Filter Basics

Filters are devices (or algorithms) which change the spectrum of signals - their most prevalent action on signals is to boost, attenuate or completely block frequencies. In the tutorial about sinusoids, we saw that any signal can be seen a sum of sinusoids, each of which having its own frequency f, amplitude A and phase φ. In general terms, filters modify the amplitudes and phases of incoming sinusoids according to their frequency. Assume that the input signal to our filter is a single sinusoid with frequency f, amplitude A and phase φ. The output signal will again be a sinusoid of frequency f but possibly with different values for the amplitude and phase. And this is already the essence of a filter: all a (linear) filter can do to a sinusoid is to multiply its amplitude by some factor and to add some offset to its phase; but a filter will never change the general shape of the sinusoid, nor will it change its frequency. So, let's call that multiplication factor for the amplitude G (for gain) and let's call that phase shift θ (a greek lowercase theta). Both, gain and phase-shift depend on the frequency of the incoming sinusoid, so both G and θ can be expressed functions of the frequency f:

$$G = G(f) \qquad \text{and} \qquad \theta = \theta(f) \tag{1}$$

These two functions are called the magnitude response and the phase response of the filter respectively. Taken together, these two functions form the frequency response of the filter - in a somewhat sloppier slang, one also often hears the term frequency response when the magnitude response alone is meant because this part of the frequency response often counts more. When we know these two functions, we can can predict the output signal of the filter for any arbitrarily shaped input signal by viewing the input signal as a sum of sinusoids.

Ideal Filters

The classical purpose of a filter is to let certain frequencies pass unchanged and to block others - hence the name filter. An idealized lowpass-filter for example would pass all frequencies up to some cutoff-frequency f_c and block all frequencies above that cutoff-frequency. The idealized lowpass magnitude response $G_{LP}(f)$ would be therefore:

$$G_{LP}(f) = \begin{cases} 1 & \text{for } f \leq f_c \\ 0 & \text{for } f > f_c \end{cases} \tag{2}$$

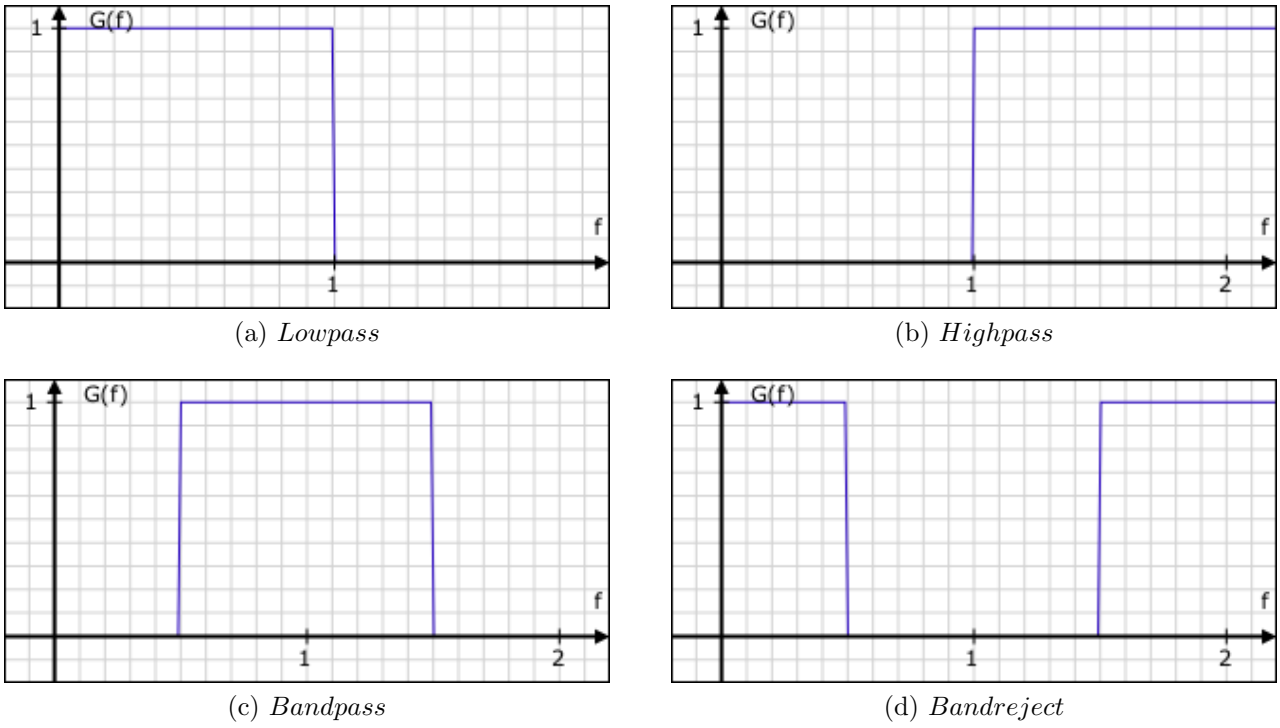
As we do not see any reason to intentionally introduce phase shift, we would want the phase response of the ideal filter to be identically zero for all frequencies:

$$\theta(f) = 0 \tag{3}$$

Ideal highpass filters, on the other hand, should pass frequencies above the cutoff frequency and block anything below. Bandpass filters should pass everything within some frequency interval between a lower cutoff frequency f_l and an upper cutoff frequency f_u. Bandreject filters are the opposite of bandpass filters - they block everything between f_l and f_h and let everything outside this interval pass unchanged. For bandreject filters with a very narrow rejection interval, one also often uses the term 'notch-filter'. The magnitude responses of ideal highpass, bandpass and bandreject filters would be:

$$G_{HP}(f) = \begin{cases} 0 & \text{for } f < f_c \\ 1 & \text{for } f \geq f_c \end{cases} \qquad G_{BP}(f) = \begin{cases} 1 & \text{for } f_l \leq f \leq f_u \\ 0 & \text{otherwise} \end{cases} \qquad G_{BR}(f) = \begin{cases} 0 & \text{for } f_l < f < f_u \\ 1 & \text{otherwise} \end{cases} \tag{4}$$

We would like the phase response to be identically zero for the filter types as well in the ideal case. In figure 1 we see the ideal magnitude responses for such filters. The lowpass and highpass filters in these plots are normalized in the sense that they have a cutoff frequency of unity. The physical unit in which frequency is measured is actually irrelevant for this discussion, but if you prefer to deal with something concrete and practical, feel free to suppose it to be kHz. The bandpass and bandreject filters have a lower cutoff of 0.5 and an upper cutoff of 1.5.



(a) *Lowpass* (b) *Highpass*

(c) *Bandpass* (d) *Bandreject*

Figure 1: magnitude responses of idealized filters

Real Filters

Unfortunately (or maybe not), we do not live in a perfect world and these ideal frequency responses as requested above are not attainable in the real world. Real filters can only approximate these requirements and various filter design techniques exist to obtain different kinds of approximations. As it turns out from the mathematics, the squared magnitude responses of realizable filters are always ratios of even polynomials (also called rational functions), that is functions of the form:

$$G^2(f) = \frac{B^2(f)}{A^2(f)} \tag{5}$$

where $B(f)$ and $A(f)$ are both polynomials of their argument f. So, in order to design a filter which approximates any desired magnitude response, we must find a rational function which best approximates the desired magnitude response. This is the so called approximation problem in filter design. For the lowpass-, highpass-, bandpass- and bandreject-responses, standard solutions to this design problem exist and these are are known under the names Butterworth, Chebyshev, inverse Chebyshev, Bessel, elliptic,

etc. The simplest of these is the Butterworth approximation. In the case of a lowpass filter with unit cutoff frequency the approximant is chosen as:

$$G^2(f) = \frac{1}{1 + f^{2N}} \tag{6}$$

Where N is some positive integer number which is called order of the filter. To obtain the (non-squared) magnitude response, we must take the square root of this:

$$G(f) = \sqrt{\frac{1}{1 + f^{2N}}} \tag{7}$$

Figure 2 shows some magnitude responses of Butterworth filters for various choices of the order N (namely $N = 1, \ldots, 6$). The graph for $N = 1$ has the shallowest slope with lowest values $G(f)$ for frequencies below

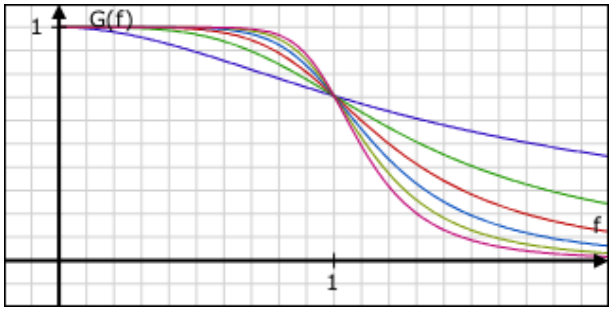


Figure 2: magnitude responses of Butterworth lowpass filters of orders 1...6

the cutoff frequency ($f_c = 1$) and the highest values above the cutoff frequency - thus, compared with our ideal response it performs worst. As the filter order N increases, the actual frequency responses approach our ideal better and better in the sense that they come closer to unity below the cutoff frequency and closer to zero above. They all meet in the common point ($f_c = 1, G(f_c) = G(1) = \sqrt{1/2}$) which we define as the cutoff point. When expressing the magnitude at the cutoff point in decibels, we obtain the value $20 \log_{10}(\sqrt{1/2}) = -3.01...dB$. At the cutoff frequency f_c, the squared magnitude response is given by $G^2(f_c) = (\sqrt{1/2})^2 = 1/2$, and because the magnitude squared is proportional to signal power, this point is sometimes also called half-power point and f_c is called half-power frequency.

The Slope of the Filter

Our perception of frequency and amplitude for audio signals spans a huge range and is (roughly) proportional to the logarithm of the quantity in question (frequency or amplitude). That's why we often draw the magnitude response of a filter in a coordinate system in which both axes are scaled logarithmically - for example the frequency axis in octaves and the magnitude axis in decibels. In the case of Butterworth lowpass filters this leads to the plot in figure 3. This time, the plot has been drawn with actual physical units on the axes and the cutoff frequency has been tuned to $1000Hz$. We observe, that the filters gain in decibels is approximately $0dB$ below the cutoff-frequency, $-3.01dB$ at the cutoff frequency and drops off approximately linearly above the cutoff frequency. This linear behavior of the filter gain in a doubly logarithmic plot makes it meaningful to associate a slope with the filter. The slope defines, by how many

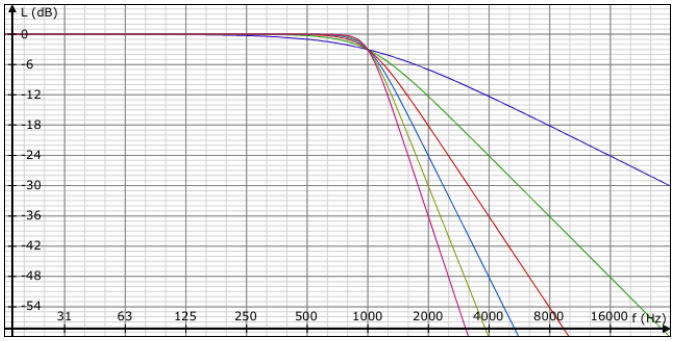


Figure 3: magnitude responses of Butterworth lowpass filters of orders 1...6 with logarithmically scaled axes - observe that the curves approach a constant slope above the cutoff frequency $f_c = 1000Hz$

decibels the filter gain drops off per (logarithmic) frequency interval above the cutoff frequency. The interval is usually measured either in octaves (frequency ratio 1/2) or in decades (frequency ratio 1/10) where in the former case the unit of the slope is decibels per octave (dB/oct) and in the latter case decibels per decade (dB/dec). The slope depends on the order of the filter and as a general rule for lowpass filters, we can say that the slope increases by 6.02 dB/oct for each increment of the filter order N, that is: the slope is given by $6.02 \cdot N$ dB/oct. The same holds for highpass filters. In the case of bandpass and bandreject filters, we will see slopes to both sides of the passband (or rejection band). Bandpass and bandreject filters are often designed from lowpass prototypes. This doubles the order of the filter and yields equal slopes to both sides - thus, we will most often see bandpass filters of even orders and with slopes of $6.02 \cdot \frac{N}{2}$ dB/oct to both sides, where $\frac{N}{2}$ is the order of the prototype lowpass from which the bandpass (or bandreject) was constructed and N is the resulting (even) order of the bandpass.

Bandwidth and the Quality factor 'Q'

For filters where it makes sense to define a lower characteristic frequency f_l and an upper characteristic frequency f_u, it makes sense to define the bandwidth B of the filter as the difference between f_u and f_l:

$$B = f_u - f_l \tag{8}$$

For bandpass and bandreject filters, for example, one would define f_l and f_u at the $-3.01dB$ points (at both sides of the pass- or rejection band). For peaking filters one could define f_l and f_u as the frequencies at which the gain is at the square root of the gain at the peak (but this definition is not as standardized as in the case of bandpass and bandreject filters). In addition, it makes often sense to define a center frequency f_c in these cases, which is often taken to be the geometric mean of f_l and f_u, such that $f_c = \sqrt{f_l \cdot f_u}$. Having defined such a center frequency, we can define the relative bandwidth B_r as the ratio between the bandwidth and the center frequency:

$$B_r = \frac{f_u - f_l}{f_c} \tag{9}$$

Relative bandwidth has the advantage of having always the same width on a plot with a logarithmically scaled frequency axis, which in turn is consistent with the way we perceive audio signals. To make that clear: the relative bandwidth for $f_l = 100Hz, f_u = 200Hz$ is the same as for $f_l = 1000Hz, f_u = 2000Hz$ whereas the absolute bandwidth is not. The 'Q' factor is now simply the reciprocal of the relative

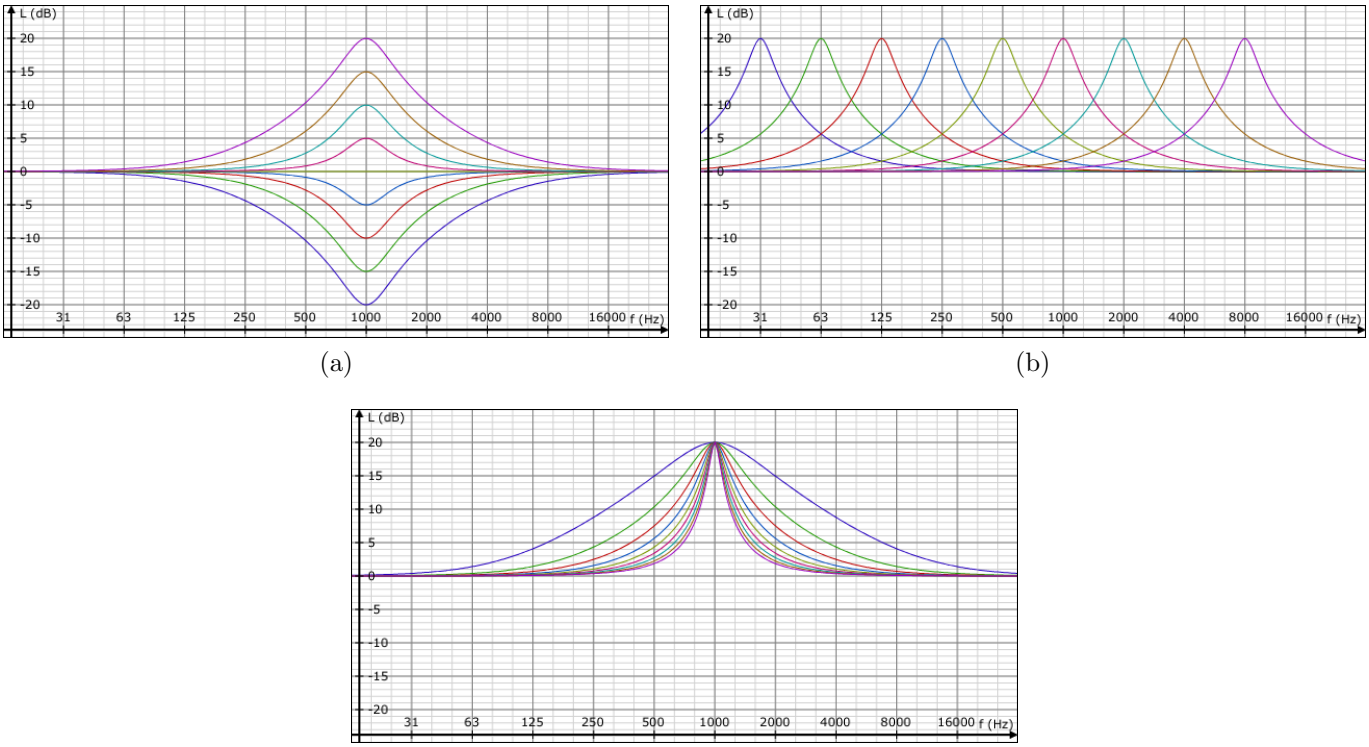
bandwidth:

$$Q = \frac{1}{B_r} = \frac{f_c}{f_u - f_l} \tag{10}$$

that is, the higher the 'Q', the narrower the filter. In general, 'Q' seems to be a bit less intuitive to deal with than relative bandwidth. Moreover, we sometimes find 'Q'-parameters for filters where it is not that clear what the upper and lower characteristic frequencies should be - for example in lowpass filters. In these cases it is likely that some non-standardized definition of 'Q' applies which may vary from one manufacturer to the other. So...well...take 'Q'-parameters always with a grain of salt because there are not always well defined authoritative standards which can be applied (except maybe for bandpass- and bandreject-filters).

Peak- and Shelving-Filters (aka Parametric Equalizers)

As opposed to the original thought behind a filter (to block certain frequencies and let pass others), the main purpose of bell- and shelving-filters is to boost or attenuate certain frequencies. Bell filters boost or attenuate a frequency range around some center frequency. They have unit gain at DC and approach unit gain again as f approaches infinity but somewhere in between, they show a peak or dip in the magnitude response. Their name derives from the bell-shaped magnitude response as seen in figure 4, where some families of magnitude responses are plotted. Shelving filters in contrast boost or attenuate the lower or



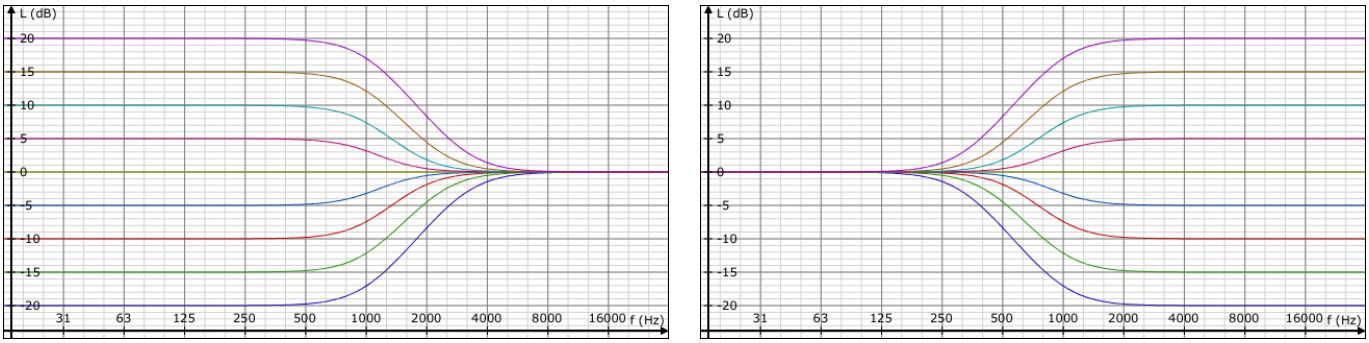
(a) (b)

(c)

Figure 4: magnitude responses of 'bell' or 'peak' filters with (a) different gains, (b) different center frequencies and (c) different bandwidths (or Q factors)

upper frequency range (below or above some corner frequency). Low shelvers have an adjustable gain at

DC and approach unit gain as f approaches infinity, for high shelvers the situation is vice versa. This is depicted in figure 5. Bell and shelving filters are frequently used in audio engineering, for example in



(a) (b)

Figure 5: magnitude responses of low and high shelving filters with a corner frequency of $1kHz$ and different gains

mixing and mastering applications. Here, these filters are better known under the name of parametric equalizers. When only the center frequency and the gain are adjustable by the user but the bandwidth is fixed, then one calls this a semiparametric equalizer. The term 'equalizer' derives from the fact that those filters were originally developed to compensate for non-flat frequency responses of physical signal channels such as radiation through the air or transmission through electric cables, or to compensate for bad acoustics in a concert hall. In this public address application domain, we also often see so called graphic equalizers, which can more or less be seen as a bank of parallel bandpass filters with adjustable level for each band. In the context of music production in a studio environment, i would at any time prefer a parametric equalizer over a graphic one since the parametrics are much more flexible in their use. But that might be a matter of personal taste. It's probably needless to say that in this context equalizers are not only used to compensate for some other undesired frequency response in order to make everything 'flat' - instead they are often used to intentionally and artistically shape the spectrum of some sound or to confine certain instruments to appropriate frequency ranges in order to make up some 'space' in the mix for other instruments.

Resonant Filters and Subtractive Synthesis

Another type of filter which the audio engineer is likely to stumble across, is the resonant filter and in particular the resonant lowpass filter. Filters of that kind are ubiquitous in subtractive synthesizers. A resonant lowpass filter is generally lowpass in its nature but it exhibits a resonant peak in the vicinity of the cutoff frequency. The peakiness of this peak (its height and narrowness, that is) is determined by an additional user parameter, most often called 'resonance' but sometimes also 'Q', 'emphasis' or 'feedback'. In figure 6 we see some plots of typical magnitude responses for different values of the resonance parameter. The depicted magnitude responses in 6 are those of one of the most famous filters in the history of synthesizers, namely the Moog lowpass. This particular filter realizes its resonance by using a series connection of 4 first order lowpasses tuned to the same cutoff frequency and feeding back the sign inverted output signal of the filter to its input. The amount of feedback determines the peakiness of the resonant peak in this case. When the amount of feedback (the factor by which the signal in the feedback path is

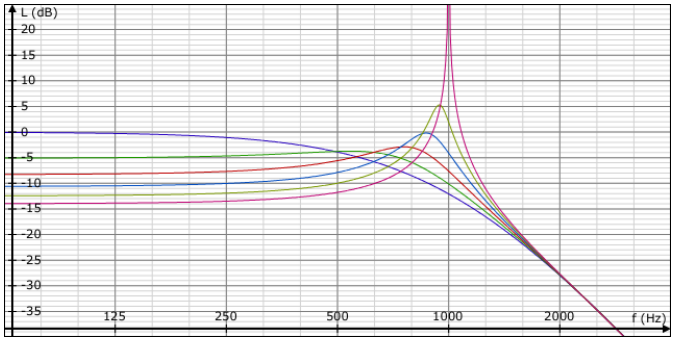


Figure 6: magnitude responses of a 4th order resonant lowpass filter with cutoff frequency $f_c = 1000Hz$ and different resonance values

scaled) reaches 4, the filter begins to oscillate by itself at the cutoff frequency - this phenomenon is called self oscillation. A word about the definition of 'cutoff-frequency' is in order: for technical applications, one most often defines the cutoff frequency as the half-power frequency or equivalently, the $-3.01\ dB$ point. Here in this case, it makes more sense to define the cutoff frequency as the frequency at which the filter resonates and this is the frequency where the gain is at $-12.04\ dB$ here. Why is this? The filter consists of a series connection of 4 identical first order lowpass filters, each contributing a drop in magnitude of $-3.01\ dB$ and a phase shift of $-45°$ at the cutoff frequency. The combined effect is therefore a magnitude drop of $-12.04\ dB$ and a phase shift of $-180°$. A sine wave which is shifted by 180° and inverted is the same sine-wave again, thus feeding back the filter output to the input will lead to an interference which is maximally constructive at the cutoff frequency (because the interfering sines are exactly in phase at that frequency. This explains why the resonance occurs where it occurs. The drop of $-12.04\ dB$ at this frequency on the other hand, explains why the resonance runs into self oscillation with a feedback gain of 4: $-12.04\ dB$ corresponds to a factor of 1/4 and the factor 4 cancels this out to yield unity gain in the overall feedback path at the resonant frequency. Theoretically, feedback gains from 4 upwards would lead to a buildup of a resonance with infinite amplitude. However, because all real filter saturate internally at some level, this won't happen in practice. What we see instead is a stable oscillation but due to the saturating nonlinearities inside the filter, this oscillation is not exactly sinusoidal but also has some overtones. Strictly speaking, when driving the filter into self oscillation and/or saturation, we cannot really apply the theory of linear filters anymore...

Nonlinear Filters

Filters are generally assumed to be linear which is equivalent to the assumption that they multiply the amplitude of an incoming sinusoid by some factor and shift the phase of the incoming sinusoid by some offset - and nothing else. It is this assumption which makes it meaningful to talk about the magnitude and phase response of the filter. When this assumption does not hold true for some signal processing device, the whole concept of magnitude and phase response is not meaningful anymore. In such cases, we can still ask, what the device does to an incoming sinusoid and i have done so in the waveshaping tutorial. But it does not make much sense to ask: 'By which factor does the device amplify (or attenuate) a sinusoid of frequency f and what phase shift does it introduce?' because the answer on this question will not only depend on the sinusoids frequency but also on its amplitude. Moreover, the answer does not tell us, whether there might be sinusoids at other frequencies present in the output signal - which is likely

to happen for nonlinear devices but impossible for linear devices (filters). The assumption of linearity, however, is itself an idealization of the real world. Real filters (in the analog world) are manufactured from electronic building blocks such as resistors, capacitors, transistors, etc. which can reasonably approximate the idealized behavior of a linear filter over a certain range of input amplitudes, but when the amplitude of the signals exceeds this operating range, nonlinear effects take place here as well. A transistor is a good example for this: it operates as a fairly linear amplifier over some range, but clips the signal above this operating range. In purely technical applications these nonlinearites are considered as undesired artifacts due to imperfections of ...mmmhhh... the physical world. In musical applications however, these artifacts can be even beneficial to the 'sound' of a filter. Such nonlinear effects are often hard to analyze (and to model) mathematically but their general effect is to introduce harmonic overtones to incoming sinusoids (or to sinusoids which are generated by the filter itself due to self-oscillation). Refer to the waveshaping tutorial for more details on this. Harmonic overtones, in turn, can add some kind of 'richness' to the sound.

Digest and appetizer for more

We have introduced the basic concept of filters, and went from purely technical applications such as ... well ... filtering (in the sense of letting pass certain frequencies and block others) to somewhat more creative and musical applications such as shaping a spectrum with a parametric equalizer and then we went on further to the entirely creative domain of sound design (filters in subtractive synthesis). A lot more can be (and has been) written about filters which has not been even mentioned here, but for a basic tutorial that much shall suffice. Maybe i'll write a follow up someday, maybe not - at least i want to mention a few more things which have been left out to give you a glimpse, how rich the field of filters actually is and to give you some keywords for further research:

There are allpass filters which have a unit magnitude response at all frequencies - their sole purpose is to introduce frequency dependent phase-shift. Allpass filters are the basic building block of phasers. There are comb filters which are the basic building block of flangers - with feedback, they can resonate at a full harmonic series, and with a further lowpass-filter in the feedback path, basic physical modeling instruments can be built. Filters can be studied in the time domain in terms of their impulse- and step-response. For example, the step response of first order lowpass filters is utilized in slew-rate limiters and analog envelope generators as well as in envelope extractors (which in turn are basic building blocks of dynamics processors such as compressors). Reverberators can be interpreted as filters. The phase response and related aspects such as the group delay can be studied further. One could consider parallel connections of filters (filter banks), which are used for the classical approach to vocoding. In the digital domain there are two fundamentally different paradigms to implement filters, namely FIR and IIR, and the choice which one uses has strong implications on the resulting signals. Then there is the whole question of filter design which was only slightly scribed here in the context of Butterworth lowpasses - filter design is a rich topic on its own and can be quite a challenge mathematically. So ... well ... i reckon there is more than enough stuff about filters out there to write a follow up tutorial. But for now, i need a break from filters.
Good night. Robin.