Analogue Filters

Revision Notes by Michael Prior-Jones

The ideal filter

An ideal filter has two characteristics:

- a "brick-wall" amplitude response: i.e. a perfectly flat passband and an infinitely sharp transition to a perfectly flat, infinite attenuation stopband.
- a constant time-delay through it at all frequencies: this gives a linear phase response.

Of course, we'd really like a zero-delay filter, but this is clearly impossible as filters need to store information/energy to do their job.
The amplitude and phase response are interlinked- one implies the other.

Butterworth filter

Our first approximation to an ideal filter.

- Maximally flat passband and stopband.
- Reasonable phase response.
- Relatively slow transition between passband and stopband.

Transfer function of the Butterworth filter:

$$\left|H(jw)\right|^2 = \frac{H_0}{1+\left(\frac{w}{w_c}\right)^{2n}}$$

This response is given in terms of power, where *H(jw)* is the voltage transfer function.

- H_0 is the passband response: i.e. the gain of the filter at d.c.
- w_c is the cut-off frequency, which for Butterworth filters is the point at which you get 50% of input power lost in the filter- the 3dB point.
- n is the order of the filter, i.e. the number of reactive components.

Butterworth filters roll-off at 20n dB/decade (6n dB/octave), which is not particularly brilliant. To make a sharp transition requires a large number of components.

Normalisation assumptions

Filters are normally designed in a *normalised form,* to simplify the algebra.
This assumes:

- the cut-off frequency, w_c, is 1 rad/s
- H_0 is 1, giving the filter unity d.c. gain.

So the transfer function for a normalised Butterworth is:

$$|H(jw)|^2 = \frac{1}{1+w^{2n}}$$

Poles and polynomials for the Butterworth filter

The poles of the Butterworth transfer function are spaced equidistantly around a unit circle in the *s*-plane. They never lie on the Imaginary axis, but may lie on the Real axis. There are 2*n* poles, as the LHP and RHP are mirror images, and we select on LHP poles to ensure stability. So, for a stable 3rd order Butterworth filter, you find 6 poles and select three.

You can find Butterworth poles by simple geometry, or you can use the formulae:

$$s_k = \sin\left(\frac{2k-1}{2n}p\right) \text{(real part of poles)}$$

$$w_k = \cos\left(\frac{2k-1}{2n}p\right) \text{(imaginary part of poles)}$$

$$\text{for } k = 1 \rightarrow 2n$$

To find the poles, draw out a table of values and select the pairs where the real part is negative. Then assemble a Butterworth polynomial in *s*. You can then substitute for $s=jw$ and rearrange to get an equation in *w*.

Since Butterworth polynomials are very standardised, they're easy to look up in tables. This makes things very easy indeed!

The Chebyshev filter

This has a selection of interesting properties:

- it has ripple in its passband
- a sharper transition between passband and stopband
- poor phase performance.

Transfer function of the normalised Chebyshev

$$|H(jw)|^2 = \frac{H_0}{1+e^2 C_n^2}$$

Epsilon (ε) is the ripple factor.
C_n represents the n^{th} order Chebyshev polynomial, which is best defined by a recurrence relation:

$$C_{(n+1)} = 2wC_n - C_{(n-1)}w$$

Using this, you can work through to find the polynomial at any order.

Order	Polynomial
C_0	1
C_1	ω
C_2	$2\omega^2$-1
C_3	$4\omega^3$-3ω

Alternatively, and this comes in handy for solving certain problems, you can use the following pair of formulae:

$$C_n = \cos(n\cos^{-1}(w)) \text{ when } w \le 1$$

$$C_n = \cosh(n\cosh^{-1}(w)) \text{ when } w \ge 1$$

Roll-off of the Chebyshev:

The Chebyshev has much faster roll-off than the Butterworth: it’s correlated to the ripple factor, more ripple equals faster roll-off.

The cut-off frequency of the Chebyshev is defined at the point where the response is attenuated by the ripple factor, rather than being 3dB down.
Also, if the filter has an odd order, the gain at d.c. is unity. If it has an even order, the d.c. gain is:

$$|H(0)|^2 = \frac{1}{1+e^2}$$

Pole locations for the Chebyshev filter:

The poles of the Chebyshev are similar to the Butterworth, but lie on an ellipse rather than a unit circle. The pole positions are given by:

$$s_k = \sin u_k \sinh v$$

$$w_k = \cos u_k \cosh v$$

$$where$$

$$u_k = \frac{(2k-1)p}{2n}$$

$$v = \frac{1}{n}\sinh^{-1}\left(\frac{1}{e}\right)$$

Filter synthesis: LC networks

1. Find power transmission coefficient for network:
   $$\left|t_{jw}\right|^2 = \text{normalised filter frequency response equation}$$
   $$\left|t_{jw}\right|^2 = \frac{1}{1+w^{2n}} \text{ (Butterworth transfer function)}$$
2. Find power reflection coefficient from network using formula:
   $\left|r_{jw}\right|^2 = 1 - \left|t_{jw}\right|^2$ and rearrange to get a simple equation in ω.
3. Transfer into *s*-domain: substitute $w = \frac{s}{j}$ and rearrange to get a sensible transfer function, $\left|r(s)\right|^2$
4. Given that $\left|r(s)\right|^2 = r(s)r(-s)$, break up the transfer function to obtain only the left-half plane part, *r*(*s*). The denominator of this function is your characteristic polynomial (i.e. Butterworth, Chebyshev).
5. Substitute your formula for *r*(*s*) into the formula for the input impedance of the filter: $Z_{in} = R_1 \frac{1 \pm r(s)}{1 \mp r(s)}$ You can choose which sign combinations to use, but it's best to chose the one that results in a vulgar fraction.
6. Perform algebraic long division, and reformulate the equation as two terms, one whole term and one fractional one. Write the fractional term as the reciprocal of a reciprocal, and divide it through again. Do this again and again until you break down the equation so that no term in *s* has a higher power than 1.
7. Draw out resultant circuit from equation for Z_{in}, using the conventional Laplace representations of passive components: inductors are represented by *s*, capacitors by s^{-1} and resistors by constants. It will almost always be an LC ladder, either in tee or pi sections.

Denormalisation and transformation

Once you have a normalised prototype circuit you can then denormalise to get a real circuit. This involves performing impedance denormalisations, frequency denormalisations and (optionally) frequency transformations.

Impedance and frequency denormalisations:

In order to get sensible component values for passive filters, it's necessary to raise the filter's impedance. Also, its usually necessary to make w_c equal to some value w_0 , rather than 1 rad/s. Put simply, choose a value of w_0 and then select R_0 to make the values sensible and alter the component values like this:

Prototype circuit	**Denormalised circuit**
Resistor, *R* ohms	Resistor, RR_0 ohms
Capacitor, *C* farads	Capacitor, $\frac{C}{R_0 w_0}$ farads
Inductor, *L* henrys	Inductor, $\frac{LR_0}{w_0}$ henrys

Frequency transformations

All filters are designed as lowpass- we can easily transform our circuits to make highpass or bandpass filters.

Highpass transform:

Prototype circuit	**Transformed and denormalised circuit**
Resistor, *R* ohms	Resistor, RR_0 ohms
Capacitor, *C* farads	Inductor, $\frac{R_0}{Cw_0}$ henrys
Inductor, *L* henrys	Capacitor, $\frac{1}{w_0 LR_0}$ farads

Bandpass transform

Decide the bandwidth B and the logarithmic midband frequency w_0 (i.e. the midpoint of the passband if drawn on a log scale) and then perform the transform:

Prototype circuit	**Transformed and denormalised circuit**
Resistor, *R* ohms	Resistor, RR_0 ohms
Capacitor, *C* farads	**Parallel combination of:** Inductor, $\frac{BR_0}{Cw_0^2}$ henrys Capacitor, $\frac{C}{BR_0}$ farads
Inductor, *L* henrys	**Series combination of:** Inductor, $\frac{LR_0}{B}$ henrys Capacitor, $\frac{B}{w_0^2 LR_0}$ farads

Asymmetric filters

If you want to make a filter with differing source and load impedances, you can divide the prototype circuit into two symmetrical halves and then denormalise each half separately. If you need to split a component, replace R or L with a series combination and C with a parallel combination.
Once you have denormalised, recombine the two halves.

Passive vs Active filters:

Passive, LC filters:

- low sensitivity to component variations
- use inductors: costly, bulky, lossy, generate magnetic fields.

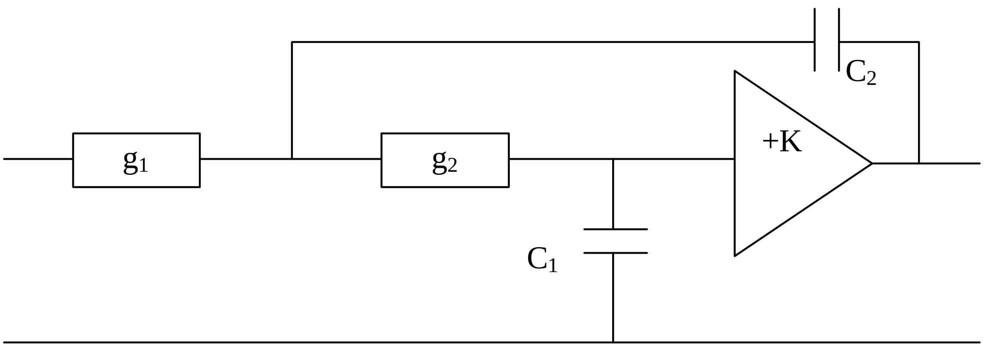
Active filters:

- no inductors, no losses
- small
- can be integrated onto chips
- require power supply
- can be more sensitive to component variations
- only work at low frequencies (limited by amplifiers)

Two types of active filter:

- direct simulation, using a *gyrator* circuit as an active replacement for an inductor.
- second order cascade approach, using Sallen & Key active filter circuit.

Sallen & Key Filters



The Sallen & Key circuit is a second order filter. They can be cascaded to create any even-order filter, and a final RC stage can be added to make any odd-order filter.

In order to realise a cascaded S&K filter, break the filter transfer function up into a product of several second order functions. Each of these second order functions can then be easily translated into an S&K circuit.

$$H(s) = \frac{G_1}{s^2 + a_1 s + b_1} \times \frac{G_2}{s^2 + a_2 s + b_2}$$

The coefficients of the denominator of each filter stage (*a* and *b*) can be translated into S&K component values using one of three sets of equations. Don't forget that *g* is conductance.

Method 1

$$K = 1$$

$$g_1 = g_2 = 1$$

$$C_1 = \frac{2}{a}$$

$$C_2 = \frac{a}{2b}$$

Method 2

$$K = 2$$

$$C_1 = C_2 = 1$$

$$g_1 = a$$

$$g_2 = \frac{b}{a}$$

Method 3

$$C_1 = C_2 = 1$$

$$g_1 = g_2 = g$$

$$g = \sqrt{b}$$

$$K = \left(3 - \frac{a}{\sqrt{b}}\right)$$

Denormalise as described for LC filters