Butterworth, Elliptic, Chebychev Filters

Objective:
- Know what each filter tries to optimize
- Know how these filters compare

An Nth-order RC filter is the closest approximation to an ideal low pass filter subject to
- There are N poles
- All poles are real
- There are no zeros
- The maximum gain is 1.000

An Nth-order Butterworth filter is the closest approximation to an ideal low pass filter subject to
- There are N poles
- There are no zeros
- The maximum gain cannot exceed 1.0000

An Nth-order Type-1 Chebychev filter is the closest approximation to an ideal low pass filter subject to
- There are N poles
- There are no zeros
- The maximum gain cannot exceed $1 + \varepsilon$. (Some ripple is permitted).

An Nth-order Type-2 Chebychev filter is the closest approximation to an ideal low pass filter subject to
- There are N poles
- There are N zeros
- The maximum gain cannot exceed $1 + \varepsilon_1$. (Some ripple is permitted).
- The maximum gain the band reject region cannot exceed $\varepsilon_2$
RC Filter

An n-pole RC filter has all n-poles on the real axis.

- It's advantage is you can build it with a passive RC filter (good)
- It's problem is it's a pretty poor filter.

For example, the gain of

\[ G(s) = \left( \frac{10}{s+10} \right)^n \]

for n=1, 2, 3, and 4 is shown below. Note that

- As n increases, the high-frequency gain gets smaller and smaller (good)
- However, the gain below 5 rad/sec starts to droop more and more (bad)

\[ \text{-->w} = [0:0.1:50]'; \]
\[ \text{-->s} = j*w; \]
\[ \text{-->G} = 10 ./ (s+10); \]
\[ \text{-->plot(w,abs([G,G.^2,G.^3,G.^4]));} \]
\[ \text{-->xlabel('rad/sec');} \]
\[ \text{-->ylabel('gain');} \]
Butterworth Filter:

If you use complex poles, you can do better. For example, take the case of \( n = 5 \).

A 5th-order RC filter with a corner at 10 rad/sec is

\[
G = \left( \frac{10}{s+10} \right)^5
\]

One of these poles has to be real. The other four, however, could be moved along the circle centered at the origin with a radius of 10. As you increase the angle of these poles, the gain at \( j10 \) increases. If you go too far, the gain at \( j10 \) starts to go above one.

A Butterworth filter is the farthest you can slide the poles while keeping the maximum gain less than one.

As it turns out, the optimal filter (i.e. the closest to an ideal low-pass filter) with

- \( N \) poles
- No zeros, and
- No resonance (max gain <= 1.0000)

has \( N \) poles with

- The amplitude of the poles is equal to the corner frequency, and
- The angle between the poles is

\[
\phi = \frac{180^0}{N}
\]

If the corner is 1 rad/sec, for example, the poles for a Butterworth filter are

<table>
<thead>
<tr>
<th>( N )</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>( s+a )</td>
<td>( 1 \angle \pm 45^0 )</td>
<td>( 1 )</td>
<td>( 1 \angle \pm 22.5^0 )</td>
<td>( -1 )</td>
</tr>
<tr>
<td></td>
<td>( 1 \angle \pm 60^0 )</td>
<td>( 1 \angle \pm 67.5^0 )</td>
<td>( -1 \angle \pm 36^0 )</td>
<td>( -1 \angle \pm 72^0 )</td>
</tr>
</tbody>
</table>

Example: Design a 4th-order Butterworth low-pass filter with a corner at 10 rad/sec.

Solution: The poles are 45 degrees apart

\[
\phi = \frac{180^0}{4} = 45^0
\]

The amplitude of the poles is the corner frequency (10 rad/sec). Hence

\[
G(s) = \left( \frac{10^4}{(s+10\angle22.5^0)(s+10\angle-22.5^0)(s+10\angle67.5^0)(s+10\angle-67.5^0)} \right)
\]

or since complex poles are always in complex conjugate pairs,
\[ G(s) = \left( \frac{10^4}{(s+10\angle\pm22.5^\circ)(s+10\angle\pm67.5^\circ)} \right) \]

The resulting gain vs. frequency is then

```matlab
% p1 = 10 * exp(j*22.5*pi/180);
p2 = conj(p1);
p3 = 10 * exp(j*67.5*pi/180);
p4 = conj(p3);
w = [0:0.01:30]';
s = j*w;
Gd = 1 * (w < 10);
Gs = 10^4 ./ ( (s+p1) .* (s+p2) .* (s+p3) .* (s+p4) ) ;
plot(w,abs(Gs),'b',w,abs(Gd),'m');
xlabel('Frequency (rad/sec)');
ylabel('Gain');
```
Note that with a Butterworth filter,

- The more poles you have the closer it gets to an ideal low-pass filter.
- All poles have the same amplitude
- All poles have an equal spacing between them

Gain of a Butterworth Filter for n=2 (red), 3 (magenta), 4 (blue), 5 (green)
Type-1 Chebychev Filter

A Chebychev filter is similar to a Butterworth filter, except you allow the gain to be slightly larger than one at some point. In return, you get a filter which is closer to ideal.

Unlike a Butterworth filter, there are an infinite number of Chebychev filters: one for each resonance you allow. The poles for a Type-1 Chebychev filter with a corner at 1 rad/sec follow and 0.2 ripple are given below.

<table>
<thead>
<tr>
<th>N=2</th>
<th>N=3</th>
<th>N=4</th>
<th>N=5</th>
<th>N=6</th>
</tr>
</thead>
<tbody>
<tr>
<td>zeros</td>
<td>none</td>
<td>none</td>
<td>none</td>
<td>none</td>
</tr>
<tr>
<td>poles</td>
<td>$-1.60 \angle \pm 50.7^0$</td>
<td>$-0.85$</td>
<td>$-0.72 \angle \pm 38.5^0$</td>
<td>$-0.48$</td>
</tr>
<tr>
<td></td>
<td>$-1.21 \angle \pm 69.5^0$</td>
<td>$-1.11 \angle \pm 77.8^0$</td>
<td>$-0.76 \angle \pm 59.3^0$</td>
<td>$-1.06 \angle \pm 82.0^0$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>n = 2</th>
<th>n = 3</th>
<th>n = 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>$</td>
<td>pole</td>
<td>$</td>
</tr>
<tr>
<td>1.5967184</td>
<td>50.654966</td>
<td></td>
</tr>
<tr>
<td>0.8494427</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>n = 5</th>
<th>n = 6</th>
<th>n = 7</th>
</tr>
</thead>
<tbody>
<tr>
<td>$</td>
<td>pole</td>
<td>$</td>
</tr>
<tr>
<td>1.0649391</td>
<td>82.007862</td>
<td></td>
</tr>
<tr>
<td>0.7583392</td>
<td>59.258028</td>
<td></td>
</tr>
<tr>
<td>0.4791522</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>n = 8</th>
<th>n = 9</th>
<th>n = 10</th>
</tr>
</thead>
<tbody>
<tr>
<td>$</td>
<td>pole</td>
<td>$</td>
</tr>
<tr>
<td>1.0236433</td>
<td>86.797811</td>
<td></td>
</tr>
<tr>
<td>0.8816164</td>
<td>79.356269</td>
<td></td>
</tr>
<tr>
<td>0.3520881</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0.2597740</td>
<td>0.2809011</td>
<td>34.87934</td>
</tr>
</tbody>
</table>

JSG 6 April 30, 2020
Example: Design a 4th-order Chebuev filter with

- A DC gain of 1.000
- A ripple of 0.2, and
- A corner at 10 rad/sec

Solution:

\[ G(s) = \left( \frac{7.2^2 \cdot 11.1^2}{(s+7.2\angle\pm38.5^0)(s+11.1\angle\pm77.8^0)} \right) \]

or

\[ G(s) = \left( \frac{6387}{s^2+11.27s+51.84}(s^2+4.69s+123.21) \right) \]

Checking the gain vs. frequency in Matlab

```matlab
p1 = 7.2 * exp(j*38.5*pi/180);
p2 = conj(p1);
p3 = 11.1 * exp(j*77.8*pi/180);
p4 = conj(p3);
w = [0:0.01:30]';
s = j*w;
Gd = 1 * (w < 10) .* (w > -10);
Gs = 7.2^2 * 11.1^2 ./ ( (s+p1) .* (s+p2) .* (s+p3) .* (s+p4) );
plot(w,abs(Gs),'b',w,abs(Gd),'m');
xlabel('Frequency (rad/sec)');
ylabel('Gain');
```

Gain vs. Frequency for a 4th-Order Chebychev Filter with a Corner at 10 rad/sec
Elliptic Filters:

An Elliptic filter adds zeros to force the gain to drop quickly at the corner. The tradeoff is the highfrequency gain doesn't roll off like they do with Butterworth of Type-1 Chebychev filters. An elliptic filter with an attenuation of 60dB in the reject region follows:

<table>
<thead>
<tr>
<th>Pass</th>
<th>W1 = 0 to 1</th>
<th>W1 = 0 to 1</th>
<th>W1 = 0 to 1</th>
<th>W1 = 0 to 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reject</td>
<td>W2 = 9 to infinity</td>
<td>W2 = 3 to infinity</td>
<td>W2 = 1.7 to infinity</td>
<td>W2 = 1.3 to infinity</td>
</tr>
<tr>
<td>zeros</td>
<td>j 9.919</td>
<td>j3.246</td>
<td>j1.831</td>
<td>j1.316</td>
</tr>
<tr>
<td></td>
<td>j7.705</td>
<td>j2.907</td>
<td>j1.524</td>
<td>j2.491</td>
</tr>
<tr>
<td>poles</td>
<td>-0.391 + j1.242</td>
<td>-0.572 + j0.467</td>
<td>-0.494</td>
<td>-0.437</td>
</tr>
<tr>
<td></td>
<td>-0.942</td>
<td>-0.221 + j1.076</td>
<td>-0.365 + j0.660</td>
<td>-0.331 + j0.583</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>-0.143 + j1.005</td>
<td>-0.159 + j0.907</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>-0.040 + j1.024</td>
</tr>
</tbody>
</table>

Note that zeros are placed on the jw axis. These force the gain to zero at these frequencies - and close to zero near them. With the use of complex zeros, you can force the gain to drop more quickly past the passband. The drawback is the gain 'just' remains below 0.001 rather than rolling off to zero as it would do with all poles and no zeros.

Also note that the closer the reject region is to the passband, the more poles and zeros are required.

Comparison of Gains vs. Frequency:
Gain vs. Frequency for the Butterworth filter (blue), Chebychev filter (green), and Elliptic filter (red).

In the above gain vs. frequency, note that the Elliptic filter has a gain which goes to zero at 2.078 and 3.212 rad/sec. In the reject region, the gain is less than 0.001 - which was one of the design constraints in the elliptic filter presented here.

The Chebychev and Butterworth filter both have gains that keep dropping off as $\omega^{-4}$ since these filters have four poles and no zeros. If you want the gain to keep rolling off, the Butterworth or Elliptic filters are better. 0.001 is small enough, the Elliptic filter is the closest approximation to an ideal low pass filter.

8th-Order Filters
Pole location of a Butterworth filter (blue) and Chebychev filter (green) along with the unit circle.

The gain vs. frequency is shown below. Note that the Chebychev filter trades off ripple in the passband for a faster rolloff outside the passband. They both have a gain which drops off as $\omega^{-8}$ with eight poles and no zeros.
8th Order Butterworth, Chebychev, 4th Order Elliptic
SciLab Code:

zpell - lowpass elliptic filter

Calling Sequence

[zeros,poles,gain]=zpell(epsilon,A,omegac,omegar)

Parameters

epsilon : real : ripple of filter in pass band ( 0<epsilon<1 )
A : real : attenuation of filter in stop band ( A>1 )
omegac : real : pass band cut-off frequency in Hertz
omegar : real : stop band cut-off frequency in Hertz
zeros : resulting zeros of filter
poles : resulting poles of filter
gain : resulting gain of filter

Description

Poles and zeros of prototype lowpass elliptic filter. gain is the gain of the filter

zpbutt - Butterworth analog filter

Calling Sequence

[pols,gain]=zpbutt(n,omegac)

Parameters

n : integer (filter order)
omegac : real (cut-off frequency in Hertz)
pols : resulting poles of filter
gain : resulting gain of filter

Description

computes the poles of a Butterworth analog filter of order n and cutoff frequency omegac transfer function H(s) is calculated by
H(s)=gain/real(poly(pols,'s'))

zpchl - Chebyshev analog filter

Calling Sequence

[poles,gain]=zpchl(n,epsilon,omegac)

Parameters

n : integer (filter order)
epsilon : real : ripple in the pass band ( 0<epsilon<1 )
ometgac : real : cut-off frequency in Hertz
poles : resulting filter poles
gain : resulting filter gain

Description
Poles of a Type 1 Chebyshev analog filter. The transfer function is given by:

\[ H(s) = \text{gain/poly}(\text{poles},'s') \]