

Lecture 16

Filter Design in CT

C.L. Wyatt

2026-03-05

Ideal Filters and Paley-Wiener Conditions

Recall from ECE 2714 the four basic frequency selective filters:

- low-pass
- high-pass
- band-pass
- band-stop / band-reject / notch

Recall also that the ideal versions of these are non-realizable because they are non-causal and violate the Paley-Wiener conditions. These conditions state that a causal filter:

- can have $H(\omega) = 0$ at only a finite number of frequencies
- cannot have $H(\omega) = 0$ over any finite range of frequencies
- transition from pass-band to stop-band and vice-versa cannot be zero width
- cannot specify $\|H(\omega)\|$ and $\angle H(\omega)$ independently

For example, the ideal low-pass filter has a frequency response

$$H(\omega) = \begin{cases} 1 & |\omega| < \omega_c \\ 0 & \text{else} \end{cases}$$

and clearly violates the first three conditions.

The inverse Fourier transform of the frequency response is the impulse response, in this case the sinc function

$$h(t) = \mathcal{F}^{-1} \{H(j\omega)\} = \frac{1}{\pi t} \sin(\omega_c t)$$

Since the impulse response is not a causal signal, the system is not causal and thus not physically realizable.

Practical Filters

Realizable filters must relax the strict conditions imposed by ideal filters. Practical filters are thus specified by one or more pass-bands, transition-bands, and stop-bands.

Low-Pass Specification

Consider a generic low-pass frequency response magnitude,

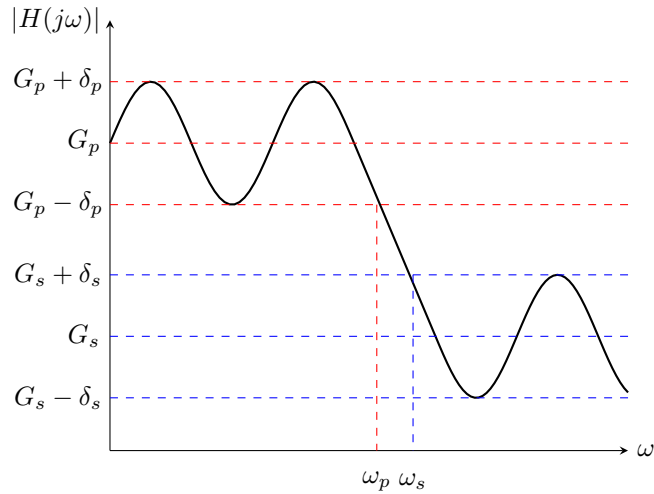


Figure 1: Illustration of low-pass filter design parameters.

where:

- G_p is the pass-band gain (often unity);
- δ_p is the pass-band ripple;
- ω_p is the pass-band edge frequency, also known as the filter bandwidth or cut-off frequency ω_c ;
- G_s is the stop-band gain;
- δ_s is the stop-band ripple;
- ω_s is the stop-band edge frequency;

- and $\omega_s - \omega_p$ is the transition bandwidth.

Generally, G_p , δ_p , G_s , and δ_s are specified in dB or V/V, and the edge frequencies, ω_p and ω_s , in rad/s or Hz.

Band-Pass Specification

The band-pass has two stop-bands (lower and upper), two transition bands (lower and upper); and a pass-band defined by a center frequency ω_0 and bandwidth, or two pass-band edge frequencies (lower and upper). Consider a generic band-pass frequency response magnitude,

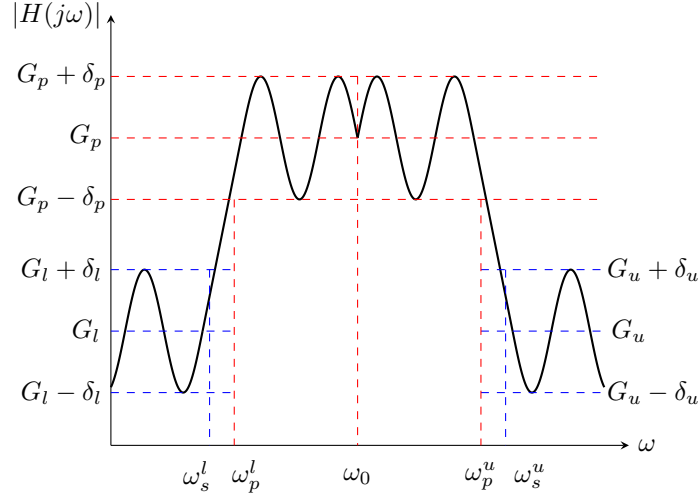


Figure 2: Illustration of band-pass filter design parameters.

where:

- G_p is the pass-band gain (often unity);
- δ_p is the pass-band ripple;
- ω_p^l, ω_p^u are the lower and upper pass-band edge frequencies;
- G_l, G_u are the lower and upper stop-band gains, often the same;
- δ_l, δ_u are the lower and upper stop-band ripple;
- ω_s^l, ω_s^u are the lower and upper stop-band edge frequencies;
- $\omega_s^u - \omega_p^u$ and $\omega_p^l - \omega_s^l$ are the upper and lower transition bandwidths;
- $\omega_p^u - \omega_p^l$ is the filter bandwidth;

- and ω_0 is the center frequency.

As with the low-pass filter specification, the gains and ripples are specified in dB or V/V, and the edge frequencies in rad/s or Hz.

Filter Design

We can focus on design of just low-pass filters since we can transform a low-pass frequency response into the others as described below. The design problem then is to find a transfer function $H(s)$ that is stable and has a corresponding frequency response, $H(\omega)$, approximately that desired. The most popular approach is to use particular parametric forms for the magnitude of the frequency response. We then find the transfer function by imposing symmetry and stability conditions that make the filter stable and have a real-valued impulse response.

Most filter design procedures allow for meeting a particular constraint on a filter parameter, while giving a best-effort on others. In general the tighter the parameters (e.g smaller transition bandwidths) the higher the order of the system required to implement that filter.

Butterworth Filters

Butterworth filters have a power spectrum of the form

$$|H(\omega)|^2 = \frac{1}{1 + \left(\frac{\omega}{\omega_c}\right)^{2N}}$$

for an order N system. They are maximally flat and are well approximated by a piece-wise linear $H(s)$ (see lecture 14).

To find the corresponding transfer function we note

$$|H(\omega)|^2 = H(\omega)H^*(\omega)$$

Since we desired a real-valued impulse response we impose conjugate symmetry, i.e. $H^*(\omega) = H(-\omega)$. Thus

$$|H(\omega)|^2 = H(\omega)H^*(\omega) = H(\omega)H(-\omega) = H(s)H(-s)|_{s=j\omega}$$

Then

$$H(s)H(-s) = \frac{1}{1 + \left(\frac{-s^2}{\omega_c^2}\right)^N}$$

When N is odd

$$\left(\frac{-s^2}{\omega_c^2}\right)^N = \frac{-s^{2N}}{\omega_c^{2N}}$$

and

$$H(s)H(-s) = \frac{-\omega_c^{2N}}{s^{2N} - \omega_c^{2N}}$$

The poles are the roots of $s^{2N} - \omega_c^{2N} = 0$ or

$$s^{2N} = \omega_c^{2N} = \omega_c^{2N} e^{j2\pi k} \quad \text{for } k = 0, 1, 2, \dots, 2N - 1$$

Thus we see that the poles are at $p_k = \omega_c e^{j\pi \frac{k}{N}}$.

By a similar argument when N is even $p_k = \omega_c e^{j\pi \frac{2k+1}{2N}}$ for $k = 0, 1, 2, \dots, 2N - 1$. This allows us to write the transfer function as

$$H(s) = \frac{\omega_c^N}{\prod_k (s - p_k)} \quad \text{for } \frac{\pi}{2} < \frac{k\pi}{N} < \frac{3\pi}{2}$$

where the condition on k is to ensure the poles are in the left-hand side of the complex plane, and thus correspond to a stable system.

To help visualize this we can plot the poles of both $H(s)$ and $H(-s)$.

If $H(s)$ has a pole $p_i = \alpha_i + j\omega_i$, then $H(-s)$ has a pole at $-p_i = -\alpha_i - j\omega_i$, the mirror image.

This configuration is sometimes called a "wall" of poles at radius ω_c . The order of the system, or equivalently the number of poles in the wall, N , must be chosen to meet the specifications.

We will skip the derivation but just give an expression in terms of the filter specification. Let $A = G_p - \delta_p$ and $B = G_s$ be specified in dB, then to meet the specifications the filter order must be the smallest integer N such that

$$N \geq \frac{\log_{10} \left[\frac{10^{-B/10} - 1}{10^{-A/10} - 1} \right]^{\frac{1}{2}}}{\log_{10} \left(\frac{\omega_s}{\omega_p} \right)}$$

For example, suppose $G_p = 0$ dB, $\delta_p = 3$ dB, $G_s = -40$ dB, and $\omega_p = 2\pi \cdot 5000$, $\omega_s = 2\pi \cdot 20000$.

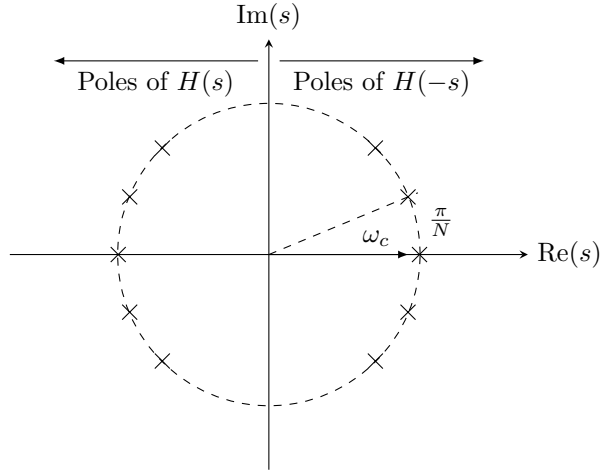


Figure 3: Illustration of pole locations for the Butterworth filter.

$$\left[\frac{10^{40/10} - 1}{10^{3/10} - 1} \right]^{\frac{1}{2}} \approx 100.2327$$

then

$$N \geq \frac{\log_{10}(100.2327)}{\log_{10}\left(\frac{20000}{5000}\right)} = 3.323$$

thus $N = 4$ is required to meet the specification.

In Matlab/Octave (and Python `spy.signal` library) the `butterd` function will give the order required to meet a given specification while the `butter` function will give the corresponding filter coefficients of the numerator and denominator polynomials, or equivalently the gain, zeros, and poles.

Chebyshev Filters

Type I Chebyshev filters have a power spectrum of the form

$$|H(\omega)|^2 = \frac{1}{1 + \epsilon^2 C_N^2\left(\frac{\omega}{\omega_p}\right)}$$

where $\epsilon > 0$ and $C_N(v)$ is the Chebyshev polynomial of order N

$$C_N(v) = \cos(N \cos^{-1}(v))$$

This polynomial is defined recursively with

$$C_{N+1}(v) = 2vC_N(v) - C_{N-1}(v)$$

where the base case of the recursion is $C_0(v) = 1$, $C_1(v) = v$. Thus

$$\begin{aligned} C_2(v) &= 2vC_1(v) - C_0(v) = 2v^2 - 1 \\ C_3(v) &= 2vC_2(v) - C_1(v) = 4v^3 - 3v \\ &\vdots = \vdots = \vdots \end{aligned}$$

The design procedure is then given ϵ , N , ω_p

1. Find $C_N(v)$
2. Substitute $v = \frac{\omega}{\omega_p}$
3. Find the poles of $H(s)$ equivalent to $|H(\omega)|$.

Step 3 is similar to the Butterworth example

$$|H(\omega)|^2 = H(s)H(-s)|_{s=j\omega} = \frac{1}{1 + \epsilon^2 C_N^2\left(\frac{s}{j\omega_p}\right)}$$

The poles of $H(s)H(-s)$ are solution to $1 + \epsilon^2 C_N^2\left(\frac{s}{j\omega_p}\right) = 0$ for $k = 0, 1, \dots, 2N - 1$.

1. Skipping the derivation

$$p_k = j\omega_p [\cos(\alpha_k) \cosh(\beta_k) - j \sin(\alpha_k) \sinh(\beta_k)]$$

where

$$\alpha_k = \frac{\pi(2k+1)}{2N} \quad \beta_k = \frac{\sinh^{-1}(1/\epsilon)}{N}$$

It remains to find N and ϵ that meet a desired specification. Again skipping the derivation, let $A = G_p - \delta_p$ and $B = G_s$.

$$\epsilon = \left[10^{-A/10} - 1\right]^{\frac{1}{2}} \quad N \geq \frac{\cosh^{-1}(F)}{\cosh^{-1}(\omega_0)}$$

where

$$F = \left[\frac{10^{-B/10} - 1}{10^{-A/10} - 1}\right]^{\frac{1}{2}} \quad \text{and} \quad \omega_0 = \frac{\omega_s}{\omega_p}$$

In Matlab/Octave (and Python `scipy.signal` library) the `cheb1ord` function will give the order required to meet a given specification while the `cheby1` function will give the corresponding filter coefficients of the numerator and denominator polynomials, or equivalently the gain, zeros, and poles.

Filter Transformations

Let $G(s)$ be the transfer function resulting from a low-pass design. Define

$$H(\lambda) = G(s)|_{s=f(\lambda)}$$

for $s, \lambda \in \mathbb{C}$ and filter transformation function $f : \mathbb{C} \rightarrow \mathbb{C}$.

low-pass to high-pass

Define

- ω_{LP} as low-pass pass-band edge
- ω_{HP} as high-pass pass-band edge

then

$$f(\lambda) = \frac{\omega_{LP} \cdot \omega_{HP}}{\lambda}$$

low-pass to band-pass

Define

- ω_{LP} as low-pass pass-band edge
- ω_{BR} as band-pass left pass-band edge
- ω_{BL} as band-pass right pass-band edge

then

$$f(\lambda) = \frac{\lambda^2 + \omega_0^2}{B\lambda}$$

where $\omega_0 = [\omega_{BR} \cdot \omega_{BL}]^{\frac{1}{2}}$ and $B = \frac{\omega_{BR} - \omega_{BL}}{\omega_{LP}}$.

Software Filter Design Examples

Below are some examples that demonstrate how to use software tools to design CT filters. These examples use the python `scipy.signal` library but this can also be done essentially the same way using Matlab.

Low-Pass Butterworth

Suppose the goal is to design the same low-pass filter with specifications:

- The pass-band gain is $G_p = 0$ dB.
- The pass-band ripple is $\delta_p = 3$ dB.
- The pass-band edge is 8 kHz.
- The stop-band gain is -20 dB.
- The stop-band edge is 10 kHz.

We start by importing the required Python libraries

```
import scipy.signal as signal
import matplotlib.pyplot as plt
import numpy as np
```

We then create variables holding the filter specification:

```
# --- Filter Specifications ---
rp = 3          # pass-band ripple (dB)
gpass = rp     # pass-band gain + ripple (dB)
wp = 2*np.pi*8000 # cut-off frequency (pass-band edge)
gstop = 20     # negative of stop-band gain (dB)
ws = 2*np.pi*10000 # stop-band edge
```

Then we call the functions to determine the required filter order and the corresponding numerator and denominator coefficients of the transfer function:

```
# --- Design the Filter ---
# Filter order required and corresponding pass-band edge
N, wn = signal.buttord(wp, ws, gpass, gstop, analog=True)
# Numerator and denominator coefficients
b, a = signal.butter(N, wn, btype='low', analog=True, output='ba')
```

The result is an 11th order filter with magnitude frequency response:

Note the pass-band is flat and the response meets the specifications, but the system order of 11 is pretty large and impractical for a CT filter, requiring six filter stages.

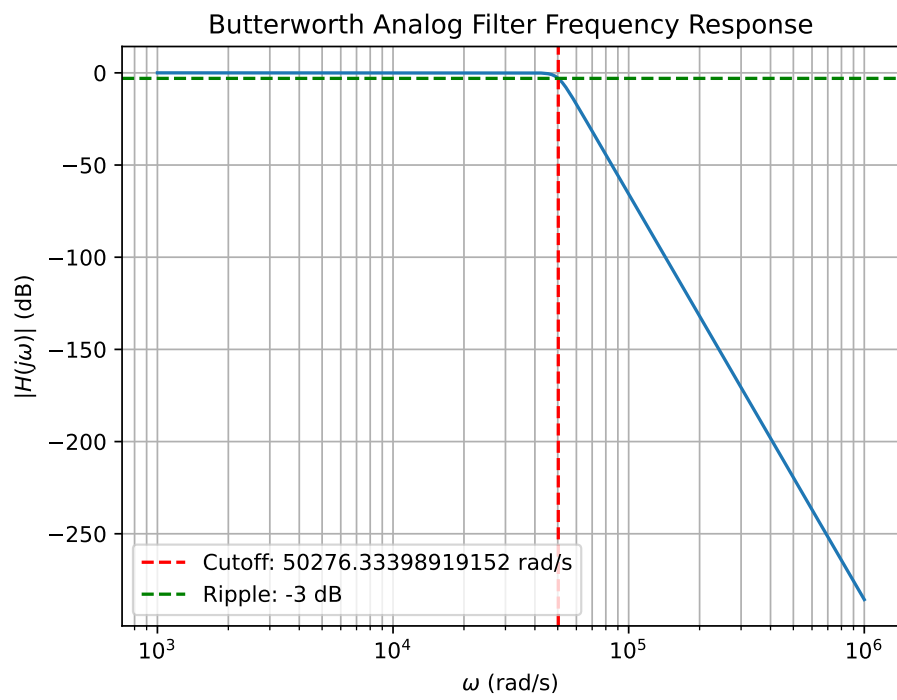


Figure 4: Magnitude of the Butterworth low-pass filter example.

Low-Pass Chebyshev

Suppose the goal is to design the same low-pass filter as in the Butterworth example above. Given the same setup as before, we call the functions to determine the required filter order and the corresponding numerator and denominator coefficients of the transfer function:

```
# --- Design the Filter ---  
# Filter order required and corresponding pass-band edge  
N, wn = signal.cheblord(wp, ws, gpass, gstop, analog=True)  
# Numerator and denominator coefficients  
b, a = signal.cheby1(N, rp, wn, btype='low', analog=True, output='ba')
```

The result is a 5th order filter with magnitude frequency response:

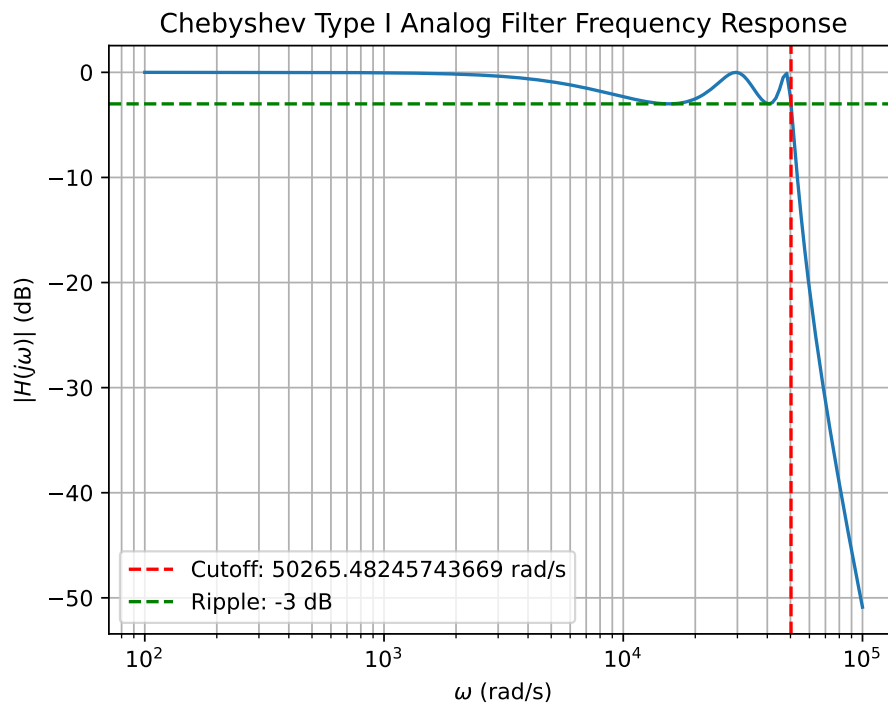


Figure 5: Magnitude of the Chebyshev low-pass filter example.

Note the required filter order is much smaller (5 vs 11) but the cost is more gain variation in the pass-band, although it meets the specifications given. This filter would be much more practical to implement requiring just three stages.