze, C. Schenk, and E. Gamm, *Halbleiter - Schaltungstechnik. Neuer Teil: Nachrichtentechnische Schaltungen*, 12th ed. Berlin: Springer, 2002.
- [6] P. Horowitz and W. Hill, *The Art of Electronics*. New York, NY, USA: Cambridge University Press, 1989.
- [7] G. S. Moschytz and P. Horn, *Active Filter Design Handbook*. New York, NY, USA: John Wiley & Sons, Inc., 1981.