Chapter 2

Filters

P. Dutilleux, U. Zölzer

2.1 Introduction

The term filter can have a large number of different meanings. In general it can be seen as a way to select certain elements with desired properties from a larger set. Let us focus on the particular field of digital audio effects and consider a signal in the frequency domain. The signal can be seen as a set of partials having different frequencies and amplitudes. The filter will perform a selection of the partials according to the frequencies that we want to reject, retain or emphasize. In other words: the filter will modify the amplitude of the partials according to their frequency. Once implemented, it will turn out that this filter is a linear transformation. As an extension, linear transformations can be said to be filters. According to this new definition of a filter, any linear operation could be said to be a filter but this would go far beyond the scope of digital audio effects. It is possible to demonstrate what a filter is by using one's voice and vocal tract. Utter a vowel, a for example, at a fixed pitch and then utter other vowels at the same pitch. By doing that we do not modify our vocal cords but we modify the volume and the interconnection pattern of our vocal tract. The vocal cords produce a signal with a fixed harmonic spectrum whereas the cavities act as acoustic filters to enhance some portions of the spectrum. We have described filters in the frequency domain here because it is the usual way to consider them but they also have an effect in the time domain. After introducing a filter classification in the frequency domain, we will review typical implementation methods and the associated effects in the time domain.

The various types of filters can be defined according to the following classification:

• Lowpass (LP) filters select low frequencies up to the cut-off frequency f_c and attenuate frequencies higher than f_c .



Figure 2.1 Filter classification.

- Highpass (HP) filters select frequencies higher than f_c and attenuate frequencies below f_c .
- Bandpass (BP) filters select frequencies between a lower cut-off frequency f_{cl} and a higher cut-off frequency f_{ch} . Frequencies below f_{cl} and frequencies higher than f_{ch} are attenuated.
- Bandreject (BR) filters attenuate frequencies between a lower cut-off frequency f_{cl} and a higher cut-off frequency f_{ch} . Frequencies below f_{cl} and frequencies higher than f_{ch} are passed.
- Notch filters attenuate frequencies in a narrow bandwidth around the cut-off frequency f_c .
- **Resonator** filters amplify frequencies in a narrow bandwidth around the cutoff frequency f_c .
- Allpass filters pass all frequencies but modify the phase of the input signal.

Other types of filters (LP with resonance, comb, multiple notch...) can be described as a combination of these basic elements. Here are listed some of the possible applications of these filter types: The lowpass with resonance is very often used in computer music to simulate an acoustical resonating structure; the highpass filter can remove undesired very low frequencies; the bandpass can produce effects such as the imitation of a telephone line or of a mute on an acoustical instrument; the bandreject can divide the audible spectrum into two bands that seem to be uncorrelated. The resonator can be used to add artificial resonances to a sound; the notch is most useful in eliminating annoying frequency components; a set of notch filters, used in combination with the input signal, can produce a phasing effect.

2.2 Basic Filters

2.2.1 Lowpass Filter Topologies

A filter can be implemented in various ways. It can be an acoustic filter, as in the case of the voice. For our applications we would rather use electronic or digital means. Although we are interested in digital audio effects, it is worth having a look at well-established analog techniques because a large body of methods have been developed in the past to design and build analog filters. There are intrinsic design methods for digital filters but many structures can be adapted from existing analog designs. Furthermore, some of them have been tailored for ease of operation within musical applications. It is therefore of interest to gain ideas from these analog designs in order to build digital filters having similar advantages. We will focus on the second-order lowpass filter because it is the most common type and other types can be derived from it. The frequency response of a lowpass filter is shown in Fig. 2.2. The tuning parameters of this lowpass filter are the cut-off frequency f_c and the damping factor ζ . The lower the damping factor, the higher the resonance at the cut-off frequency.

Analog Design, Sallen & Key

Let us remind ourselves of an analog circuit that implements a second-order lowpass filter with the least number of components: the Sallen & Key filter (Figure 2.2).



Figure 2.2 Sallen & Key second-order analog filter and frequency response.

The components (R_1, R_2, C) are related to the tuning parameters as:

$$f_c = \frac{1}{2\pi C \sqrt{R_1 R_2}} \qquad \zeta = \frac{R_1 + R_2}{2\sqrt{R_1 R_2}} \tag{2.1}$$

These relations are straightforward but both tuning coefficients are coupled. It is therefore difficult to vary one while the other remains constant. This structure is therefore not recommended when the parameters are to be tuned dynamically and when low damping factors are desired.

Digital Design, Canonical

The canonical second-order structure, as shown in Fig. 2.3, can be implemented by the difference equation

$$y(n) = b_0 x(n) + b_1 x(n-1) + b_2 x(n-2) -a_1 y(n-1) - a_2 y(n-2).$$
(2.2)



Figure 2.3 Canonical second-order digital filter.

It can be used for any second-order transfer function according to

$$H(z) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}}.$$
(2.3)

In order to modify the cut-off frequency or the damping factor, all 5 coefficients have to be modified. They can be computed from the specification in the frequency plane or from a prototype analog filter. One of the methods that can be used is based on the bilinear transform [DJ85]. The following set of formulas compute the coefficients for a lowpass filter:

$$\begin{aligned} f_c & \text{analog cut-off frequency} \\ \zeta & \text{damping factor} \\ f_s & \text{sampling frequency} \\ C &= 1/[\tan(\pi f_c/f_s)] \end{aligned}$$

$$(2.4)$$

$$b_{0} = 1/(1 + 2\zeta C + C^{2})$$

$$b_{1} = 2b_{0}$$

$$b_{2} = b_{0}$$

$$a_{1} = 2b_{0}(1 - C^{2})$$

$$a_{2} = b_{0}(1 - 2\zeta C + C^{2})$$

(2.5)

This structure has the advantage that it requires very few elementary operations to process the signal itself. It has unfortunately some severe drawbacks. Modifying the filter tuning (f_c, ζ) involves rather complex computations. If the parameters are varied continuously, the complexity of the filter is more dependent on the coefficient computation than on the filtering process itself. Another drawback is the poor signal to noise ratio for low frequency signals. Other filter structures are available that cope with these problems. We will again review a solution in the analog domain and its counterpart in the digital domain.

State Variable Filter, Analog

For musical applications of filters one wishes to have an independent control over the cut-off frequency and the damping factor. A technique originating from the analog computing technology can solve our problem. It is called the state variable filter (Figure 2.4). This structure is more expensive than the Sallen & Key but has independent tuning components (R_f, R_{ζ}) for the cut-off frequency and the damping factors:

$$f_c = 1/(2\pi R_f C)$$
 $\zeta = R/[2(R+R_{\zeta})]$ (2.6)

Furthermore, it provides simultaneously three types of outputs: lowpass, highpass and bandpass.



Figure 2.4 Analog state variable filter.

State Variable Filter, Digital

The state variable filter has a digital implementation, as shown in Fig. 2.5 [Cha80], where

x(n)	input signal
$y_l(n)$	lowpass output
$y_b(n)$	bandpass output
$y_h(n)$	highpass output

and the difference equations for the output signals are given by



Figure 2.5 Digital state variable filter.

$$y_l(n) = F_1 y_b(n) + y_l(n-1) y_b(n) = F_1 y_h(n) + y_b(n-1) y_h(n) = x(n) - y_l(n-1) - Q_1 y_b(n-1).$$
(2.7)

With tuning coefficients F_1 and Q_1 , related to the tuning parameters f_c and ζ as:

$$F_1 = 2\sin(\pi f_c/f_s)$$
 $Q_1 = 2\zeta$ (2.8)

it can be shown that the lowpass transfer function is:

$$r = F_1 \qquad q = 1 - F_1 Q_1$$

$$H(z) = \frac{r^2}{1 + (r^2 - q - 1)z^{-1} + qz^{-2}}$$
(2.9)

This structure is particularly effective not only as far as the filtering process is concerned but above all because of the simple relations between control parameters and tuning coefficients. One should consider the stability of this filter, because at higher cut-off frequencies and larger damping factors it becomes unstable. A "usability limit" given by $F_1 < 2 - Q_1$ assures the stable operation of the state variable implementation [Dut91, Die00]. In most musical applications however it is not a problem because the tuning frequencies are usually small compared to the sampling frequency and the damping factor is usually set to small values [Dut89a, Dat97]. This filter has proven its suitability for a large number of applications. The nice properties of this filter have been exploited to produce endless glissandi out of natural sounds and to allow smooth transitions between extreme settings [Dut89b, m-Vas93]. It is also used for synthesizer applications [Die00]. We have considered here two different digital filter structures. More are available and each has its advantages and drawbacks. An optimum choice can only be made in agreement with the application [Zöl97].

Normalization

Filters are usually designed in the frequency domain and we have seen that they have an action also in the time domain. Another correlated impact lies in the loudness of the filtered sounds. The filter might produce the right effect but the result might be useless because the sound has become too weak or too strong. The method of compensating for these amplitude variations is called normalization. Usual normalization methods are called L_1 , L_2 and L_{∞} [Zöl97]. L_1 is used when the filter should never be overloaded under any circumstances. This is overkill most of the time. L_2 is used to normalize the loudness of the signal. It is accurate for broadband signals and fits many practical musical applications. L_{∞} actually normalizes the frequency response. It is best when the signal to filter is sinusoidal or periodical. With a suitable normalization scheme the filter can prove to be very easy to handle whereas with the wrong normalization, the filter might be rejected by musicians because they cannot operate it. The normalization of the state variable filter has been studied in [Dut91] where several implementation schemes are proposed that lead to an effective implementation. In practice, a first-order lowpass filter that processes the input signal will perform the normalization in f_c and an amplitude correction in $\sqrt{\zeta}$ will normalize in ζ (Figure 2.6). This normalization scheme allows us to operate the filter with damping factors down to 10^{-4} where the filter gain reaches about 74 dB at f_c .



Figure 2.6 L_2 -normalization in f_c and ζ for the state variable filter.

Sharp Filters

Apart from FIR filters (see section 2.2.3), we have so far only given examples of second-order filters. These filters are not suitable for all applications. On the one hand, smooth spectral modifications are better realized by using first-order filters. On the other hand, processing two signal components differently that are close in frequency, or imitating the selectivity of our hearing system calls for higher order filters. FIR filterss can offer the right selectivity but again, they will not be easily tuned. Butterworth filters have attractive features in this case. Such filters are optimized for a flat frequency response until f_c and yield a 6n dB/octave attenuation for frequencies higher than f_c . Filters of order 2n can be built out of n second-order sections, All sections are tuned to the same cut-off frequency f_c but each section has a different damping factor ζ (Table 2.1) [LKG72].

These filters can be implemented accurately in the canonical second-order digital filter structure but modifying the tuning frequency in real time can lead to temporary instabilities. The state variable structure is less accurate for high tuning frequencies (i.e. $f_c > f_s/10$) but allows faster tuning modifications. A bandpass filter comprising a 4th-order highpass and a 4th-order lowpass was implemented

n	ζ of second-order sections				
2	0.707				
4	0.924	0.383			
6	0.966	0.707	0.259		
8	0.981	0.831	0.556	0.195	
10	0.988	0.891	0.707	0.454	0.156

 Table 2.1 Damping factors for Butterworth filters.

and used to imitate a fast varying mute on a trombone [Dut91]. Higher order filters (up to about 16) are useful to segregate spectral bands or even individual partials within complex sounds.

Behavior in the Time Domain

We so far considered the action of the filters in the frequency domain. We cannot forget the time domain because it is closely related to it. Narrow bandpass filters, or resonant filters even more, will induce long ringing time responses. Filters can be optimized for their frequency response or time response. It is easier to grasp the time behavior of FIRs than IIRs. FIRs have the drawback of a time delay that can impair the responsiveness of digital audio effects.

2.2.2 Parametric AP, LP, HP, BP and BR Filters

Introduction

In this subsection we introduce a special class of parametric filter structures for allpass, lowpass, highpass, bandpass and bandreject filter functions. Parametric filter structures denote special signal flow graphs where a coefficient inside the signal flow graph directly controls the cut-off frequency and bandwidth of the corresponding filter. These filter structures are easily tunable by changing only one or two coefficients. They play an important role for real-time control with minimum computational complexity.

Signal Processing

The basis for parametric first- and second-order IIR filters is the first- and secondorder allpass filter. We will first discuss the first-order allpass and show simple lowand highpass filters, which consist of a tunable allpass filter together with a direct path. First-order allpass. A first-order allpass filter is given by the transfer function

$$A(z) = \frac{z^{-1} + c}{1 + cz^{-1}}$$
(2.10)

$$c = \frac{\tan(\pi f_c/f_s) - 1}{\tan(\pi f_c/f_s) + 1}.$$
 (2.11)

The magnitude/phase response and the group delay of a first-order allpass are shown in Fig. 2.7. The magnitude response is equal to one and the phase response is approaching -180 degrees for high frequencies. The group delay shows the delay of the input signal in samples versus frequency. The coefficient c in (2.10) controls the cut-off frequency of the allpass, where the phase response passes -90 degrees (see Fig. 2.7).



Figure 2.7 First-order allpass filter with $f_c = 0.1 \cdot f_s$.

From (2.10) we can derive the corresponding difference equation

$$y(n) = cx(n) + x(n-1) - cy(n-1),$$
 (2.12)

which leads to the block diagram in Fig. 2.8. The coefficient c occurs twice in this signal flow graph and can be adjusted according to (2.11) to change the cut-off frequency. A variant allpass structure with only one delay element is shown in the right part of Fig. 2.8. It is implemented by the difference equations

$$x_h(n) = x(n) - cx_h(n-1)$$
 (2.13)

$$y(n) = cx_h(n) + x_h(n-1).$$
 (2.14)



Figure 2.8 Block diagram for a first-order allpass filter.

The resulting transfer function is equal to (2.10). For simple implementations a table with a number of coefficients for different cut-off frequencies is sufficient, but even for real-time applications this structure offers very few computations. In the following we use this first-order allpass filter to perform low/highpass filtering.

First-order low/highpass. A first-order lowpass filter can be achieved by adding or subtracting (+/-) the input signal from the output signal of a first-order allpass filter. As the output signal of the first-order allpass filter has a phase shift of -180 degrees for high frequencies, this operation leads to low/highpass filtering. The transfer function of a low/highpass filter is then given by

$$H(z) = \frac{1}{2} (1 \pm A(z)) \quad (LP/HP + /-)$$
(2.15)

$$A(z) = \frac{z^{-1} + c}{1 + cz^{-1}}$$
(2.16)

$$c = \frac{\tan(\pi f_c/f_s) - 1}{\tan(\pi f_c/f_s) + 1},$$
(2.17)

where a tunable first-order allpass A(z) with tuning parameter c is used. The plus sign (+) denotes the lowpass operation and the minus sign (-) the highpass operation. A block diagram in Fig. 2.9 represents the operations involved in performing the low/highpass filtering. The allpass filter can be implemented by the difference equation (2.12) as shown in Fig. 2.8.



Figure 2.9 Block diagram of a first-order low/highpass filter.

The magnitude/phase response and group delay are illustrated for low- and highpass filtering in Fig. 2.10. The -3dB point of the magnitude response for lowpass and



Figure 2.10 First-order low/highpass filter with $f_c = 0.1 f_s$.

highpass is passed at the cut-off frequency. With the help of the allpass subsystem in Fig. 2.9 tunable low- and highpass systems are achieved.

Second-order allpass. The implementation of tunable bandpass and bandreject filters can be achieved with a second-order allpass filter. The transfer function of a second-order allpass filter is given by

$$A(z) = \frac{-c + d(1-c)z^{-1} + z^{-2}}{1 + d(1-c)z^{-1} - cz^{-2}}$$
(2.18)

$$c = \frac{\tan(\pi f_b/f_s) - 1}{\tan(\pi f_b/f_s) + 1}$$
(2.19)

$$d = -\cos(2\pi f_c/f_s).$$
 (2.20)

The parameter d adjusts the cut-off frequency and the parameter c the bandwidth. The magnitude/phase response and the group delay of a second-order allpass are shown in Fig. 2.7. The magnitude response is again equal to one and the phase response approaches -360 degrees for high frequencies. The cut-off frequency ω_C determines the point on the phase curve, where the phase response passes -180 degrees. The width or slope of the phase transition around the cut-off frequency is controlled by the bandwidth parameter ω_B . From (2.18) the corresponding difference equation

$$y(n) = -cx(n) + d(1-c)x(n-1) + x(n-2) -d(1-c)y(n-1) + cy(n-2)$$
(2.21)

can be derived, which leads to the block diagram in Fig. 2.12. The cut-off frequency is controlled by the coefficient d and the bandwidth by coefficient c.



Figure 2.11 Second-order allpass filter with $f_c = 0.1 f_s$ and $f_b = 0.022 f_s$.



Figure 2.12 Block diagram for a second-order allpass filter.

2.2 Basic Filters

Second-order bandpass/bandreject. Second-order bandpass and bandreject filters can be described by the following transfer function

$$H(z) = \frac{1}{2} [1 \mp A(z)] \quad (BP/BR - /+)$$
(2.22)

$$A(z) = \frac{-c + d(1-c)z^{-1} + z^{-2}}{1 + d(1-c)z^{-1} - cz^{-2}}$$
(2.23)

$$c = \frac{\tan(\pi f_b/f_s) - 1}{\tan(2\pi f_b/f_s) + 1}$$
(2.24)

$$d = -\cos(2\pi f_c/f_s), \qquad (2.25)$$

where a tunable second-order allpass A(z) with tuning parameters c and d is used. The plus sign (+) denotes the bandpass operation and the minus sign (-) the bandreject operation. The block diagram in Fig. 2.13 shows the bandpass and bandreject filter implementation based on a second-order allpass subsystem, which can be implemented by the signal flow graph of Fig. 2.12. The magnitude/phase response and group delay are illustrated in Fig. 2.14 for both filter types.



Figure 2.13 Second-order bandpass and bandreject filter.

Second-order low/highpass filters. The coefficients for second-order lowand highpass filters given by the transfer function of (2.3) are shown in Table 2.2. A control of single coefficients for adjusting the cut-off frequency is not possible. A complete set of coefficients is necessary, if the cut-off frequency is changed. The implementation of these second-order low- and highpass filters can be achieved by the difference equation (2.2) and the filter structure in Fig. 2.3.

	lowpass (se	cond-order) with	$1 K = \tan\left(\pi f_c/f_s\right)$)
b_0	b ₁	b_2	a_1	a_2
$\frac{\frac{K^2}{1+\sqrt{2}K+K^2}}$	$\frac{2K^2}{1+\sqrt{2}K+K^2}$	$\frac{K^2}{1+\sqrt{2}K+K^2}$	$\frac{2(K^2-1)}{1+\sqrt{2}K+K^2}$	$\frac{1-\sqrt{2}K+K^2}{1+\sqrt{2}K+K^2}$
	highpass (se	econd-order) wit	$h K = \tan\left(\pi f_c/f_s\right)$	s)
b ₀	b_1	b ₂	a_1	a_2
$\frac{1}{1+\sqrt{2}K+K^2}$	$\frac{-2}{1+\sqrt{2}K+K^2}$	$\frac{1}{1+\sqrt{2}K+K^2}$	$\frac{2(K^2-1)}{1+\sqrt{2}K+K^2}$	$\frac{1-\sqrt{2}K+K^2}{1+\sqrt{2}K+K^2}$

Table 2.2 Filter coefficients for second-order lowpass/highpass filters [Zöl97].



Figure 2.14 Second-order bandpass/bandreject filter with $f_c = 0.1 f_s$ and $f_b = 0.022 f_s$.

Series connection of first- and second-order filters. If several filters are necessary for spectrum shaping, a series connection of first- and second-order filters

$$H_{1\text{st-order}}(z) = \frac{b_0 + b_1 z^{-1}}{1 + a_1 z^{-1}}$$
(2.26)

$$H_{\text{2nd-order}}(z) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}}$$
(2.27)

is performed, which is given by the product of the single transfer functions

$$H(z) = \frac{Y(z)}{X(z)} = H_1(z) \cdot H_2(z) \cdot H_3(z).$$
(2.28)

A series connection of three stages is shown in Fig. 2.15. The resulting difference



Figure 2.15 Series connection of first/second-order stages.

equation can be split into three difference equations as given by

stage 3

$$y_1(n) = b_0^{s_1} x(n) + b_1^{s_1} x(n-1) - a_1^{s_1} y_1(n-1)$$
(2.29)
stage 2

$$\begin{aligned} x_2(n) &= y_1(n) \\ y_2(n) &= b_0^{s^2} x_2(n) + b_1^{s^2} x_2(n-1) + b_2^{s^2} x_2(n-2) \end{aligned}$$
 (2.30)

$$x_3(n) = y_2(n)$$
 (2.32)

$$y(n) = b_0^{s_3} x_3(n) + b_1^{s_3} x_3(n-1) + b_2^{s_3} x_3(n-2) -a_1^{s_3} y(n-1) - a_2^{s_3} y(n-2).$$
(2.33)

Musical Applications

The simple control of the cut-off frequency and the bandwidth of these parametric filters leads to very efficient implementations for real-time audio applications. Only second-order low- and highpass filters need the computation of a complete set of coefficients. The series connection of these filters can be done very easily as shown in the previous paragraph.

2.2.3 FIR Filters

Introduction

The digital filter that we have seen before is said to have an infinite impulse response. Because of the feedback loops within the structure, an input sample will excite an output signal whose duration is dependent on the tuning parameters and can extend over a fairly long period of time. There are other filter structures without



Figure 2.16 Finite Impulse Response Filter.

feedback loops (Figure 2.16). These are called finite impulse response filters (FIR), because the response of the filter to a unit impulse lasts only for a fixed period of time. These filters allow the building of sophisticated filter types where strong attenuation of unwanted frequencies or decomposition of the signal into several frequency bands is necessary. They typically require more computing power than IIR structures to achieve similar results but when they are implemented in the form known as fast convolution they become competitive, thanks to the FFT algorithm. It is rather unwieldy to tune these filters interactively. As an example, let us briefly consider the vocoder application. If the frequency bands are fixed, then the FIR implementation can be most effective but if the frequency bands have to be subtly tuned by a performer, then the IIR structures will certainly prove superior [Mai97]. However, the filter structure in Fig. 2.16 finds widespread applications for headrelated transfer functions and the approximation of first room reflections, as will be shown in Chapter 6. For applications where the impulse response of a real system has been measured, the FIR filter structure can be used directly to simulate the measured impulse response.

Signal Processing

The output/input relation of the filter structure in Fig. 2.16 is described by the difference equation

$$y(n) = \sum_{i=0}^{N-1} b_i \cdot x(n-i)$$
(2.34)

$$= b_0 x(n) + b_1 x(n-1) + \dots + b_{N-1} x(n-N+1), \qquad (2.35)$$

which is a weighted sum of delayed input samples. If the input signal is a unit impulse $\delta(n)$, which is one for n = 0 and zero for $n \neq 0$, we get the impulse response of the system according to

$$h(n) = \sum_{i=0}^{N-1} b_i \cdot \delta(n-i).$$
(2.36)

A graphical illustration of the impulse response of a 5-tap FIR filter is shown in Fig. 2.17. The Z-transform of the impulse response gives the transfer function

$$H(z) = \sum_{i=0}^{N-1} b_i \cdot z^{-i}$$
(2.37)



Figure 2.17 Impulse response of an FIR filter.

and with $z = e^{j\Omega}$ the frequency response

$$H(e^{j\Omega}) = b_0 + b_1 e^{-j\Omega} + b_2 e^{-j2\Omega} + \dots + b_{N-1} \cdot e^{-j(N-1)\Omega}$$
(2.38)
with $\Omega = 2\pi f/f_s = \omega T.$

Filter design. The filters already described such as LP, HP, BP and BR are also possible with FIR filter structures (see Fig. 2.18). The N coefficients b_0, \ldots, b_{N-1} of a nonrecursive filter have to be computed by special design programs, which are discussed in all DSP text books. The N coefficients of the impulse response can be designed to yield a linear phase response, when the coefficients fulfill certain symmetry conditions. The simplest design is based on the inverse discrete-time Fourier transform of the ideal lowpass filter, which leads to the impulse response

$$h(n) = \frac{2f_c}{f_s} \cdot \frac{\sin\left[2\pi f_c/f_s\left(n - \frac{N-1}{2}\right)\right]}{2\pi f_c/f_s\left(n - \frac{N-1}{2}\right)}, n = 0, \dots, N-1.$$
(2.39)

To improve the frequency response this impulse response can be weighted by an appropriate window function like Hamming or Blackman according to

$$h_B(n) = h(n) \cdot w_B(n) \tag{2.40}$$

$$h_H(n) = h(n) \cdot w_H(n) \tag{2.41}$$

$$n=0,1,\ldots,N-1.$$

If a lowpass filter is designed and an impulse response $h_{LP}(n)$ is derived, a frequency transformation of this lowpass filter leads to highpass, bandpass and bandreject filters (see Fig. 2.18).



Figure 2.18 Frequency transformations: LP and frequency transformations to BP and HP.

Frequency transformations are performed in the time domain by taking the lowpass impulse response $h_{LP}(n)$ and computing the following equations:

• LP-HP

$$h_{HP}(n) = h_{LP}(n) \cdot \cos\left[\pi\left(n - \frac{N-1}{2}\right)\right] \quad n = 0, \dots, N-1$$
 (2.42)

• LP-BP

$$h_{BP}(n) = 2h_{LP}(n) \cdot \cos\left[2\pi \frac{f_c}{f_s}\left(n - \frac{N-1}{2}\right)\right] \quad n = 0, \dots, N-1 \quad (2.43)$$

• LP-BR

$$h_{BR}(n) = \delta\left(n - \frac{N-1}{2}\right) - h_{BP}(n) \quad n = 0, \dots, N-1.$$
 (2.44)

Another simple FIR filter design is based on the FFT algorithm and is called *fre-quency sampling*. Design examples for audio processing with this design technique can be found in [Zöl97].

Musical Applications

If linear phase processing is required, FIR filtering offers magnitude equalization without phase distortions. They allow real-time equalization by making use of the frequency sampling design procedure [Zöl97] and are attractive equalizer counterparts to IIR filters, as shown in [McG93]. A discussion of more advanced FIR filters for audio processing can be found in [Zöl97].

2.2.4 Convolution

Introduction

Convolution is a generic signal processing operation like addition or multiplication. In the realm of computer music it has nevertheless the particular meaning of imposing a spectral or temporal structure onto a sound. These structures are usually not defined by a set of few parameters, such as the shape or the time response of a filter, but given by a signal which lasts typically a few seconds or more. Although convolution has been known and used for a very long time in the signal processing community, its significance for computer music and audio processing has grown with the availability of fast computers that allow long convolutions to be performed in a reasonable period of time.

Signal Processing

We could say in general that the convolution of two signals means filtering the one with the other. There are several ways of performing this operation. The straight-forward method is a direct implementation in a FIR filter structure but it is computationally very ineffective when the impulse response is several thousand samples long. Another method, called the fast convolution, makes use of the FFT algorithm to dramatically speed up the computation. The drawback of the fast convolution is that it has a processing delay equal to the length of two FFT blocks, which is objectionable for real-time applications whereas the FIR method has the advantage of providing a result immediately after the first sample has been computed. In order to take advantage of the FFT algorithm while keeping the processing delay to a minimum, low-latency convolution schemes have been developed which are suitable for real-time applications [Gar95, MT99].

The result of convolution can be interpreted in both the frequency and time domains. If a(n) and b(n) are the two convolved signals, the output spectrum will be given by the product of the two spectra $S(f) = A(f) \cdot B(f)$. The time interpretation derives from the fact that if b(n) is a pulse at time k, we will obtain a copy of a(n)shifted at time k_0 , i.e. s(n) = a(n-k). If b(n) is a sequence of pulses, we will obtain a copy of a(n) in correspondence to every pulse, i.e. a rhythmic, pitched, or reverberated structure, depending on the pulse distance. If b(n) is pulse-like, we obtain the same pattern with a filtering effect. In this case b(n) should be interpreted as an impulse response. Thus convolution will result in subtractive synthesis, where the frequency shape of the filter is determined by a real sound. For example the convolution with a bell sound will be heard as filtered by the resonances of the bell. In fact the bell sound is generated by a strike on the bell and can be considered as the impulse response of the bell. In this way we can simulate the effect of a sound hitting a bell, without measuring the resonances and designing the filter. If both sounds a(n) and b(n) are complex in time and frequency, the resulting sound will be blurred and will tend to lack the original sound's character. If both sounds are of long duration and each has a strong pitch and smooth attack, the result will contain both pitches and the intersection of their spectra.

Musical Applications

The sound example "quasthal" [m-quasthal] illustrates the use of the impulse response as a way of characterizing a linear system. In this example, a spoken word is convolved with a series of impulses which are derived from measurements of 2 loudspeakers and of 3 rooms. The first loudspeaker, a small studio monitor, alters at least the original sound. The second loudspeaker, a spherical one, colors the sound strongly. When the sound is convolved with the impulse responses of a room, it is projected in the corresponding virtual auditory space [DMT99]. A diffuse reverberation can be produced by convolving with broad band noise having a sharp attack and exponentially decreasing amplitude. Another example features a tuba glissando convolved by a series of snare-drum strokes. The tuba is transformed in something like a tibetan trumpet playing in the mountains. Each stroke of the snare drum produces a copy of the tuba sound. Since each stroke is noisy and broadband, it acts like a reverberator. The series of strokes acts like several diffusing boundaries and produces the type of echo that can be found in natural landscapes [DMT99, m-tubg5sna].

The convolution can be used to map a rhythm pattern onto a sampled sound. The rhythm pattern can be defined by positioning a unit impulse at each desired time within a signal block. The convolution of the input sound with the time pattern will produce copies of the input signal at each of the unit impulses. If the unit impulse is replaced by a more complex sound, each copy will be modified in its timbre and in its time structure. If a snare drum stroke is used, the attacks will be smeared and some diffusion will be added [m-gendsna]. The convolution has an effect both in the frequency and in the time domain. Take a speech sound with sharp frequency resonances and a rhythm pattern defined by a series of snare-drum strokes. Each word will appear with the rhythm pattern, also the rhythm pattern will be heard in each word with the frequency resonances of the initial speech sound [m-chu5sna].

The convolution as a tool for musical composition has been investigated by composers such as Horacio Vaggione [m-Vag96, Vag98] and Curtis Roads [Roa97]. Because the convolution has a combined effect in the time and frequency domains, some expertise is necessary to foresee the result of the combination of two sounds.

2.3 Equalizers

Introduction and Musical Applications

In contrast to low/highpass and bandpass/reject filters, which attenuate the audio spectrum above or below a cut-off frequency, equalizers shape the audio spectrum by enhancing certain frequency bands while others remain unaffected. They are built by a series connection of first- and second-order shelving and peak filters, which are controlled independently (see Fig. 2.19). Shelving filters boost or cut the low or high frequency bands with the parameter cut-off frequency f_c and gain G. Peak filters boost or cut mid-frequency bands with parameters cut-off frequency f_c , bandwidth f_b and gain G. One often used filter type is the constant Q peak filter. The Q factor is defined by the ratio of the bandwidth to cut-off frequency $Q = \frac{f_b}{f_c}$. The cut-off frequency of peak filters are then tuned, while keeping the Q factor constant. This means that the bandwidth is increased when the cut-off frequency is increased and vice versa. Several proposed digital filter structures for shelving and peak filters can be found in the literature [Whi86, RM87, Dut89a, HB93, Bri94, Orf96, Orf97, Zöl97].

Applications of these parametric filters can be found in parametric equalizers, octave equalizers ($f_c=31.25$, 62.5, 125, 250, 500, 1000, 2000, 4000, 8000, 16000 Hz) and all kinds of equalization devices in mixing consoles, outboard equipment and foot pedal controlled stomp boxes.



Figure 2.19 Series connection of shelving and peak filters.

2.3.1 Shelving Filters

First-order Design

First-order low/high frequency shelving filters $[Z\ddot{o}197]$ can be described by the transfer function

$$H(z) = 1 + \frac{H_0}{2} \left[1 + \pm A(z) \right] \quad (\text{LF/HF +/-}) \tag{2.45}$$

with the first-order allpass

$$A(z) = \frac{z^{-1} + a_{B/C}}{1 + a_{B/C} z^{-1}}.$$
(2.46)

The block diagram in Fig. 2.20 shows a first-order low/high-frequency shelving



Figure 2.20 First-order low/high-frequency shelving filter.

filter, which leads to the following difference equations:

$$y_1(n) = a_{B/C}x(n) + x(n-1) - a_{B/C}y_1(n-1)$$
(2.47)

$$y(n) = \frac{H_0}{2} [x(n) \pm y_1(n)] + x(n). \qquad (2.48)$$

The gain G in dB for low/high frequencies can be adjusted by the parameter

$$H_0 = V_0 - 1$$
, with $V_0 = 10^{G/20}$. (2.49)

The cut-off frequency parameter a_B for boost and a_C for cut can be calculated as

$$a_B = \frac{\tan((\pi f_c/f_s) - 1)}{\tan((\pi f_c/f_s) + 1)}, \omega_c = 2\pi f_c$$
(2.50)

$$a_C = \frac{\tan((\pi f_c/f_s) - V_0)}{\tan((\pi f_c/f_s) + V_0)}.$$
(2.51)

The cut-off frequency parameters for boost and cut for a first-order high-frequency shelving filter [Zöl97] are calculated by

$$a_B = \frac{\tan(\pi f_c/f_s) - 1}{\tan((\pi f_c/f_s) + 1)}$$
(2.52)

$$a_C = \frac{V_0 \tan((\pi f_c/f_s) - 1)}{V_0 \tan((\pi f_c/f_s) + 1)}.$$
(2.53)

Magnitude responses for a low-frequency shelving filter are illustrated in the left part of Fig. 2.21 for several cut-off frequencies and gain factors. The slope of the frequency curves for these first-order filters are with 6 dB per octave.

Second-order Design

For several applications especially in advanced equalizer designs the slope of the shelving filter is further increased by second-order transfer functions. Design formulas for second-order shelving filters are given in Table 2.3 from [Zöl97]. Magnitude responses for second-order low/high frequency shelving filters are illustrated in the right part of Fig. 2.21 for two cut-off frequencies and several gain factors.

2.3.2 Peak Filters

A second-order peak filter [Zöl97] is given by the transfer function

$$H(z) = 1 + \frac{H_0}{2} \left[1 - A_2(z) \right], \qquad (2.54)$$

where

$$A_2(z) = \frac{-a_B + (d - da_B)z^{-1} + z^{-2}}{1 + (d - da_B)z^{-1} - a_B z^{-2}}$$
(2.55)



Table 2.3 Second-order shelving filter design with $K = \tan(\pi f_c/f_s)$ [Zöl97].

Figure 2.21 Frequency responses for first-order and second-order shelving filters.

is a second-order allpass filter. The block diagram in Fig. 2.22 shows the secondorder peak filter, which leads to the following difference equations:

$$y_1(n) = -a_{B/C}x(n) + d(1 - a_{B/C})x(n-1) + x(n-2) -d(1 - a_{B/C})y_1(n-1) + a_{B/C}y_1(n-2)$$
(2.56)

$$y(n) = \frac{H_0}{2} [x(n) - y_1(n)] + x(n).$$
(2.57)



Figure 2.22 Second-order peak filter.

The center/cut-off frequency parameter d and the coefficient H_0 are given by

$$d = -\cos(2\pi f_c/f_s) \tag{2.58}$$

$$V_0 = H(e^{j2\pi f_c/f_s}) = 10^{G/20}$$
(2.59)

$$H_0 = V_0 - 1. (2.60)$$

The bandwidth f_b is adjusted through the parameters a_B and a_C for boost and cut and are given by

$$a_B = \frac{\tan(\pi f_b/f_s) - 1}{\tan(\pi f_b/f_s) + 1}$$
(2.61)

$$a_C = \frac{\tan(\pi f_b/f_s) - V_0}{\tan(\pi f_b/f_s) + V_0}.$$
(2.62)

This peak filter offers almost independent control of all three musical parameters center/cut-off frequency, bandwidth and gain. Another design approach from [Zöl97] shown in Table 2.4 allows direct computation of the five coefficients for a second-order transfer function as given in the difference equation (2.2).

Frequency responses for several settings of a peak filter are shown in Fig. 2.23. The left part shows a variation of the gain with a fixed center frequency and bandwidth. The right part show for fixed gain and center frequency a variation of the bandwidth or Q factor.



Table 2.4 Peak filter design with $K = \tan(\pi f_c/f_s)$ [Zöl97].

Figure 2.23 Frequency responses second-order peak filters.

2.4 Time-varying Filters

The parametric filters discussed in the previous sections allow the time-varying control of the filter parameters gain, cut-off frequency and bandwidth or Q factor. Special applications of time-varying audio filters will be shown in the following.

2.4.1 Wah-wah Filter

The wah-wah effect is produced mostly by foot-controlled signal processors containing a bandpass filter with variable center/resonant frequency and a small bandwidth. Moving the pedal back and forth changes the bandpass cut-off/center frequency. The "wah-wah" effect is then mixed with the direct signal as shown in Fig. 2.24. This effect leads to a spectrum shaping similar to speech and produces a speech like "wah-wah" sound. If the variation of the center frequency is controlled by the



Figure 2.24 Wah-wah: time-varying bandpass filter.

input signal, a low-frequency oscillator is used to change the center frequency. Such an effect is called an auto-wah filter. If the effect is combined with a low-frequency amplitude variation, which produces a tremolo, the effect is denoted a tremolo-wah filter. Replacing the unit delay in the bandpass filter by an M tap delay leads to the M-fold wah-wah filter [Dis99], which is shown in Fig. 2.25. M bandpass filters are spread over the entire spectrum and simultaneously change their center frequency. When a white noise input signal is applied to an M-fold wah-wah filter, a spectrogram of the output signal shown in Fig. 2.26 illustrates the periodic enhancement of the output spectrum. Table 2.5 contains several parameter settings for different effects.



Figure 2.25 M-fold wah-wah filter.

	M	Q^{-1}/f_m	$\Delta\omega/2\pi$
Wah-Wah	1	-/3kHz	200 Hz
M-fold Wah-Wah	5-20	0.5/-	200-500Hz
Bell effect	100	0.5/-	100 Hz

Table 2.5 Effects with M-fold wah-wah filter [Dis99].

2.4.2 Phaser

The previous effect relies on varying the center frequency of a bandpass filter. Another effect uses notch filters: *phasing*. A set of notch filters, that can be realized as a cascade of second-order IIR sections, is used to process the input signal. The output of the notch filters is then combined with the direct sound. The frequencies of the notches are slowly varied using a low-frequency oscillator (Figure 2.27) [Smi84]. "The strong phase shifts that exist around the notch frequencies combine



Figure 2.26 Spectrogram of output signal of a time-varying M-fold wah-wah filter [Dis99].

with the phases of the direct signal and cause phase cancellations or enhancements that sweep up and down the frequency axis" [Orf96]. Although this effect does not rely on a delay line, it is often considered to go along with delay-line based effects because the sound effect is similar to that of *flanging*. An extensive discussion on this topic is found in [Str83]. A different phasing approach is shown in Figure 2.28. The notch filters have been replaced by second-order allpass filters with time-varying center frequencies. The cascade of allpass filters produces time-varying phase shifts which lead to cancellations and amplifications of different frequency bands when used in the feedforward and feedback configuration.



Figure 2.27 Phasing.



Figure 2.28 Phasing with time-varying allpass filters.

2.4.3 Time-varying Equalizers

• Time-varying octave bandpass filters, as shown in Fig. 2.29, offer the possibility of achieving wah-wah-like effects. The spectrogram of the output signal in Fig. 2.30 demonstrates the octave spaced enhancement of this approach.



Figure 2.29 Time-varying octave filters.

• Time-varying shelving and peak filters: the special allpass realization of shelving and peak filters has shown that a combination of lowpass, bandpass and allpass filters gives access to several frequency bands inside such a filter structure. Integrating level measurement or envelope followers (see Chapter 5) into these frequency bands can be used for adaptively changing the filter parameters gain, cut-off/center frequency and bandwidth or Q factor. The combination of dynamics processing, which will be discussed in Chapter 5, and parametric filter structures allows the creation of signal dependent filtering effects with a variety of applications.



Figure 2.30 Spectrogram of output signal for time-varying octave filters.

• Feedback cancellers, which are based on time-varying notch filters, play an important role in sound reinforcement systems. The spectrum is continuously monitored for spectral peaks and a very narrow-band notch filter is applied to the signal path.

2.5 Conclusion

Filtering is still one of the most commonly used effect tools for sound recording and production. Nevertheless, its successful application is heavily dependent on the specialized skills of the operator. In this chapter we have described basic filter algorithms for time-domain audio processing. These algorithms perform the filtering operations by the computation of difference equations. The coefficients for the difference equations are given for several filter functions such as lowpass, highpass, bandpass, shelving and peak filters. Simple design formulas for various equalizers lead to efficient implementations for time-varying filter applications. The combination of these basic filters together with the signal processing algorithms of the following chapters allows the building of more sophisticated effects.

Sound and Music

- [m-chu5sna] chu5sna: vocoder speech convolved with snare-drum strokes. Demo Sound. DAFX Sound Library.
- [m-gendsna] gendsna: snare-drum rhythm pattern is mapped onto a gender sound. Demo Sound. DAFX Sound Library.
- [m-Mai97] M. Maiguashca: Reading Castañeda. CD. Wergo2053-2, zkm 3 edition, 1997.
- [m-Pie99] F. Pieper: Das Effekte Praxisbuch. GC Carstensen, 1999. CD, Tr. 1, 35, 36.
- [m-quasthal] quasthal: convolution of speech with impulse responses. Demo Sound. DAFX Sound Library.
- [m-Vag96] H. Vaggione: MYR-S, Composition for cello, electroacoustic set-up and tape. Festival Synthèse, Bourges 1996.
- [m-Vas93] P. Vasseur: PURPLE FRAME, in Le sens caché 100 mystères, CD FREE WAY MUSIQUE, No. 869193, 1993.
- [m-tubg5sna] tubg5sna: tuba glissando convolved by a series of snare-drum strokes. Demo Sound. DAFX Sound Library.

Bibliography

- [Bri94] R. Bristow. The equivalence of various methods of computing biquad coefficients for audio parametric equalizers. In Proc. 97th AES Convention, Preprint 3906, San Francisico, 1994.
- [Cha80] H. Chamberlin. Musical Applications of Microprocessors. Hayden Book Company, 1980.
- [Dat97] J. Dattoro. Effect design, part 1: Reverberator and other filters. J. Audio Eng. Soc., 45(9):660-684, September 1997.
- [Die00] S. Diedrichsen. Personal communication. emagic GmbH, 2000.
- [Dis99] S. Disch. Digital audio effects modulation and delay lines. Master's thesis, Technical University Hamburg-Harburg, 1999.
- [DJ85] C. Dodge and T. A. Jerse. Computer Music; Synthesis, Composition and Performance. Schirmer Books, 1985.

- [DMT99] P. Dutilleux and C. Müller-Tomfelde. AML | Architecture and Music Laboratory, a museum installation. In Proc. of the 16th AES Int. Conf. on Spatial Sound Reproduction. Rovaniemi, Finland. Audio Engineering Society, pp. 191-206, April 10-12 1999.
- [Dut89a] P. Dutilleux. Simple to operate digital time-varying filters. In Proc. 86th AES Convention, Preprint 2757, Hamburg, March 1989.
- [Dut89b] P. Dutilleux. Spinning the sounds in real-time. In Proc. International Computer Music Conference, pp. 94–97, November Columbus 1989.
- [Dut91] P. Dutilleux. Vers la machine à sculpter le son, modification en temps réel des caractéristiques fréquentielles et temporelles des sons. PhD thesis, University of Aix-Marseille II, 1991.
- [Gar95] W.G. Gardner. Efficient convolution without input-output delay. J. Audio Eng. Soc., 43(3):127–136, March 1995.
- [HB93] F. Harris and E. Brooking. A versatile parametric filter using an imbeded all-pass sub-filter to independently adjust bandwidth, center frequency, and boost or cut. In 95th AES Convention, Preprint 3757, New York, 1993.
- [LKG72] M. Labarrère, J. Krief, and B. Gimonet. Le filtrage et ses applications. CEPADUES, 1972.
- [Mai97] M. Maiguashca. Reading Castañeda. Booklet. Wergo2053-2, zkm 3 edition, 1997.
- [McG93] D.S. McGrath. An efficient 30-band graphic equalizer implementation for a low cost dsp processor. In Proc. 95th AES Convention, Preprint 3756, New York, 1993.
- [MT99] C. Müller-Tomfelde. Low-latency convolution for real-time applications. In Proc. of the 16th AES Int. Conf. on Spatial Sound Reproduction, Rovaniemi, Finland, pp. 454–460. Audio Engineering Society, April 10–12 1999.
- [Orf96] S.J. Orfanidis. Introduction to Signal Processing. Prentice-Hall, 1996.
- [Orf97] S.J. Orfanidis. Digital parametric equalizer design with prescribed nyquist-frequency gain. J. Audio Eng. Soc., 45(6):444-455, June 1997.
- [RM87] P.A. Regalia and S.K. Mitra. Tunable digital frequency response equalization filters. *IEEE Trans. Acoustics, Speech and Signal Processing*, 35(1):118–120, January 1987.
- [Roa97] C. Roads. Sound transformation by convolution. In C. Roads, St. Pope, A. Piccialli, and G. De Poli (eds), *Musical Signal Processing*, pp. 411–438. Swets & Zeitlinger Publishers, 1997.

[Smi84]	J.O. Smith. An all-pass approach to digital phasing and flanging. In	n
	Proc. International Computer Music Conference, pp. 103–109, 1984.	

- [Str83] A. Strange. Electronic Music, Systems, Techniques and Controls. W. C. Brown, 1983.
- [Vag98] H. Vaggione. Transformations morphologiques: quelques exemples. In Proceedings of JIM98, CNRS-LMA, pp. G1.1–G1.10, Marseille 1998.
- [Whi86] S.A. White. Design of a digital biquadratic peaking or notch filter for digital audio equalization. J. Audio Eng. Soc., 34(6):479-482, June 1986.
- [Zöl97] U. Zölzer. Digital Audio Signal Processing. John Wiley & Sons, Ltd, 1997.