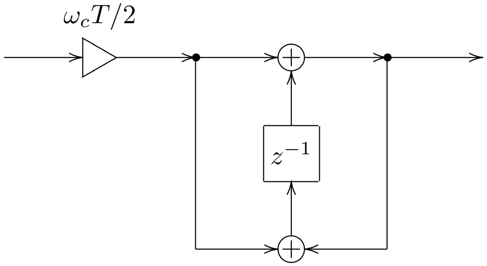
THE ART OF
VA FILTER DESIGN



Vadim Zavalishin

rev. 1.0.3 (June 1, 2012)

About this book: the book covers the theoretical and practical aspects of the virtual analog filter design in the music DSP context. Only a basic amount of DSP knowledge is assumed as a prerequisite. For digital musical instrument and effect developers.

Front picture: BLT integrator.

DISCLAIMER: THIS BOOK IS PROVIDED “AS IS”, SOLELY AS AN EXPRESSION OF THE AUTHOR’S BELIEFS AND OPINIONS AT THE TIME OF THE WRITING, AND IS INTENDED FOR THE INFORMATIONAL PURPOSES ONLY.

To the memory of Elena Golushko,
may her soul travel the happiest path...

Contents

Preface

The classical way of presentation of the DSP theory is not very well suitable for the purposes of virtual analog filter design. The linearity and time-invariance of structures are not assumed merely to simplify certain analysis and design aspects, but are handled more or less as an "ultimate truth". The connection to the continuous-time (analog) world is lost most of the time. The key focus points, particularly the discussed filter types, are of little interest to a digital music instrument developer. This makes it difficult to apply the obtained knowledge in the music DSP context, especially in the virtual analog filter design.

This book attempts to amend this deficiency. The concepts are introduced with the musical VA filter design in mind. The depth of theoretical explanation is restricted to an intuitive and practically applicable amount. The focus of the book is the design of digital models of classical musical analog filter structures using the *topology-preserving transform* approach, which can be considered as a generalization of bilinear transform, zero-delay feedback and trapezoidal integration methods. This results in digital filters having nice amplitude and phase responses, nice time-varying behavior and plenty of options for nonlinearities. In a way, this book can be seen as a detailed explanation of the materials provided in the author's article *"Preserving the LTI system topology in s- to z-plane transforms."*

The prerequisites for the reader include familiarity with the basic DSP concepts, complex algebra and the basic ideas of mathematical analysis. Some basic knowledge of electronics may be helpful at one or two places, but is not critical for the understanding of the presented materials.

The author would like to apologize for possible mistakes and messy explanations. This book has been written in a large haste, just in a few days and didn't go through any serious proofreading yet.

Acknowledgements

The author would like to express his gratitude to a number of people who work (or worked at a certain time) at NI and helped him with the matters related to the creation of this book in one or another way: Daniel Haver, Mate Galic, Tom Kurth, Nicolas Gross, Maike Weber, Martijn Zwartjes, and Mike Daliot. Special thanks to Stephan Schmitt, Egbert Jürgens and Maximilian Zagler.

The author is also grateful to a number of people on the KVR Audio DSP forum and the music DSP mailing list for productive discussions regarding the matters discussed in the book. Particulary to Martin Eisenberg for the detailed and extensive discussion of the delayless feedback, to Dominique Wurtz for the idea of the full equivalence of different BLT integrators and to Teemu Voipio for the introduction of the transposed direct form II BLT integrator in the TPT context. Thanks to Robin Schmidt and Richard Hoffmann for reporting a number of mistakes in the book text.

One shouldn't underestimate the small but invaluable contribution by Helene Kolpakova, whose questions and interest in the VA filter design matters have triggered the idea to quickly write this book.

Last, but most importantly, big thanks to Bob Moog for inventing the voltage-controlled transistor ladder filter.

Chapter 1

Fourier theory

When we are talking about filters we say that filters modify the frequency content of the signal. E.g. a lowpass filter lets the low frequencies through, while suppressing the high frequencies, a highpass filter does vice versa etc. In this chapter we are going to develop a formal definition[1] of the concept of frequencies "contained" in a signal. We will later use this concept to analyse the behavior of the filters.

[1] More precisely we will develop a number of definitions.

1.1 Complex sinusoids

In order to talk about the filter theory we need to introduce complex sinusoidal signals. Consider the complex identity:

$$e^{jt} = \cos t + j \sin t \qquad (t \in \mathbb{R})$$

(notice that, if t is the time, then the point e^{jt} is simply moving along a unit circle in the complex plane). Then

$$\cos t = \frac{e^{jt} + e^{-jt}}{2}$$

and

$$\sin t = \frac{e^{jt} - e^{-jt}}{2j}$$

Then a real sinusoidal signal $a \cos(\omega t + \varphi)$ where a is the real amplitude and φ is the initial phase can be represented as a sum of two complex conjugate sinusoidal signals:

$$\begin{aligned} a \cos(\omega t + \varphi) =& \frac{a}{2}\left(e^{j(\omega t+\varphi)} + e^{-j(\omega t+\varphi)}\right) = \\ &= \left(\frac{a}{2}e^{j\varphi}\right)e^{j\omega t} + \left(\frac{a}{2}e^{-j\varphi}\right)e^{-j\omega t} \end{aligned}$$

Notice that we have a sum of two complex conjugate sinusoids $e^{\pm j\omega t}$ with respective complex conjugate amplitudes $(a/2)e^{\pm j\varphi}$. So, the complex amplitude simultaneously encodes both the amplitude information (in its absolute magnitude) and the phase information (in its argument). For the positive-frequency component $(a/2)e^{j\varphi} \cdot e^{j\omega t}$, the complex "amplitude" $a/2$ is a half of the real amplitude and the complex "phase" φ is equal to the real phase.

1.2 Fourier series

Let $x(t)$ be a real periodic signal of a period T:

$$x(t) = x(t + T)$$

Let $\omega = 2\pi/T$ be the fundamental frequency of that signal. Then $x(t)$ can be represented[2] as a sum of a finite or infinite number of sinusoidal signals of harmonically related frequencies $jn\omega$ plus the *DC offset* term[3] $a_0/2$:

$$x(t) = \frac{a_0}{2} + \sum_{n=1}^{\infty} a_n \cos(jn\omega t + \varphi_n) \tag{1.1}$$

The representation (1.1) is referred to as *real-form Fourier series.* The respective sinusoidal terms are referred to as the *harmonics* or the harmonic *partials* of the signal.

Using the complex sinusoid notation the same can be rewritten as

$$x(t) = \sum_{n=-\infty}^{\infty} X_n e^{jn\omega t} \tag{1.2}$$

where each harmonic term $a_n \cos(jn\omega t + \varphi_n)$ will be represented by a sum of $X_n e^{jn\omega t}$ and $X_{-n} e^{-jn\omega t}$, where X_n and X_{-n} are mutually conjugate: $X_n = X^*_{-n}$. The representation (1.2) is

[2] Formally speaking, there are some restrictions on $x(t)$. It would be sufficient to require that $x(t)$ is bounded and continuous, except for a finite number of discontinuous jumps per period.

[3] The reason the DC offset term is notated as $a_0/2$ and not as a_0 has to do with simplifying the math notation in other related formulas.

referred to as *complex-form Fourier series*. Note that we don't have an explicit DC offset partial in this case, it is implicitly contained in the series as the term for $n = 0$.

It can be easily shown that the real- and complex-form coefficients are related as

$$\begin{aligned} X_n &= \frac{a_n}{2} e^{j\varphi_n} \qquad (n > 0) \\ X_0 &= \frac{a_0}{2} \end{aligned}$$

This means that intuitively we can use the absolute magnitude and the argument of X_n (for positive-frequency terms) as the amplitudes and phases of the real Fourier series partials.

Complex-form Fourier series can also be used to represent complex (rather than real) periodic signals in exactly the same way, except that the equality $X_n = X^*_{-n}$ doesn't hold anymore.

Thus, any real periodic signal can be represented as a sum of harmonically related real sinusoidal partials plus the DC offset. Alternatively, any periodic signal can be represented as a sum of harmonically related complex sinusoidal partials.

1.3 Fourier integral

While periodic signals are representable as a sum of a countable number of sinusoidal partials, a nonperiodic real signal can be represented[4] as a sum of an uncountable number of sinusoidal

[4] As with Fourier series, there are some restrictions on $x(t)$. It is sufficient to require $x(t)$ to be absolutely integrable, bounded and continuous

partials:

$$x(t) = \int_0^\infty a(\omega)\cos\bigl(\omega t + \varphi(\omega)\bigr)\,\frac{\mathrm{d}\omega}{2\pi} \tag{1.3}$$

The representation (1.3) is referred to as *Fourier integral.*[5] The DC offset term doesn't explicitly appear in this case.

The complex-form version of Fourier integral[6] is

$$x(t) = \int_{-\infty}^{\infty} X(\omega)e^{j\omega t}\,\frac{\mathrm{d}\omega}{2\pi} \tag{1.4}$$

For real $x(t)$ we have a Hermitian $X(\omega)$: $X(\omega) = X^*(-\omega)$, for complex $x(t)$ there is no such restriction. The function $X(\omega)$ is referred to as *Fourier transform* of $x(t)$.[7]

(except for a finite number of discontinuous jumps per any finite range of the argument value). The most critical requirement here is probably the absolute integrability, which is particularly fulfilled for the timelimited signals.

[5] The $1/2\pi$ factor is typically used to simplify the notation in the theoretical analysis involving the computation. Intuitively, the integration is done with respect to the ordinary, rather than circular frequency:

$$x(t) = \int_0^\infty a(t)\cos\bigl(2\pi f t + \varphi(f)\bigr)\mathrm{d}f$$

Some texts do not use the $1/2\pi$ factor in this position, in which case it appears in other places instead.

[6] A more common term for (1.4) is *inverse Fourier transform.* However the term *inverse Fourier transform* stresses the fact that $x(t)$ is obtained by computing the inverse of some transform, whereas in this book we are more interested in the fact that $x(t)$ is representable as a combination of sinusoidal signals. The term *Fourier integral* better reflects this aspect. It also suggests a similarity to the Fourier series representation.

[7] The notation $X(\omega)$ for Fourier transform shouldn't be confused with

It can be easily shown that the relationship between the parameters of the real and complex forms of Fourier transform is

$$X(\omega) = \frac{a(\omega)}{2} e^{j\varphi(\omega)} \qquad (\omega > 0)$$

This means that intuitively we can use the absolute magnitude and the argument of $X(\omega)$ (for positive frequencies) as the amplitudes and phases of the real Fourier integral partials.

Thus, any timelimited signal can be represented as a sum of an uncountable number of sinusoidal partials of infinitely small amplitudes.

1.4 Dirac delta function

The *Dirac delta function* $\delta(t)$ is intuitively defined as a very high and a very short symmetric impulse with a unit area (Fig. 1.1):

$$\begin{aligned}
&\delta(t) = \begin{cases} +\infty & \text{if } t = 0 \\ 0 & \text{if } t \neq 0 \end{cases} \\
&\delta(-t) = \delta(t) \\
&\int_{-\infty}^{\infty} \delta(t)\,\mathrm{d}t = 1
\end{aligned}$$

the notation $X(s)$ for Laplace transform. Typically one can be told from the other by the semantics and the notation of the argument. Fourier transform has a real argument, most commonly denoted as ω. Laplace transform has a complex argument, most commonly denoted as s.

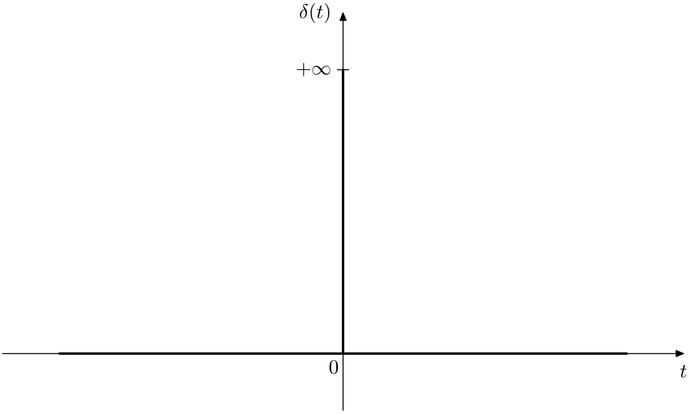


Figure 1.1: Dirac delta function.

Since the impulse is infinitely narrow and since it has a unit area,

$$\int_{-\infty}^{\infty} f(\tau)\delta(\tau)\,\mathrm{d}\tau = f(0) \qquad \forall f$$

from where it follows that a convolution of any function $f(t)$ with $\delta(t)$ doesn't change $f(t)$:

$$(f * \delta)(t) = \int_{-\infty}^{\infty} f(\tau)\delta(t-\tau)\,\mathrm{d}\tau = f(t)$$

Dirac delta can be used to represent Fourier series by a

Fourier integral. If we let

$$X(\omega) = \sum_{n=-\infty}^{\infty} 2\pi\delta(\omega - n\omega_f)X_n$$

then

$$\sum_{n=-\infty}^{\infty} X_n e^{jn\omega_f t} = \int_{-\infty}^{\infty} X(\omega)e^{j\omega t}\,\frac{d\omega}{2\pi}$$

From now on, we'll not separately mention Fourier series, assuming that Fourier integral can represent any necessary signal.

Thus, most signals can be represented as a sum of (a possibly infinite number of) sinusoidal partials.

1.5 Laplace transform

Let $s = j\omega$. Then, a complex-form Fourier integral can be rewritten as

$$x(t) = \int_{-j\infty}^{+j\infty} X(s)e^{st}\,\frac{ds}{2\pi j}$$

where the integration is done in the complex plane along the straight line from $-j\infty$ to $+j\infty$ (apparently $X(s)$ is a different function than $X(\omega)$[8]). For timelimited signals the function $X(s)$

[8] As already mentioned, the notation $X(\omega)$ for Fourier transform shouldn't be confused with the notation $X(s)$ for Laplace transform. Typically one can be told from the other by the semantics and the notation of the argument. Fourier transform has a real argument, most commonly denoted as ω. Laplace transform has a complex argument, most commonly denoted as s.

can be defined on the entire complex plane in such a way that the integration can be done along any line which is parallel to the imaginary axis:

$$x(t) = \int_{\sigma-j\infty}^{\sigma+j\infty} X(s)e^{st}\,\frac{\mathrm{d}s}{2\pi j} \qquad (\sigma \in \mathbb{R}) \tag{1.5}$$

In many other cases such $X(s)$ can be defined within some strip $\sigma_1 < \operatorname{Re} s < \sigma_2$. Such function $X(s)$ is referred to as bilateral *Laplace transform* of $x(t)$, whereas the representation (1.5) can be referred to as *Laplace integral.*[9][10]

Notice that the *complex exponential* e^{st} is representable as

$$e^{st} = e^{\operatorname{Re} s\cdot t}e^{\operatorname{Im} s\cdot t}$$

Considering $e^{\operatorname{Re} s\cdot t}$ as the amplitude of the complex sinusoid $e^{\operatorname{Im} s\cdot t}$ we notice that e^{st} is:

- an exponentially decaying complex sinusoid if $\operatorname{Re} s < 0$,
- an exponentially growing complex sinusoid if $\operatorname{Re} s > 0$,
- a complex sinusoid of constant amplitude if $\operatorname{Re} s = 0$.

[9] A more common term for (1.5) is *inverse Laplace transform.* However the term *inverse Laplace transform* stresses the fact that $x(t)$ is obtained by computing the inverse of some transform, whereas is this book we are more interested in the fact that $x(t)$ is representable as a combination of exponential signals. The term *Laplace integral* better reflects this aspect.

[10] The representation of periodic signals by Laplace integral (using Dirac delta function) is problematic for $\sigma \neq 0$. Nevertheless, we can represent them by a Laplace integral if we restrict σ to $\sigma = 0$ (that is $\operatorname{Re} s = 0$ for $X(s)$).

Thus, most signals can be represented as a sum of (a possibly infinite number of) complex exponential partials, where the amplitude growth or decay speed of these partials can be relatively arbitrarily chosen.

SUMMARY

The most important conclusion of this chapter is: any signal occurring in practice can be represented as a sum of sinusoidal (real or complex) components. The frequencies of these sinusoids can be referred to as the “frequencies contained in the signal”. For complex representation, the real amplitude and phase information is encoded in the absolute magnitude and the argument of the complex amplitudes of the positive-frequency partials (where the absolute magnitude of the complex amplitude is a half of the real amplitude).

It is also possible to use complex exponentials instead of sinusoids.

Chapter 2

Analog 1-pole filters

In this chapter we are going to introduce the basic analog RC-filter and use it as an example to develop the key concepts of the analog filter analysis.

2.1 RC filter

Consider the circuit in Fig. 2.1, where the voltage $x(t)$ is the input signal and the capacitor voltage $y(t)$ is the output signal. This circuit represents the simplest 1-pole *lowpass filter*, which we are now going to analyse.

Writing the equations for that circuit we have:

$$x = U_R + U_C$$

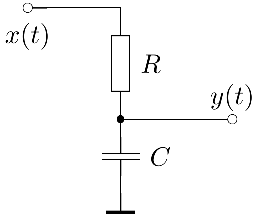


Figure 2.1: A simple RC lowpass filter.

$$\begin{aligned} y &= U_C \\ U_R &= RI \\ I &= \dot{q}_C \\ q_C &= CU_C \end{aligned}$$

where U_R is the resistor voltage, U_C is the capacitor voltage, I is the current through the circuit and q_C is the capacitor charge. Reducing the number of variables, we can simplify the equation system to:

$$x = RC\dot{y} + y$$

or

$$\dot{y} = \frac{1}{RC}(x - y)$$

or, integrating with respect to time:

$$y = y(t_0) + \int_{t_0}^{t} \frac{1}{RC}\bigl(x(\tau) - y(\tau)\bigr)\,\mathrm{d}\tau$$

where t_0 is the *initial time moment*. Introducing the notation $\omega_c = 1/RC$ we have

$$y = y(t_0) + \int_{t_0}^{t} \omega_c\bigl(x(\tau) - y(\tau)\bigr)\,\mathrm{d}\tau \tag{2.1}$$

We will reintroduce ω_c later as the *cutoff* of the filter.

Notice that we didn't factor $1/RC$ (or ω_c) out of the integral for the case when the value of R is varying with time. The varying R corresponds to the varying cutoff of the filter, and this situation is highly typical in the music DSP context.[1]

2.2 Block diagrams

The integral equation (2.1) can be expressed in the block diagram form (Fig. 2.2).

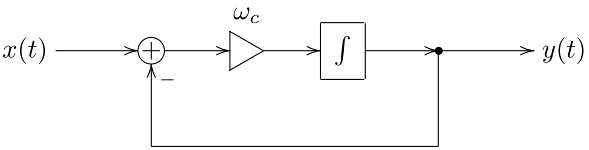


Figure 2.2: A 1-pole RC lowpass filter in the block diagram form.

[1] We didn't assume the varying C because then our simplification of the equation system doesn't hold anymore, since $\dot{q}_C \neq C\dot{U}_C$ in this case.

The meaning of the elements of the diagram should be intuitively clear. The *gain element* (represented by a triangle) multiplies the input signal by ω_c. Notice the inverting input of the summator, denoted by "$-$". The integrator simply integrates the input signal:

$$\mathit{output}(t) = \mathit{output}(t_0) + \int_{t_0}^{t} \mathit{input}(\tau)\,\mathrm{d}\tau$$

The representation of the system by the integral (rather than differential) equation and the respective usage of the integrator element in the block diagram has an important intuitive meaning. Intuitively, the capacitor integrates the current flowing through it, accumulating it as its own charge:

$$q_C(t) = q_C(t_0) + \int_{t_0}^{t} I(\tau)\,\mathrm{d}\tau$$

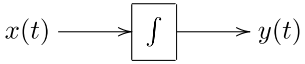
or, equivalently

$$U_C(t) = U_C(t_0) + \frac{1}{C}\int_{t_0}^{t} I(\tau)\,\mathrm{d}\tau$$

One can observe from Fig. 2.2 that the output signal is always trying to "reach" the input signal. Indeed, the difference $x - y$ is always "directed" from y to x. Since $\omega_c > 0$, the integrator will respectively increase or decrease its output value in the respective direction. This corresponds to the fact that the capacitor voltage in Fig. 2.1 is always trying to reach the input voltage. Thus, the circuit works as a kind of smoother of the input signal.

2.3 Transfer function

Consider the integrator:



Suppose $x(t) = e^{st}$ (where $s = j\omega$ or, possibly, another complex value). Then

$$y(t) = y(t_0) + \int_{t_0}^{t} e^{s\tau}\,\mathrm{d}\tau = y(t_0) + \frac{1}{s}e^{s\tau}\bigg|_{\tau=t_0}^{t} =$$
$$= \frac{1}{s}e^{st} + \left(y(t_0) - \frac{1}{s}e^{st_0}\right)$$

Thus, a complex sinusoid (or exponential) e^{st} sent through an integrator comes out as the same signal e^{st} just with a different amplitude $1/s$ plus some DC term $y(t_0) - e^{st_0}/s$. Similarly, a signal $X(s)e^{st}$ (where $X(s)$ is the complex amplitude of the signal) comes out as $(X(s)/s)e^{st}$ plus some DC term. That is, if we forget about the extra DC term, *the integrator simply multiplies the amplitudes of complex exponential signals* e^{st} *by* $1/s$.

Now, the good news is: for our purposes of filter analysis we can simply *forget* about the extra DC term. The reason for this is the following. Suppose the initial time moment t_0 was quite long ago ($t_0 \ll 0$). Suppose further that the integrator is contained in a *stable* filter (we will discuss the filter stability later, for now we'll simply mention that we're mostly interested in the stable filters for the purposes of the current discussion).

It can be shown that in this case the effect of the extra DC term on the output signal is negligible. Since the initial state $y(t_0)$ is incorporated into the same DC term, it also means that the effect of the initial state is negligible![2]

Thus, we simply write (for an integrator):

$$\int e^{s\tau}\,\mathrm{d}\tau = \frac{1}{s}e^{st}$$

This means that e^{st} is an *eigenfunction* of the integrator with the respective eigenvalue $1/s$.

Since the integrator is linear,[3] not only are we able to factor $X(s)$ out of the integration:

$$\int X(s)e^{s\tau}\,\mathrm{d}\tau = X(s)\int e^{s\tau}\,\mathrm{d}\tau = \frac{1}{s}X(s)e^{st}$$

but we can also apply the integration independently to all Fourier (or Laplace) partials of an arbitrary signal $x(t)$:

$$\int\left(\int_{\sigma-j\infty}^{\sigma+j\infty} X(s)e^{s\tau}\,\frac{\mathrm{d}s}{2\pi j}\right)\mathrm{d}\tau =$$

[2] In practice, typically, a zero initial state is assumed. Then, particularly, in the case of absence of the input signal, the output signal of the filter is zero from the very beginning (rather than for $t \gg t_0$).

[3] The linearity here is understood in the sense of the operator linearity. An operator $\hat{H}$ is linear, if

$$\hat{H}\left(\lambda_1 f_1(t) + \lambda_2 f_2(t)\right) = \lambda_1 \hat{H} f_1(t) + \lambda_2 \hat{H} f_2(t)$$

$$= \int_{\sigma-j\infty}^{\sigma+j\infty} \left(\int X(s) e^{s\tau} \, \mathrm{d}\tau \right) \frac{\mathrm{d}s}{2\pi j} =$$
$$= \int_{\sigma-j\infty}^{\sigma+j\infty} \frac{X(s)}{s} e^{s\tau} \, \frac{\mathrm{d}s}{2\pi j}$$

That is, the integrator changes the complex amplitude of each partial by a $1/s$ factor.

Consider again the structure in Fig. 2.2. Assuming the input signal $x(t)$ has the form e^{st} we can replace the integrator by a gain element with a $1/s$ factor. We symbolically reflect this by replacing the integrator symbol in the diagram with the $1/s$ fraction (Fig. 2.3).[4]

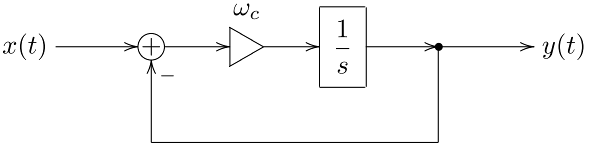


Figure 2.3: A 1-pole RC lowpass filter in the block diagram form with a $1/s$ notation for the integrator.

[4] Often in such cases the input and output signal notation for the block diagram is replaced with $X(s)$ and $Y(s)$. Such diagram then "works" in terms of Laplace transform, the input of the diagram is the Laplace transform $X(s)$ of the input signal $x(t)$, the output is respectively the Laplace transform $Y(s)$ of the output signal $y(t)$. The integrators can then be seen as s-dependent gain elements, where the gain coefficient is $1/s$.

So, suppose $x(t) = X(s)e^{st}$ and suppose we know $y(t)$. Then the input signal for the integrator is $\omega_c(x - y)$. We now will further take for granted the knowledge that $y(t)$ will be the same signal e^{st} with some different complex amplitude $Y(s)$, that is $y(t) = Y(s)e^{st}$ (notably, this holds only if ω_c is constant, that is, if the system is *time-invariant*!!!)[5] Then the input signal of the integrator is $\omega_c(X(s) - Y(s))e^{st}$ and the integrator simply multiplies its amplitude by $1/s$. Thus the output signal of the integrator is $\omega_c(x - y)/s$. But, on the other hand $y(t)$ is the output signal of the integrator, thus

$$y(t) = \omega_c \frac{x(t) - y(t)}{s}$$

or

$$Y(s)e^{st} = \omega_c \frac{X(s) - Y(s)}{s} e^{st}$$

or

$$Y(s) = \omega_c \frac{X(s) - Y(s)}{s}$$

from where

$$sY(s) = \omega_c X(s) - \omega_c Y(s)$$

and

$$Y(s) = \frac{\omega_c}{s + \omega_c} X(s)$$

Thus, the circuit in Fig. 2.3 (or in Fig. 2.2) simply scales the amplitude of the input sinusoidal (or exponential) signal $X(s)e^{st}$ by the $\omega_c/(s + \omega_c)$ factor.

[5] In other words, we take for granted the fact that e^{st} is an eigenfunction of the entire circuit.

Let's introduce the notation

$$H(s) = \frac{\omega_c}{s+\omega_c} \tag{2.2}$$

Then

$$Y(s) = H(s)X(s)$$

$H(s)$ is referred to as the *transfer function* of the structure in Fig. 2.3 (or Fig. 2.2). Notice that $H(s)$ is a complex function of a complex argument.

For an arbitrary input signal $x(t)$ we can use the Laplace transform representation

$$x(t) = \int_{\sigma-j\infty}^{\sigma+j\infty} X(s)e^{st}\,\frac{\mathrm{d}s}{2\pi j}$$

From the *linearity*[6] of the circuit in Fig. 2.3, it follows that the result of the application of the circuit to a linear combination of some signals is equal to the linear combination of the results of the application of the circuit to the individual signals. That is, for each input signal of the form $X(s)e^{st}$ we obtain the output signal $H(s)X(s)e^{st}$. Then for an input signal which is an integral sum of $X(s)e^{st}$, we obtain the output signal which is an integral sum of $H(s)X(s)e^{st}$. That is

$$y(t) = \int_{\sigma-j\infty}^{\sigma+j\infty} H(s)X(s)e^{st}\,\frac{\mathrm{d}s}{2\pi j}$$

[6] Here we again understand the linearity in the operator sense:

$$\hat{H}\left(\lambda_1 f_1(t) + \lambda_2 f_2(t)\right) = \lambda_1\hat{H}f_1(t) + \lambda_2\hat{H}f_2(t)$$

The operator here corresponds to the circuit in question: $y(t) = \hat{H}x(t)$ where $x(t)$ and $y(t)$ are the input and output signals of the circuit.

So, the circuit in Fig. 2.3 independently modifies the complex amplitudes of the sinusoidal (or exponential) partials e^{st} by the $H(s)$ factor!

Notably, the transfer function can be introduced for any system which is linear and time-invariant. For the systems, whose block diagrams consist of integrators, summators and fixed gains, the transfer function is always a *non-strictly proper*[7] rational function of s. Particularly, this holds for the electronic circuits, where the differential elements are capacitors and inductors, since these types of elements logically perform integration (capacitors integrate the current to obtain the voltage, while inductors integrate the voltage to obtain the current).

It is important to realize that in the derivation of the transfer function concept we used the linearity and time-invariance (the absence of parameter modulation) of the structure. If these properties do not hold, the transfer function can't be introduced! This means that all transfer function-based analysis holds only in the case of fixed parameter values. In practice, if the parameters are not changing too quickly, one can assume that they are approximately constant during certain time range. That is we can "approximately" apply the transfer function concept (and the discussed later derived concepts, such as amplitude and phase responses, poles and zeros, stability criterion etc.) if the modulation of the parameter values is "not too fast".

[7] A rational function is nonstrictly proper, if the order of its numerator doesn't exceed the order of its denominator.

2.4 Complex impedances

Actually, we could have obtained the transfer function of the circuit in Fig. 2.1 using the concept of *complex impedances*.

Consider the capacitor equation:

$$I = C\dot{U}$$

If

$$\begin{aligned} I(t) &= I(s)e^{st} \\ U(t) &= U(s)e^{st} \end{aligned}$$

(where $I(t)$ and $I(s)$ are obviously two different functions, the same for $U(t)$ and $U(s)$), then

$$\dot{U} = sU(s)e^{st} = sU(t)$$

and thus

$$I(t) = I(s)e^{st} = C\dot{U} = CsU(s)e^{st} = sCU(t)$$

that is

$$I = sCU$$

or

$$U = \frac{1}{sC}I$$

Now the latter equation looks almost like Ohm's law for a resistor: $U = RI$. The complex value $1/sC$ is called the *complex impedance* of the capacitor. The same equation can be written in the Laplace transform form: $U(s) = (1/sC)I(s)$.

For an inductor we have $U = L\dot{I}$ and respectively, for $I(t) = I(s)e^{st}$ and $U(t) = U(s)e^{st}$ we obtain $U(t) = sLI(t)$ or $U(s) = sLI(s)$. Thus, the complex impedance of the inductor is sL.

Using the complex impedances as if they were resistances (which we can do, assuming the input signal has the form $X(s)e^{st}$), we simply write the voltage division formula for the circuit in in Fig. 2.1:

$$y(t) = \frac{U_C}{U_R + U_C}x(t)$$

or, cancelling the common current factor $I(t)$ from the numerator and the denominator, we obtain the impedances instead of voltages:

$$y(t) = \frac{1/sC}{R + 1/sC}x(t)$$

from where

$$H(s) = \frac{y(t)}{x(t)} = \frac{1/sC}{R + 1/sC} = \frac{1}{1 + sRC} = \frac{1/RC}{s + 1/RC} = \frac{\omega_c}{s + \omega_c}$$

which coincides with (2.2).

2.5 Amplitude and phase responses

Consider again the structure in Fig. 2.3. Let $x(t)$ be a real signal and let

$$x(t) = \int_{\sigma - j\infty}^{\sigma + j\infty} X(s)e^{st}\,\frac{\mathrm{d}s}{2\pi j}$$

be its Laplace integral representation. Let $y(t)$ be the output signal (which is obviously also real) and let

$$y(t) = \int_{\sigma-j\infty}^{\sigma+j\infty} Y(s)e^{st}\,\frac{\mathrm{d}s}{2\pi j}$$

be its Laplace integral representation. As we have shown, $Y(s) = H(s)X(s)$ where $H(s)$ is the transfer function of the circuit.

The respective Fourier integral representation of $x(t)$ is apparently

$$x(t) = \int_{-\infty}^{+\infty} X(j\omega)e^{j\omega t}\,\frac{\mathrm{d}\omega}{2\pi}$$

where $X(j\omega)$ is the Laplace transform $X(s)$ evaluated at $s = j\omega$. The real Fourier integral representation is then obtained as

$$\begin{aligned} a_x(\omega) &= 2\cdot|X(j\omega)| \\ \varphi_x(\omega) &= \arg X(j\omega) \end{aligned}$$

For $y(t)$ we respectively have[8][9]

$$\begin{aligned}
a_y(\omega) &= 2\cdot|Y(j\omega)| = 2\cdot|H(j\omega)X(j\omega)| = |H(j\omega)|\cdot a_x(\omega)\\
\varphi_y(\omega) &= \arg Y(j\omega) = \arg\left(H(j\omega)X(j\omega)\right) = \varphi_x(\omega)+\arg H(j\omega)\\
&\qquad(\omega\geq 0)
\end{aligned}$$

Thus, the amplitudes of the real sinusoidal partials are magnified by the $|H(j\omega)|$ factor and their phases are shifted by $\arg H(j\omega)$ $(\omega \geq 0)$. The function $|H(j\omega)|$ is referred to as the *amplitude response* of the circuit and the function $\arg H(j\omega)$ is referred to as the *phase response* of the circuit. Note that both the amplitude and the phase response are real functions of a real argument ω.

The complex-valued function $H(j\omega)$ of the real argument ω is referred to as the *frequency response* of the circuit. Simply put, the frequency response is equal to the transfer function evaluated on the imaginary axis.

Since the transfer function concept works only in the linear time-invariant case, so do the concepts of the amplitude, phase and frequency responses!

[8] This relationship holds only if $H(j\omega)$ is Hermitian: $H(j\omega)=H^*(-j\omega)$. If it weren't the case, the Hermitian property wouldn't hold for $Y(j\omega)$ and $y(t)$ couldn't have been a real signal (for a real input $x(t)$). Fortunately, for real systems $H(j\omega)$ is always Hermitian. Particularly, rational transfer functions $H(s)$ with real coefficients obviously result in Hermitian $H(j\omega)$.

[9] Formally, $\omega = 0$ requires special treatment in case of a Dirac delta component at $\omega=0$ (arising particularly if the Fourier series is represented by a Fourier integral and there is a nonzero DC offset). Nevertheless, the resulting relationship between $a_y(0)$ and $a_x(0)$ is exactly the same as for $\omega>0$, that is $a_y(0)=H(0)a_x(0)$. A more complicated but same argument holds for the phase.

2.6 Lowpass filtering

Consider again the transfer function of the structure in Fig. 2.2:

$$H(s) = \frac{\omega_c}{s + \omega_c}$$

The respective amplitude response is

$$|H(j\omega)| = \left|\frac{\omega_c}{\omega_c + j\omega}\right|$$

Apparently at $\omega = 0$ we have $H(0) = 1$. On the other hand, as ω grows, the magnitude of the denominator grows as well and the function decays to zero: $H(+j\infty) = 0$. This suggests the lowpass filtering behavior of the circuit: it lets the partials with frequencies $\omega \ll \omega_c$ through and stops the partials with frequencies $\omega \gg \omega_c$. The circuit is therefore referred to as a *lowpass filter*, while the value ω_c is defined as the *cutoff* frequency of the circuit.

It is convenient to plot the amplitude response of the filter in a fully logarithmic scale. The amplitude gain will then be plotted in decibels, while the frequency axis will have a uniform spacing of octaves. For $H(s) = \omega_c/(s + \omega_c)$ the plot looks like the one in Fig. 2.4.

Notice that the plot falls off in an almost straight line as $\omega \to \infty$. Apparently, at $\omega \gg \omega_c$ and respectively $|s| \gg \omega_c$ we have $H(s) \approx \omega_c/s$ and $|H(s)| \approx \omega_c/\omega$. This is a hyperbola in the linear scale and a straight line in a fully logarithmic scale. If ω doubles (corresponding to a step up by one octave), the

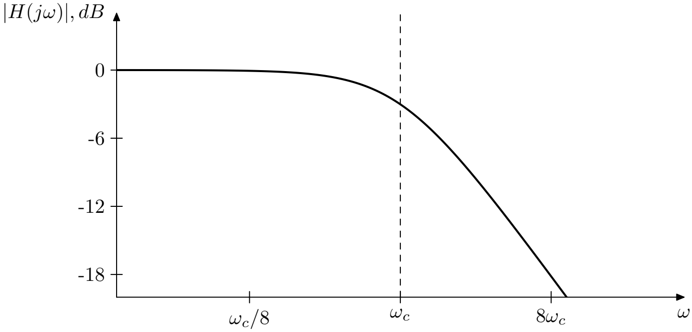


Figure 2.4: Amplitude response of a 1-pole lowpass filter.

amplitude gain is approximately halved (that is, drops by approximately 6 decibel). We say that this lowpass filter has a *rolloff* of 6dB/oct.

Another property of this filter is that the amplitude drop at the cutoff is −3dB. Indeed

$$|H(j\omega_c)| = \left|\frac{\omega_c}{\omega_c + j\omega_c}\right| = \left|\frac{1}{1+j}\right| = \frac{1}{\sqrt{2}} \approx -3\text{dB}$$

2.7 Cutoff parametrization

Suppose $\omega_c = 1$. Then the lowpass transfer function (2.2) turns into

$$H(s) = \frac{1}{s+1}$$

Now perform the substitution $s \leftarrow s/\omega_c$. We obtain

$$H(s) = \frac{1}{s/\omega_c + 1} = \frac{\omega_c}{s + \omega_c}$$

which is again our familiar transfer function of the lowpass filter.

Consider the amplitude response graph of $1/(s+1)$ in a logarithmic scale. The substitution $s \leftarrow s/\omega_c$ simply shifts this graph to the left or to the right (depending on whether $\omega_c < 1$ or $\omega_c > 1$) without changing its shape. Thus, the variation of the cutoff parameter doesn't change the shape of the amplitude response graph (Fig. 2.5), or of the phase response graph, for that matter (Fig. 2.6).

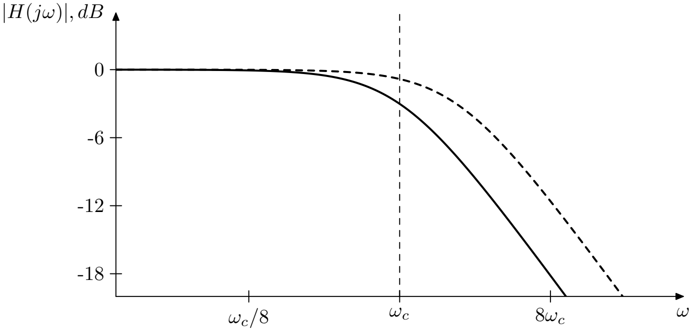


Figure 2.5: 1-pole lowpass filter's amplitude response shift by a cutoff change.

The substitution $s \leftarrow s/\omega_c$ is a generic way to handle cutoff parametrization for analog filters, because it doesn't change the

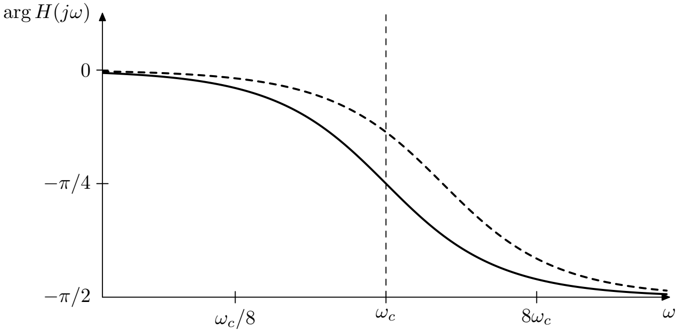
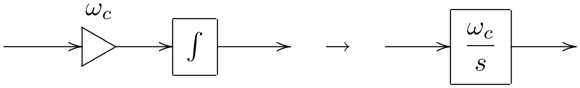


Figure 2.6: 1-pole lowpass filter's phase response shift by a cutoff change.

response shapes. This has a nice counterpart on the block diagram level. For all types of filters we simply visually combine an ω_c gain and an integrator into a single block:[10]

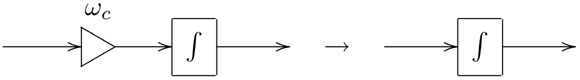


[10] Notice that including the cutoff gain into the integrator makes the integrator block invariant to the choice of the time units:

$$y(t) = y(t_0) + \int_{t_0}^{t} \omega_c x(\tau)\,\mathrm{d}\tau$$

because the product $\omega_c\,\mathrm{d}\tau$ is invariant to the choice of the time units. This will become important once we start building discrete-time models of filters, where we would often assume unit sampling period.

Apparently, the reason for the ω_c/s notation is that this is the transfer function of the serial connection of an ω_c gain and an integrator. Alternatively, we simply assume that the cutoff gain is contained inside the integrator:



The internal representation of such integrator block is of course still a cutoff gain followed by an integrator. Whether the gain should precede the integrator or follow it may depend on the details of the analog prototype circuit. In the absence of the analog prototype it's better to put the integrator *after* the cutoff gain, because then the integrator will smooth the jumps and further artifacts arising out of the cutoff modulation.

With the cutoff gain implied inside the integrator block, the structure from Fig. 2.2 is further simplified to the one in Fig. 2.7:

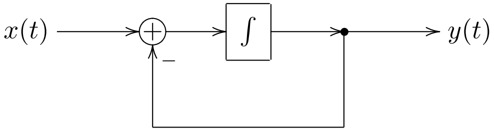


Figure 2.7: A 1-pole RC lowpass filter with an implied cutoff.

As a further shortcut arising out of the just discussed facts, it is common to assume $\omega_c = 1$ during the filter analysis. Particularly, the transfer function of a 1-pole lowpass filter is often written as

$$H(s) = \frac{1}{s+1}$$

It is assumed that the reader will perform the $s \leftarrow s/\omega_c$ substitution as necessary.

2.8 Highpass filter

If instead of the capacitor voltage in Fig. 2.1 we pick up the resistor voltage as the output signal, we obtain the block diagram representation as in Fig. 2.8.

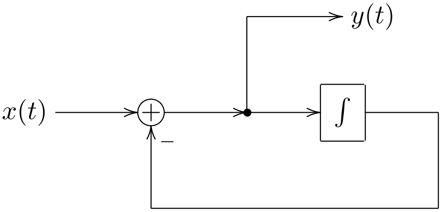


Figure 2.8: A 1-pole highpass filter.

Obtaining the transfer function of this filter we get

$$H(s) = \frac{s}{s+\omega_c}$$

(or $s/(s+1)$ in the unit cutoff form). It's easy to see that $H(0) = 0$ and $H(+j\infty) = 1$, whereas the biggest change in the amplitude response occurs again around $\omega = \omega_c$. Thus, we have a *highpass filter* here. The amplitude response of this filter is shown in Fig. 2.9 (in the logarithmic scale).

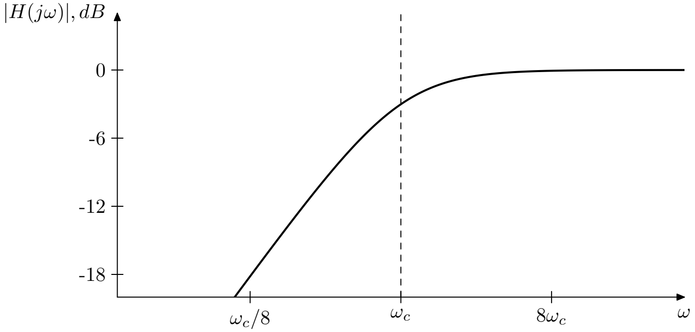


Figure 2.9: Amplitude response of a 1-pole highpass filter.

It's not difficult to observe and not difficult to show that this response is a mirrored version of the one in Fig. 2.4.[11] Particularly, at $\omega \ll \omega_c$ we have $H(s) \approx s/\omega_c$, so when the frequency is halved (dropped by an octave), the amplitude gain is approximately halved as well (drops by approximately 6dB). Again, we

[11] In the unit cutoff notation, it's easy to notice that the highpass transfer function $\frac{s}{1+s}$ can be obtained from the lowpass transfer function $\frac{1}{1+s}$ by the substitution $s \leftarrow 1/s$. This substitution is referred to as *LP to HP* (lowpass to highpass) substitution. For Hermitian transfer functions (corresponding to real systems), the *LP to HP* substitution simply mirrors the amplitude response in the logarithmic frequency scale.

have a 6dB/oct rolloff.

2.9 Poles, zeros and stability

Consider the lowpass transfer function:

$$H(s) = \frac{\omega_c}{s + \omega_c}$$

Apparently, this function has a pole in the complex plane at $s = -\omega_c$. Similarly, the highpass transfer function

$$H(s) = \frac{s}{s + \omega_c}$$

also has a pole at $s = -\omega_c$, but it also has a zero at $s = 0$.

Recall that the transfer functions of linear time-invariant differential systems are nonstrictly proper rational functions of s. Thus they always have poles and often have zeros, the numbers of poles and zeros matching the orders of the numerator and the denominator respectively. The poles and zeros of transfer function (especially the poles) play an important role in the filter analysis. For simplicity they are referred to as the poles and zeros of the filters.

The transfer functions of real linear time-invariant differential systems have real coefficients in the numerator and denominator polynomials. Apparently, this doesn't prevent them from having complex poles and zeros, however, being roots of real polynomials, those must come in complex conjugate pairs. E.g. a transfer function with a 3rd order denominator can have either three real poles, or one real and two complex conjugate poles.

The lowpass and highpass filters discussed so far, each have one pole. For that reason they are referred to as 1-pole filters. Actually, the number of poles is always equal to the order of the filter or (which is the same) to the number of integrators in the filter.[12] Therefore it is common, instead of e.g. a "4th-order filter" to say a "4-pole filter".

The most important property of the poles is that a filter[13] is *stable* if and only if all its poles are located in the left complex semiplane (that is to the left of the imaginary axis). For our lowpass and highpass filters this is apparently true, as long as $\omega_c > 0$.[14] If $\omega_c < 0$, the pole is moved to the right semiplane, the filter becomes unstable and will "explode". Also the definition of the frequency response doesn't make much sense in this case. If we put a sinusoidal signal through a stable filter we will (as we have shown) obtain an amplitude-modified and phase-shifted sinusoidal signal of the same frequency.[15] If we put a sinusiodal

[12] In certain singular cases, depending on the particular definition details, these numbers might be not equal to each other.

[13] More precisely a linear time-invariant system, which particularly implies fixed parameters. This remark is actually unnecessary, since, as we mentioned, the transfer function (and respectively the poles) are defined only for the linear time-invariant case.

[14] Notably, the same condition ensures the stability of the 1-pole RC lowpass and highpass filters in the time-varying case, which can be directly seen from the fact that the lowpass filter's output never exceeds the maximum level of its input.

[15] Strictly speaking, this will happen only after the filter has stabilized itself "to the new signal". This takes a certain amount of time. The closer the poles are to the imaginary axis (from the left), the larger is this stabilization time. The characteristic time value of the stabilization has the order of magnitude of $-1/\max\{\operatorname{Re} p_n\}$, where p_n are the poles. Actually

signal through an unstable filter, the filter simply "explodes" (its output grows infinitely), thus it makes no sense to talk of amplitude and phase responses.

It is also possible to obtain an intuitive understanding of the effect of the pole position on the filter stability. Consider a transfer function of the form

$$H(s) = \frac{F(s)}{\prod\limits_{n=1}^{N}(s-p_n)}$$

where $F(s)$ is the numerator of the transfer function and p_n are the poles. Suppose all poles are initially in the left complex semiplane and now one of the poles (let's say p_1) starts moving towards the imaginary axis. As the pole gets closer to the axis, the amplitude response at $\omega = \operatorname{Im} p_1$ grows. When p_1 gets onto the axis, the amplitude response at $\omega = \operatorname{Im} p_1$ is infinitely large (since $j\omega = p_1$, we have $H(j\omega) = H(p_1) = \infty$). This corresponds to the filter getting unstable.[16][17]

The poles and zeros also define the rolloff speed of the amplitude response. Let N_p be the number of poles and N_z be the

the effects of the transition (occurring at the moment of the appearance of the sinusoidal signal) decay exponentially as $e^{t\max\{\operatorname{Re} p_n\}}$.

[16] The reason, why the stable area is the left (and not the right) complex semiplane, falls outside the scope of this book.

[17] The discussed 1-pole lowpass filter is actually still kind of stable at $\omega = 0$ (corresponding to the pole at $s = 0$. In fact, it has a constant output level (its state is not changing) in this case. However, strictly speaking, this case is not really stable, since all signals in a truely stable filter must decay to zero in the absence of the input signal.

number of zeros. Since the transfer function must be nonstrictly proper, $N_p \geq N_z$. It's not difficult to see that the amplitude response rolloff at $\omega \to +\infty$ is $6(N_p - N_z)$dB/oct. Respectively, the rolloff at $\omega \to 0$ is $6N_{z0}$dB/oct, where N_{z0} is the number of zeros at $s = 0$ (provided there are no poles at $s = 0$). Considering that $0 \leq N_{z0} \leq N_z \leq N_p$, the rolloff speed at $\omega \to +\infty$ or at $\omega \to 0$ can't exceed $6N_p$dB/oct. Also, if all zeros of a filter are at $s = 0$ (that is $N_{z0} = N_z$) then the sum of the rolloff speeds at $\omega \to 0$ and $\omega \to +\infty$ is exactly $6N_p$dB/oct.

2.10 Multimode filter

Actually, we can pick up the lowpass and highpass signals simultaneously from the same structure (Fig. 2.10). This is referred to as a *multimode filter*.

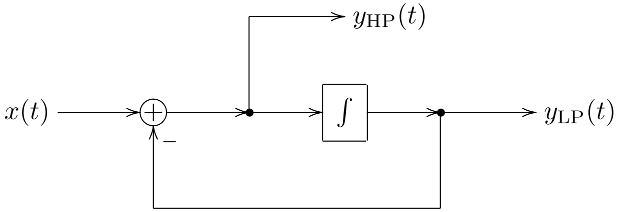


Figure 2.10: A 1-pole multimode filter.

It's easy to observe that $y_{\mathrm{LP}}(t) + y_{\mathrm{HP}}(t) = x(t)$, that is the input signal is split by the filter into the lowpass and highpass components. In the transfer function form this corresponds to

$$H_{\text{LP}}(s) + H_{\text{HP}}(s) = \frac{\omega_c}{s+\omega_c} + \frac{s}{s+\omega_c} = 1$$

The multimode filter can be used to implement a 1st-order differential filter for practically any given transfer function, by simply mixing its outputs. Indeed, let

$$H(s) = \frac{b_1 s + b_0}{a_1 s + a_0} \qquad (a_1 \neq 0, a_0 \neq 0)$$

(the case $a_1 = 0$ turns the transfer function into an improper rational function, while $a_0 = 0$ is not defining a stable filter). Dividing the numerator and the denominator by a_1 we obtain

$$\begin{aligned} H(s) = \frac{(b_1/a_1)s + (b_0/a_1)}{s + a_0/a_1} &= \left(\frac{b_1}{a_1}\right)\frac{s}{s+\omega_c} + \left(\frac{b_0}{a_0}\right)\frac{\omega_c}{s+\omega_c} = \\ &= \left(\frac{b_1}{a_1}\right)H_{\text{HP}}(s) + \left(\frac{b_0}{a_0}\right)H_{\text{LP}}(s) \qquad (\text{where } \omega_c = a_0/a_1) \end{aligned}$$

Thus we simply need to set the filter's cutoff to a_0/a_1 and take the sum

$$y = \left(\frac{b_1}{a_1}\right)y_{\text{HP}}(t) + \left(\frac{b_0}{a_0}\right)y_{\text{LP}}(t)$$

as the output signal.

2.11 Shelving filters

By adding/subtracting the lowpass-filtered signal to/from the unmodified input signal one can build a low-shelving filter:

$$y(t) = x(t) + K \cdot y_{\text{LP}}(t)$$

The transfer function of the low-shelving filter is respectively:

$$H(s) = 1 + K\frac{1}{s+1}$$

The amplitude response is plotted Fig. 2.11. Typically $K \geq -1$. At $K = 0$ the signal is unchanged. At $K = -1$ the filter turns into a highpass.

The high-shelving filter is built in a similar way:

$$y(t) = x(t) + K \cdot y_{\text{HP}}(t)$$

and

$$H(s) = 1 + K\frac{s}{s+1}$$

The amplitude response is plotted Fig. 2.12.

There are a couple of nontrivial moments here, though. The first one has to do with the fact that the amplitude boost or drop for the “shelf” is more convenient to be specified in decibels. Which requires translation of the level change specified in decibels into the K factor. It’s not difficult to realize that

$$\text{dB} = 20 \log_{10}(K + 1)$$

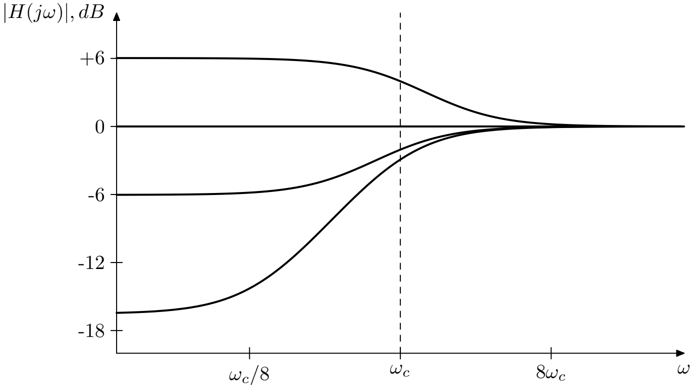


Figure 2.11: Amplitude response of a 1-pole low-shelving filter (for various K).

Indeed, e.g. for the low-shelving filter at $\omega = 0$ (that is $s = 0$) we have[18]

$$H(0) = 1 + K$$

We also obtain $H(+j\infty) = 1 + K$ for the high-shelving filter.

A further nontrivial moment is that the definition of the cutoff at $\omega = 1$ for such filters is not really convenient. Indeed, looking at the amplitude response graphs in Fig. 2.11 and Fig. 2.12 we would rather wish to have the cutoff point positioned exactly at the middle of the respective slopes. Let's find

[18] $H(0) = 1 + K$ is not a fully trivial result here. We have it only because the lowpass filter doesn't change the signal's phase at $\omega = 0$. If instead it had e.g. inverted the phase, then we would have obtained $1 - K$ here.

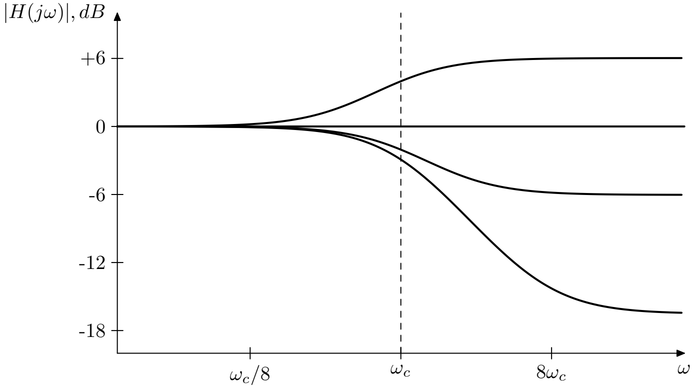


Figure 2.12: Amplitude response of a 1-pole high-shelving filter (for various K).

where the middle is. E.g. for the lowpass (and remembering that both scales of the graph are logarithmic) we first find the mid-height, which is the geometric average of the shelf's gain and the unit gain: $\sqrt{1+K}$. Then we need to find ω at which the amplitude response is $\sqrt{1+K}$:

$$\left|1+K\frac{1}{j\omega+1}\right|^2=\left|\frac{j\omega+1+K}{j\omega+1}\right|^2=\frac{(1+K)^2+\omega^2}{1+\omega^2}=1+K$$

from where

$$1+2K+K^2+\omega^2=1+K+\omega^2+K\omega^2$$

and

$$K + K^2 = K\omega^2$$

from where

$$\omega = \sqrt{1+K}$$

This is the "intuitive" cutoff position for the low-shelving filter built from a unit-cutoff lowpass filter. Respectively, given the "intuitive" cutoff position of the low-shelving filter, we need to divide it by $\sqrt{1+K}$ to obtain the cutoff of the underlying lowpass filter.

Similarly, for the highpass:

$$\left|1 + K\frac{j\omega}{j\omega+1}\right|^2 = \left|\frac{1+j\omega(K+1)}{j\omega+1}\right|^2 = \frac{1+(1+K)^2\omega^2}{1+\omega^2} = 1+K$$

from where

$$1 + \omega^2(1 + 2K + K^2) = 1 + K + \omega^2 + K\omega^2$$

and

$$\omega^2(K + K^2) = K$$

from where

$$\omega = \frac{1}{\sqrt{1+K}}$$

This is the "intuitive" cutoff position for the high-shelving filter built from a unit-cutoff highpass filter. Respectively, given the "intuitive" cutoff position of the high-shelving filter, we need to multiply it by $\sqrt{1+K}$ to obtain the cutoff of the underlying highpass filter.

2.12 Allpass filter

By subtracting the highpass output from the lowpass output of the multimode filter we obtain the *allpass filter*:

$$H(s) = H_{\text{LP}}(s) - H_{\text{HP}}(s) = \frac{1}{1+s} - \frac{s}{1+s} = \frac{1-s}{1+s}$$

The amplitude response of the allpass filter is always unity:

$$|H(j\omega)| = 1 \qquad \forall\omega$$

Indeed, the numerator $1-j\omega$ and the denominator $1+j\omega$ of the frequency response are mutually conjugate, therefore they have equal magnitudes.

The allpass filter is used because of its phase response (Fig. 2.13). That is sometimes we wish to change the phases of the signal's partials without changing their amplitudes. The most common VA use for the allpass filters is probably in phasers.

We could also subtract the lowpass from the highpass:

$$H(s) = \frac{s}{s+1} - \frac{1}{s+1} = \frac{s-1}{1+s}$$

Apparently the result differs from the previous one only by the inverted phase.

In regards to the unit amplitude response of the 1-pole allpass filter, we could have simply noticed that the zero and the pole of the filter are mutually symmetric relatively to the imaginary axis. This is a general property of differential allpass filters:

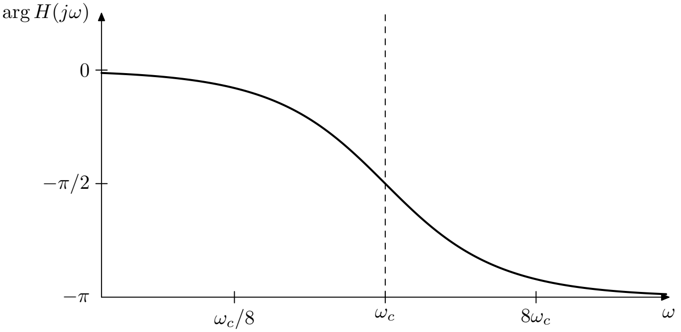


Figure 2.13: Phase response of a 1-pole allpass filter.

their poles and zeros always come in pairs, located symmetrically relatively to the imaginary axis (since the poles of a stable filter have to be in the left complex semiplane, the zeros will be in the right complex semiplane). Expressing the transfer function's numerator and denominator in the multiplicative form, we have

$$|H(s)| = \left|\frac{\prod_{n=1}^{N}(s-z_n)}{\prod_{n=1}^{N}(s-p_n)}\right| = \frac{\prod_{n=1}^{N}|s-z_n|}{\prod_{n=1}^{N}|s-p_n|}$$

where p_n and z_n are poles and zeros. If each pair p_n and z_n is mutually symmetric relatively to the imaginary axis ($p_n = -z_n^*$), then the factors $|j\omega - z_n|$ and $|j\omega - p_n|$ of the amplitude response are always equal, thus the amplitude response is always unity.

2.13 Transposed multimode filter

We could apply the *transposition* to the block diagram in Fig. 2.10. The transposition process is defined as reverting the direction of all signal flow, where forks turn into summators and vice versa (Fig. 2.14).[19] The transposition keeps the transfer function relationship within each pair of an input and an output (where the input becomes the output and vice versa). Thus in Fig. 2.14 we have a lowpass and a highpass input and a single output.

The transposed multimode filter has less practical use than the nontransposed one in Fig. 2.10. However, one particular usage case is feedback shaping. Imagine we are mixing an input signal $x_{\text{in}}(t)$ with a feedback signal $x_{\text{fbk}}(t)$, and we wish to filter each one of those by a 1-pole filter, and the cutoffs of these 1-pole filters are identical. That is, the transfer functions of those filters share a common denominator. Then we could use a single transposed 1-pole multimode filter as in Fig. 2.15.

The mixing coefficients A, B, C and D will define the numerators of the respective two transfer functions (in exactly the same way as we have been mixing the outputs of a nontrans-

[19] The inverting input of the summator in the transposed version was obtained from the respective inverting input of the summator in the nontransposed version as follows. First the inverting input is replaced by an explicit inverting gain element (gain factor -1), then the transposition is performed, then the inverting gain is merged into the new summator.

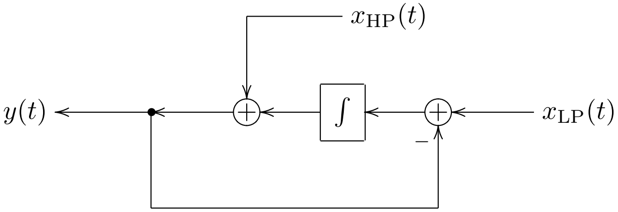


Figure 2.14: A 1-pole transponsed multimode filter.

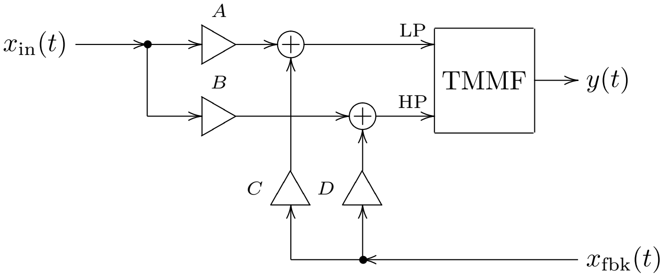


Figure 2.15: A transposed multimode filter (TMMF) used for feedback signal mixing.

posed multimode filter), whereas the denominator will be $s+\omega_c$, where ω_c is the cutoff of the transposed multimode filter.

SUMMARY

The analog 1-pole filter implementations are built around the idea of the multimode 1-pole filter in Fig. 2.10. The transfer functions of the lowpass and highpass 1-pole filters are

$$H_{\text{LP}}(s) = \frac{\omega_c}{s + \omega_c}$$

and

$$H_{\text{HP}}(s) = \frac{s}{s + \omega_c}$$

respectively. Other 1-pole filter types can be built by combining the lowpass and the highpass signals.

Chapter 3

Time-discretization

Now that we have introduced the basic ideas of analog filter analysis, we will develop an approach to convert analog filter models to the discrete time.

3.1 Discrete-time signals

The discussion of the basic concepts of discrete-time signal representation and processing is outside the scope of this book. We are assuming that the reader is familiar with the basic concepts of discrete-time signal processing, such as sampling, sampling rate, sampling period, Nyquist frequency, analog-to-digital and digital-to-analog signal conversion. However we are going to make some remarks in this respect.

As many other texts do, we will use the square bracket notation to denote discrete-time signals and round parentheses notation to denote continuous-time signals: e.g. $x[n]$ and $x(t)$.

We will often assume a unit sampling rate $f_s = 1$ (and, respectively, a unit sampling period $T = 1$), which puts the Nyquist frequency at $1/2$, or, in the circular frequency terms, at π. Apparently, this can be achieved simply by a corresponding choice of time units.

Theoretical DSP texts typically state that discrete-time signals have periodic frequency spectra. This might be convenient for certain aspects of theoretical analysis such as analog-to-digital and digital-to-analog signal conversion, but it's highly unintuitive otherwise. It would be more intuitive, whenever talking of a discrete-time signal, to imagine an ideal DAC connected to this signal, and think that the discrete-time signal represents the respective continuous-time signal produced by such DAC. Especially, since by sampling this continuous-time signal we obtain the original discrete-time signal again. So the DAC and ADC conversions are exact inverses of each other (in this case). Now, the continuous-time signal produced by such DAC doesn't contain any partials above the Nyquist frequency. Thus, its Fourier integral representation (assuming $T = 1$) is

$$x[n] = \int_{-\pi}^{\pi} X(\omega)e^{j\omega n}\,\frac{d\omega}{2\pi}$$

and its Laplace integral representation is

$$x[n] = \int_{\sigma-j\pi}^{\sigma+j\pi} X(s)e^{sn}\,\frac{ds}{2\pi j}$$

Introducing notation $z = e^s$ and noticing that

$$\mathrm{d}s = \mathrm{d}(\log z) = \frac{\mathrm{d}z}{z}$$

we can rewrite the Laplace integral as

$$x[n] = \oint X(z)z^n \,\frac{\mathrm{d}z}{2\pi jz}$$

(where $X(z)$ is apparently a different function than $X(s)$) where the integration is done counterclockwise along a circle of radius e^σ centered at the complex plane's origin:[1]

$$z = e^s = e^{\sigma+j\omega} = e^\sigma \cdot e^{j\omega} \qquad (-\pi \le \omega \le \pi) \tag{3.1}$$

We will refer the representation (3.1) as the *z-integral.*[2] The function $X(z)$ is referred to as the *z-transform* of $x[n]$.

In case of non-unit sampling period $T \neq 1$ the formulas are the same, except that the frequency-related parameters get multiplied by T (or divided by f_s), or equivalently, the n index gets multiplied by T in continuous-time expressions:[3]

$$x[n] = \int_{-\pi f_s}^{\pi f_s} X(\omega)e^{j\omega Tn}\,\frac{\mathrm{d}\omega}{2\pi}$$

[1] As with Laplace transform, sometimes there are no restrictions on the radius e^σ of the circle, sometimes there are.

[2] A more common term for (3.1) is the *inverse z-transform*, but we will prefer the *z-integral* term for the same reason as with Fourier and Laplace integrals.

[3] Formally the σ parameter of the Laplace integral (and z-integral) should have been multiplied by T as well, but it doesn't matter, since this parameter is chosen rather arbitrarily.

$$x[n] = \int_{\sigma-j\pi f_s}^{\sigma+j\pi f_s} X(s)e^{sTn}\,\frac{\mathrm{d}s}{2\pi j}$$

$$z = e^{sT}$$

$$x[n] = \oint X(z)z^n\,\frac{\mathrm{d}z}{2\pi jz} \qquad (z = e^{\sigma+j\omega T},\ -\pi f_s \le \omega \le \pi f_s)$$

The notation z^n is commonly used for discrete-time complex exponential signals. A continuous-time signal $x(t) = e^{st}$ is written as $x[n] = z^n$ in discrete-time, where $z = e^{sT}$. The Laplace-integral amplitude coefficient $X(s)$ in $X(s)e^{st}$ then may be replaced by a z-integral amplitude coefficient $X(z)$ such as in $X(z)z^n$.

3.2 Naive integration

The most "interesting" element of analog filter block diagrams is obviously the integrator. The time-discretization for other elements is trivial, so we should concentrate on building the discrete-time models of the analog integrator.

The continuous-time integrator equation is

$$y(t) = y(t_0) + \int_{t_0}^{t} x(\tau)\,\mathrm{d}\tau$$

In discrete time we could approximate the integration by a summation of the input samples. Assuming for simplicity $T = 1$, we could have implemented a discrete-time integrator as

$$y[n] = y[n_0 - 1] + \sum_{\nu=n_0}^{n} x[\nu]$$

We will refer to the above as the *naive* digital integrator.

A pseudocode routine for this integrator could simply consist of an accumulating assignment:

```
// perform one sample tick of the integrator
integrator_output := integrator_output
                          + integrator_input;
```

It takes the current state of the integrator stored in the *integrator_output* variable and adds the current sample's value of the *integrator_input* on top of that.

In case of a non-unit sampling period $T \neq 1$ we have to multiply the accumulated input values by T:[4]

```
// perform one sample tick of the integrator
integrator_output := integrator_output
                        + integrator_input*T;
```

3.3 Naive lowpass filter

We could further apply this "naive" approach to construct a discrete-time model of the lowpass filter in Fig. 2.2. We will use the naive integrator as a basis for this model.

Let the x variable contain the current input sample of the filter. Considering that the output of the filter in Fig. 2.2 coincides with the output of the integrator, let the y variable contain the

[4] Alternatively, we could, of course, scale the integrator's output by T, but this is less useful in practice, because the T factor will be usually combined with the cutoff gain factor ω_c preceding the integrator.

integrator state and simultaneously serve as the output sample. As we begin to process the next input sample, the y variable will contain the previous output value. At the end of the processing of the sample (by the filter model) the y variable will contain the new output sample. In this setup, the input value for the integrator is apparently $(x-y)\omega_c$, thus we simply have

```
// perform one sample tick of the lowpass filter
y := y + (x-y)*omega_c;
```

(mind that ω_c must have been scaled to the time units corresponding to the unit sample period!)

A naive discrete-time model of the multimode filter in Fig. 2.10 could have been implemented as:

```
// perform one sample tick of the multimode filter
hp := x-lp;
lp := lp + hp*omega_c;
```

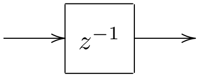
where the integrator state is stored in the *lp* variable.

The above naive implementations (and any other similar naive implementations, for that matter) work reasonably well as long as $\omega_c \ll 1$, that is the cutoff must be much lower than the sampling rate. At larger ω_c the behavior of the filter becomes rather strange, ultimately the filter gets unstable. We will now develop some theoretical means to analyse the behavior of the discrete-time filter models, figure out what are the problems with the naive implementations, and then introduce another discretization approach.

3.4 Block diagrams

Let's express the naive discrete-time integrator in the form of a discrete-time block diagram. The discrete-time block diagrams are constructed from the same elements as continuous-time block diagrams, except that instead of integrators they have *unit delays*. A unit delay simply delays the signal by one sample. That is the output of a unit delay comes "one sample late" compared to the input. Apparently, the implementation of a unit delay requires a variable, which will be used to store the new incoming value and keep it there until the next sample. Thus, a unit delay element has a *state*, while the other block diagram elements are obviously stateless. This makes the unit delays in a way similar to the integrators in the analog block diagrams, where the integrators are the only elements with a state.

A unit delay element in a block diagram is denoted as:



The reason for the notation z^{-1} will be explained a little bit later. Using a unit delay, we can create a block diagram for our naive integrator (Fig. 3.1). For an arbitrary sampling period we obtain the structure in Fig. 3.2. For an integrator with embedded cutoff gain we can combine the ω_c gain element with the T gain element (Fig. 3.3). Notice that the integrator thereby becomes invariant to the choice of the time units, since $\omega_c T$ is invariant to this choice.

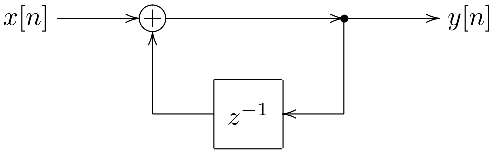


Figure 3.1: Naive integrator for $T = 1$.

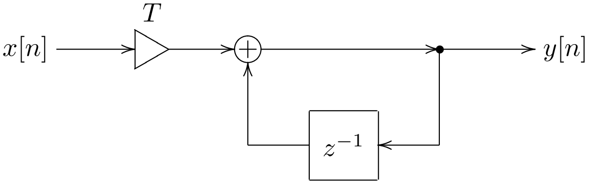


Figure 3.2: Naive integrator for arbitrary T.

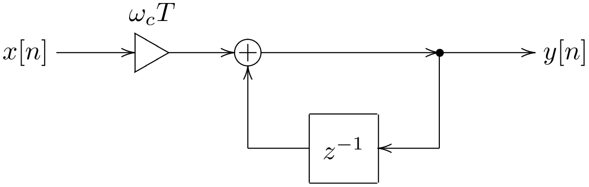


Figure 3.3: Naive integrator with embedded cutoff.

Now let's construct the block diagram of the naive 1-pole lowpass filter. Recalling the implementation routine:

```
// perform one sample tick of the lowpass filter
y := y + (x-y)*omega_c;
```

we obtain the diagram in Fig. 3.4. The z^{-1} element in the feedback from the filter's output to the leftmost summator is occurring due to the fact that we are picking up the *previous* value of y in the routine when computing the difference $x - y$.

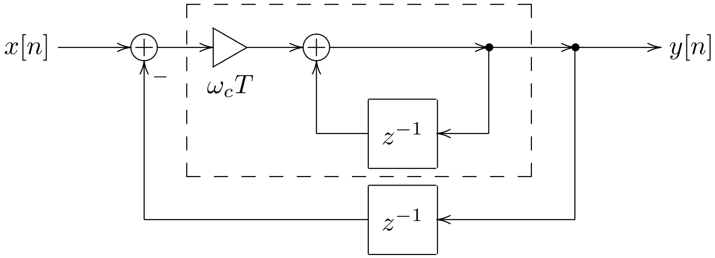


Figure 3.4: Naive 1-pole lowpass filter (the dashed line denotes the integrator).

This unit delay occurring in the discrete-time feedback is a common problem in discrete-time implementations. This problem is solvable, however it doesn't make too much sense to solve it for the naive integrator-based models, as the increased complexity doesn't justify the improvement in sound. We will address the problem of the zero-delay discrete-time feedback later, for now we'll concentrate on the naive model in Fig. 3.4. This model can be simplified a bit, by combining the two z^{-1} elements into one (Fig. 3.5), so that the block diagram explicitly contains a single state variable (as does its pseudocode counterpart).

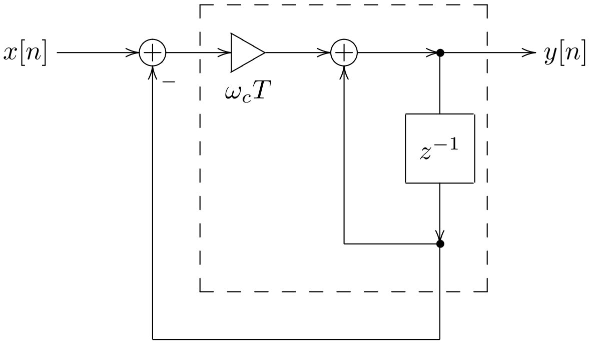
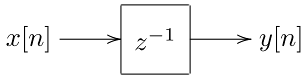


Figure 3.5: Naive 1-pole lowpass filter with just one z^{-1} element (the dashed line denotes the integrator).

3.5 Transfer function

Let $x[n]$ and $y[n]$ be respectively the input and the output signals of a unit delay:



For a complex exponential input $x[n] = e^{sn} = z^n$ we obtain

$$y[n] = e^{s(n-1)} = e^{sn}e^{-s} = z^n z^{-1} = z^{-1}x[n]$$

That is

$$y[n] = z^{-1}x[n]$$

That is, z^{-1} is the *transfer function* of the unit delay! It is common to express discrete-time transfer functions as functions of z rather than functions of s. The reason is that in this case the transfer functions are nonstrictly proper[5] rational functions, similarly to the continuous-time case, which is pretty convenient. So, for a unit delay we could write $H(z) = z^{-1}$.

Now we can obtain the transfer function of the naive integrator in Fig. 3.1. Suppose[6] $x[n] = X(z)z^n$ and $y[n] = Y(z)z^n$, or shortly, $x = X(z)z^n$ and $y = Y(z)z^n$. Then the output of the z^{-1} element is yz^{-1}. The output of the summator is then $x + yz^{-1}$, thus

$$y = x + yz^{-1}$$

[5] Under the assumption of causality, which holds if the system is built of unit delays.

[6] As in continuous-time case, we take for granted the fact that complex exponentials z^n are eigenfunctions of discrete-time linear time-invariant systems.

from where

$$y(1 - z^{-1}) = x$$

and

$$H(z) = \frac{y}{x} = \frac{1}{1 - z^{-1}}$$

This is the transfer function of the naive integrator (for $T = 1$).

It is relatively common to express discrete-time transfer functions as rational functions of z^{-1} (like the one above) rather than rational functions of z. However, for the purposes of the analysis it is also often convenient to have them expressed as rational functions of z (particularly, for finding their poles and zeros). We can therefore multiply the numerator and the denominator of the above $H(z)$ by z, obtaining:

$$H(z) = \frac{z}{z - 1}$$

Since $z = e^s$, the *frequency response* is obtained as $H(e^{j\omega})$. The amplitude and phase responses are $\left|H(e^{j\omega})\right|$ and $\arg H(e^{j\omega})$ respectively.[7]

For $T \neq 1$ we obtain

$$H(z) = T\frac{z}{z - 1}$$

and, since $z = e^{sT}$, the frequency response is $H(e^{j\omega T})$.

Now let's obtain the transfer function of the naive 1-pole lowpass filter in Fig. 3.5, where, for the simplicity of notation,

[7] Another way to look at this is to notice that in order for z^n to be a complex sinusoid $e^{j\omega n}$ we need to let $z = e^{j\omega}$.

we assume $T = 1$. Assuming complex exponentials $x = X(z)z^n$ and $y = Y(z)z^n$ we have x and yz^{-1} as the inputs of the first summator. Respectively the integrator's input is $\omega_c(x - yz^{-1})$. And the integrator output is the sum of yz^{-1} and the integrator's input. Therefore

$$y = yz^{-1} + \omega_c(x - yz^{-1})$$

From where

$$\bigl(1 - (1 - \omega_c)z^{-1}\bigr)y = \omega_c x$$

and

$$H(z) = \frac{y}{x} = \frac{\omega_c}{1 - (1 - \omega_c)z^{-1}} = \frac{\omega_c z}{z - (1 - \omega_c)}$$

The transfer function for $T \neq 1$ can be obtained by simply replacing ω_c by $\omega_c T$.

The respective amplitude response is plotted in Fig. 3.6. Comparing it to the amplitude response of the analog prototype we can observe serious deviation closer to the Nyquist frequency. The phase response (Fig. 3.7) has similar deviation problems.

3.6 Poles

Discrete-time block diagrams are differing from continuous-time block diagrams only by having z^{-1} elements instead of integrators. Recalling that the transfer function of an integrator is s^{-1}, we conclude that from the formal point of view the difference is purely notational.

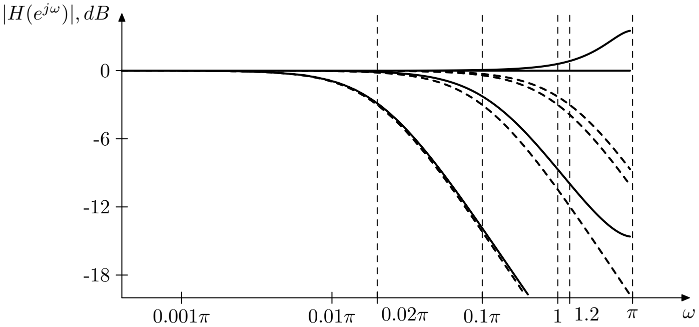


Figure 3.6: Amplitude response of a naive 1-pole lowpass filter for a number of different cutoffs. Dashed curves represent the respective analog filter responses for the same cutoffs.

Now, the transfer functions of continuous-time block diagrams are nonstrictly proper rational functions of s. Respectively, the transfer functions of discrete-time block diagrams are nonstrictly proper rational functions of z.

Thus, discrete-time transfer functions will have poles and zeros in a way similar to continuous-time transfer functions. Similarly to continuous-time transfer functions, the poles will define the stability of a linear time-invariant filter. Consider that $z = e^{sT}$ and recall the stability criterion $\operatorname{Re} s < 0$ (where $s = p_n$, where p_n are the poles). Apparently, $\operatorname{Re} s < 0 \iff |z| < 1$. We might therefore intuitively expect the discrete-time stability criterion to be $|p_n| < 1$ where p_n are the discrete-time poles.

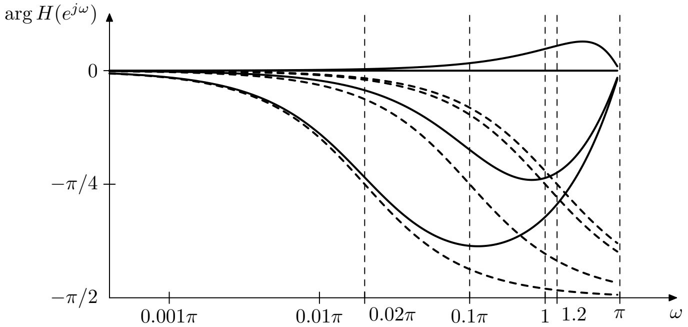


Figure 3.7: Phase response of a naive 1-pole lowpass filter for a number of different cutoffs. Dashed curves represent the respective analog filter responses for the same cutoffs.

This is indeed the case, a linear time-invariant difference system[8] is stable if and only if all its poles are located inside the unit circle.

3.7 Trapezoidal integration

Instead of naive integration, we could attempt using the trapezoidal integration method ($T = 1$):

[8] Difference systems can be defined as those, whose block diagrams consist of gains, summators and unit delays. More precisely those are causal difference systems. There are also difference systems with a lookahead into the future, but we don't consider them in this book.

```
// perform one sample tick of the integrator
integrator_output := integrator_output +
    (integrator_input +
                 previous_integrator_input)/2;
previous_integrator_input := integrator_input;
```

Notice that now we need two state variables per integrator: *integrator_output* and *previous_integrator_input*. The block diagram of a trapezoidal integrator is shown in Fig. 3.8. We'll refer to this integrator as a *direct form I trapezoidal integrator*. The reason for this term will be explained later.

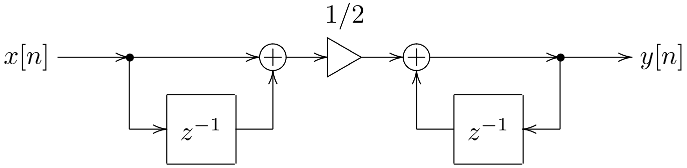


Figure 3.8: Direct form I trapezoidal integrator ($T = 1$).

We could also construct a trapezoidal integrator implementation with only a single state variable. Consider the expression for the trapezoidal integrator's output:

$$y[n] = y[n_0 - 1] + \sum_{\nu=n_0}^{n} \frac{x[\nu-1] + x[\nu]}{2} \tag{3.2}$$

Suppose $y[n_0-1]=0$ and $x[n_0-1]$=0, corresponding to a zero initial state (recall that both $y[n_0-1]$ and $x[n_0-1]$ are technically stored in the z^{-1} elements). Then

$$\begin{aligned}
y[n] &= \sum_{\nu=n_0}^{n}\frac{x[\nu-1]+x[\nu]}{2} = \frac{1}{2}\left(\sum_{\nu=n_0}^{n}x[\nu-1]+\sum_{\nu=n_0}^{n}x[\nu]\right) = \\
&= \frac{1}{2}\left(\sum_{\nu=n_0+1}^{n}x[\nu-1]+\sum_{\nu=n_0}^{n}x[\nu]\right) = \\
&= \frac{1}{2}\left(\sum_{\nu=n_0}^{n-1}x[\nu]+\sum_{\nu=n_0}^{n}x[\nu]\right) = \\
&= \frac{u[n-1]+u[n]}{2}
\end{aligned}$$

where

$$u[n]=\sum_{\nu=n_0}^{n}x[\nu]$$

Now notice that $u[n]$ is the output of a naive integrator, whose input signal is $x[n]$. At the same time $y[n]$ is the average of the previous and the current output values of the naive integrator. This can be implemented by the structure in Fig. 3.9. Similar considerations apply for nonzero initial state. We'll refer to the integrator in Fig. 3.9 as a *direct form II* or *canonical* trapezoidal integrator. The reason for this term will be explained later.

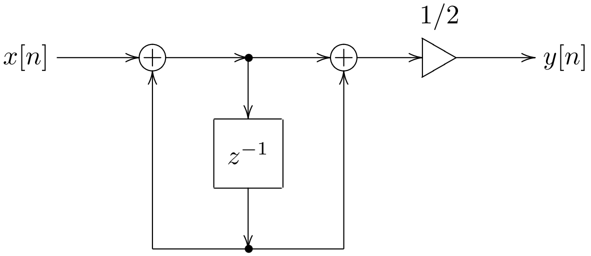


Figure 3.9: Direct form II (canonical) trapezoidal integrator ($T = 1$).

We can develop yet another form of the bilinear integrator with a single state variable. Let's rewrite (3.2) as

$$y[n] = y[n_0 - 1] + \frac{x[n_0 - 1]}{2} + \sum_{\nu=n_0}^{n-1} x[\nu] + \frac{x[n]}{2}$$

and let

$$u[n-1] = y[n] - \frac{x[n]}{2} = y[n_0 - 1] + \frac{x[n_0 - 1]}{2} + \sum_{\nu=n_0}^{n-1} x[\nu]$$

Notice that

$$y[n] = u[n-1] + \frac{x[n]}{2} \tag{3.3}$$

and

$$u[n] = u[n-1] + x[n] = y[n] + \frac{x[n]}{2} \tag{3.4}$$

Expressing (3.3) and (3.4) in a graphical form, we obtain the structure in Fig. 3.10, where the output of the z^{-1} block corresponds to $u[n-1]$. We'll refer to the integrator in Fig. 3.10 as a *transposed direct form II* or *transposed canonical* trapezoidal integrator. The reason for this term will be explained later.

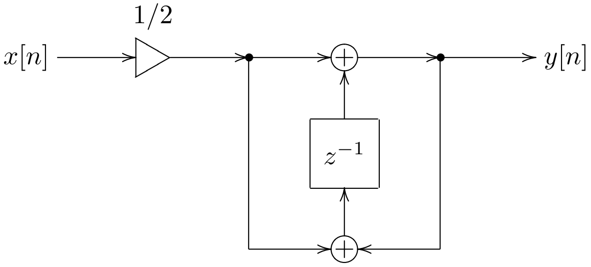


Figure 3.10: Transposed direct form II (transposed canonical) trapezoidal integrator ($T = 1$).

The positioning of the 1/2 gain prior to the integrator in Fig. 3.10 is quite convenient, because we can combine the 1/2 gain with the cutoff gain into a single gain element. In case of an arbitrary sampling period we could also include the T factor into the same gain element, thus obtaining the structure in Fig. 3.11. A similar trick can be performed for the other two integrators, if we move the 1/2 gain element to the input of the respective integrator. Since the integrator is a linear time-invariant system, this doesn't affect the integrator's behavior in a slightest way.

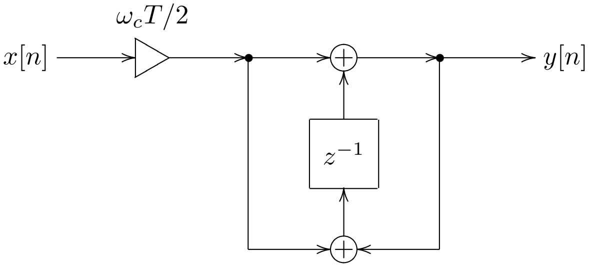


Figure 3.11: Transposed direct form II (transposed canonical) trapezoidal integrator with “embedded” cutoff gain.

Typically one would prefer the direct form II integrators to the direct form I integrator, because the former have only one state variable. In this book we will mostly use the transposed direct form II integrator, because this is resulting in slightly simpler zero-delay feedback equations and also offers a nice possibility for the internal saturation in the integrator.

The transfer functions of all three integrators are identical. Let’s obtain e.g. the transfer function of the transposed canonical integrator (in Fig. 3.10). Let u be the output signal of the z^{-1} element. Assuming signals of the exponential form z^n, we have

$$u = \left(\frac{x}{2} + y\right) z^{-1}$$

$$y = \frac{x}{2} + u$$

from where

$$u = y - \frac{x}{2}$$

and

$$y - \frac{x}{2} = \left(\frac{x}{2} + y\right) z^{-1}$$

or

$$\left(y - \frac{x}{2}\right) z = \frac{x}{2} + y$$

from where

$$y(z-1) = \frac{x}{2}(z+1)$$

and the transfer function of the trapezoidal integrator is thus

$$H(z) = \frac{y}{x} = \frac{1}{2} \cdot \frac{z+1}{z-1}$$

For an arbitrary T one has to multiply the result by T, to take the respective gain element into account:

$$H(z) = \frac{T}{2} \cdot \frac{z+1}{z-1}$$

If also the cutoff gain is included, we obtain

$$H(z) = \frac{\omega_c T}{2} \cdot \frac{z+1}{z-1}$$

One can obtain the same results for the other two integrators.

What is so special about this transfer function, that makes the trapezoidal integrator so superior to the naive one, is to be discussed next.

3.8 Bilinear transform

Suppose we take an arbitrary continuous-time block diagram, like the familiar lowpass filter in Fig. 2.2 and replace all continuous-time integrators by discrete-time trapezoidal integrators. On the transfer function level, this will correspond to replacing all s^{-1} with $\frac{T}{2} \cdot \frac{z+1}{z-1}$. That is, technically we perform a subsitution

$$s^{-1} = \frac{T}{2} \cdot \frac{z+1}{z-1}$$

in the transfer function expression.

It would be more convenient to write this substitution explicitly as

$$s = \frac{2}{T} \cdot \frac{z-1}{z+1} \tag{3.5}$$

The substitution (3.5) is referred to as the *bilinear transform*, or shortly BLT. For that reason we can also refer to trapezoidal integrators as *BLT integrators*. Let's figure out, how does the bilinear transform affect the frequency response of the filter, that is, what is the relationship between the original continuous-time frequency response prior to the substitution and the resulting discrete-time frequency response after the substitution.

Let $H_a(s)$ be the original continuous-time transfer function. Then the respective discrete-time transfer function is

$$H_d(z) = H_a\left(\frac{2}{T} \cdot \frac{z-1}{z+1}\right) \tag{3.6}$$

Respectively, the discrete-time frequency response is

$$\begin{aligned}H_d(e^{j\omega T}) &= H_a\left(\frac{2}{T}\cdot\frac{e^{j\omega T}-1}{e^{j\omega T}+1}\right)=\\ &= H_a\left(\frac{2}{T}\cdot\frac{e^{j\omega T/2}-e^{-j\omega T/2}}{e^{j\omega T/2}+e^{-j\omega T/2}}\right)=\\ &= H_a\left(\frac{2}{T}j\tan\frac{\omega T}{2}\right)\end{aligned}$$

Notice that $H_a(s)$ in the last expression is evaluated on the imaginary axis!!! That is, the bilinear transform maps the imaginary axis in the s-plane to the unit circle in the z-plane! Now, $H_a\left(\frac{2}{T}j\tan\frac{\omega T}{2}\right)$ is the analog frequency response evaluated at $\frac{2}{T}\tan\frac{\omega T}{2}$. That is, the digital frequency response at ω is equal to the analog frequency response at $\frac{2}{T}\tan\frac{\omega T}{2}$. This means that the analog frequency response in the range $0 \leq \omega < +\infty$ is mapped into the digital frequency range $0 \leq \omega T < \pi$ $(0 \leq \omega < \pi f_s)$, that is from zero to Nyquist![9] Denoting the analog frequency as ω_a and the digital frequency as ω_d we can express the argument mapping of the frequency response function as

$$\omega_a = \frac{2}{T}\tan\frac{\omega_d T}{2} \tag{3.7}$$

or, in a more symmetrical way

$$\frac{\omega_a T}{2} = \tan\frac{\omega_d T}{2} \tag{3.8}$$

[9] A similar mapping obviously occurs for the negative frequencies.

Notice that for frequencies much smaller that Nyquist frequency we have $\omega T \ll 1$ and respectively $\omega_a \approx \omega_d$.

This is what is so unique about the bilinear transform. It simply warps the frequency range $[0, +\infty)$ into the zero-to-Nyquist range, but otherwise doesn't change the frequency response at all! Considering in comparison a naive integrator, we would have obtained:

$$\begin{gathered}
s^{-1} = \frac{z}{z-1} \\
s = \frac{z-1}{z} \qquad (3.9) \\
H_d(z) = H_a\left(\frac{z-1}{z}\right) \\
H_d(e^{j\omega}) = H_a\left(\frac{e^{j\omega}-1}{e^{j\omega}}\right) = H_a\left(1-e^{-j\omega}\right)
\end{gathered}$$

which means that the digital frequency response is equal to the analog transfer function evaluated on a circle of radius 1 centered at $s = 1$. This hardly defines a clear relationship between the two frequency responses.

So, by simply replacing the analog integrators with digital trapezoidal integrators, we obtain a digital filter whose frequency response is essentially the same as the one of the analog prototype, except for the frequency warping. Particularly, the relationship between the amplitude and phase responses of the filter is fully preserved, which is particularly highly important if the filter is to be used as a building block in a larger filter. Very close to perfect!

Furthermore, the bilinear transform maps the left complex semiplane in the s-domain into the inner region of the unit circle in the z-domain. Indeed, let's obtain the inverse bilinear transform formula. From (3.5) we have

$$(z+1)\frac{sT}{2} = z - 1$$

from where

$$1 + \frac{sT}{2} = z\left(1 - \frac{sT}{2}\right)$$

and

$$z = \frac{1 + \frac{sT}{2}}{1 - \frac{sT}{2}} \tag{3.10}$$

The equation (3.10) defines the *inverse bilinear transform.* Now, if $\operatorname{Re} s < 0$, then, obviously

$$\left|1 + \frac{sT}{2}\right| < \left|1 - \frac{sT}{2}\right|$$

and $|z| < 1$. Thus, the left complex semiplane in the s-plane is mapped to the inner region of the unit circle in the z-plane. In the same way one can show that the right complex semiplane is mapped to the outer region of the unit circle. And the imaginary axis is mapped to the unit circle itself. Comparing the stability criterion of analog filters (the poles must be in the left complex semiplane) to the one of digital filters (the poles must be inside the unit circle), we conclude that the bilinear transform exactly preserves the stability of the filters!

In comparison, for a naive integrator replacement we would have the following. Inverting the (3.9) substitution we obtain

$$sz = z - 1$$

$$z(1 - s) = 1$$

and

$$z = \frac{1}{1 - s}$$

Assuming $\operatorname{Re} s < 0$ and considering that in this case

$$\left|z - \frac{1}{2}\right| = \left|\frac{1}{1-s} - \frac{1}{2}\right| = \left|\frac{1 - \frac{1}{2} + \frac{s}{2}}{1-s}\right| = \left|\frac{1}{2} \cdot \frac{1+s}{1-s}\right| < \frac{1}{2}$$

we conclude that the left semiplane is mapped into a circle of radius 0.5 centered at $z = 0.5$. So the naive integrator overpreserves the stability, which is not nice, since we would rather have digital filters behaving as closely to their analog prototypes as possible. Considering that this comes in a package with a poor frequency response transformation, we should rather stick with trapezoidal integrators.

So, let's replace e.g. the integrator in the familiar lowpass filter structure in Fig. 2.2 with a trapezoidal integrator. Performing the integrator replacement, we obtain the structure in Fig. 3.12. We will refer to the trapezoidal integrator replacement method as the *topology-preserving transform* (TPT) method. This term will be explained and properly introduced later. For now, before we simply attempt to implement the structure in Fig. 3.12 in code, we should become aware of a few further issues.

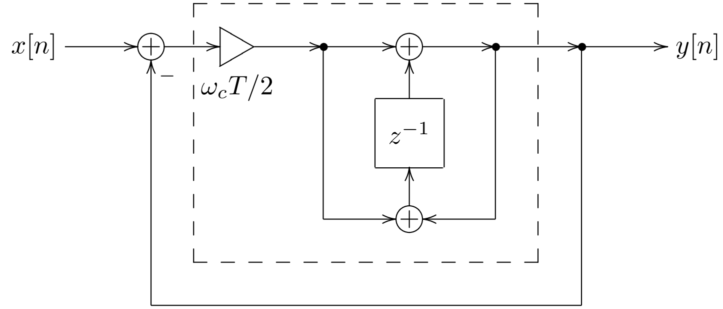


Figure 3.12: 1-pole TPT lowpass filter (the dashed line denotes the trapezoidal integrator).

3.9 Cutoff prewarping

Suppose we are using the lowpass filter structure in Fig. 3.12 and we wish to have its cutoff at ω_c. If we however simply put this ω_c parameter into the respective integrator gain element $\omega_c T/2$, our frequency response at the cutoff will be different from the expected one. Considering the transfer function of an analog 1-pole lowpass filter (2.2), at the cutoff we expect

$$H(j\omega_c) = \frac{\omega_c}{\omega_c + j\omega_c} = \frac{1}{1+j}$$

corresponding to a -3dB drop in amplitude and a 45° phase shift. However, letting $\omega_a = \omega_c$ in (3.8) we will obtain some $\omega_d \neq \omega_c$. That is the cutoff point of the analog frequency response will be mapped to some other frequency ω_d in the digital frequency response (Fig. 3.13). This is the result of the frequency axis warping by the bilinear transform.[10]

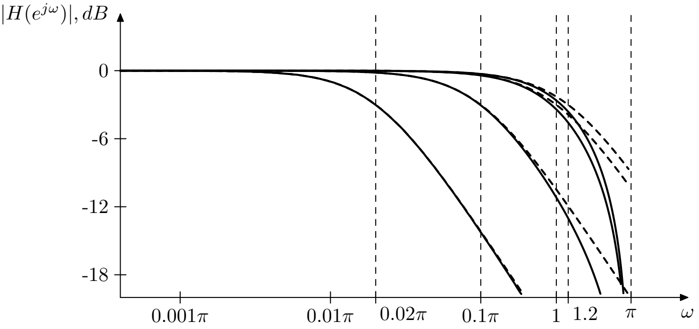


Figure 3.13: Amplitude response of an unprewarped bilinear-transformed 1-pole lowpass filter for a number of different cutoffs. Dashed curves represent the respective analog filter responses for the same cutoffs. Observe the difference between the analog and digital responses at each cutoff.

[10] The response difference at the cutoff in Fig. 3.13 might seem negligible. However it will be even higher for cutoffs closer to Nyquist. Also for filters with strong resonance the detuning of the cutoff by frequency warping might be way more noticeable.

However, if we desire to have the $1/(1+j)$ frequency response exactly at $\omega_d = \omega_c$, we can simply apply (3.7) to $\omega_d = \omega_c$, thereby obtaining some ω_a. This ω_a should be used in the gain element of the integrator, that is the gain should be $\omega_a T/2$ instead of $\omega_c T/2$. This cutoff substitution is referred to as the *cutoff prewarping.* The result of the cutoff prewarping is illustrated in Fig. 3.14.

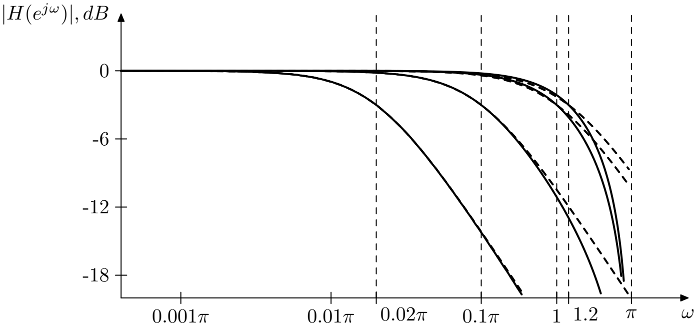


Figure 3.14: Amplitude response of an prewarped bilinear-transformed 1-pole lowpass filter for a number of different cutoffs. Dashed curves represent the respective analog filter responses for the same cutoffs. Observe the identical values of the analog and digital responses at each cutoff.

Apparently, the importance of the cutoff prewarping grows as the cutoff values get higher. For cutoff values much lower than the Nyquist frequency the prewarping has only a minor effect.

Notice that it's possible to choose any other point for the prewarping, not necessarily the cutoff point. That is it's possible to make any single chosen point on the analog frequency response to be located at the desired digital frequency. In order to do so we first choose ω_d of interest, then use (3.7) to find the respective ω_a. Now we want a particular point on the analog frequency response to be located at ω_a, which can be achieved by a proper choice of the analog cutoff value. Now we put this cutoff value into the integrators and that's it!

3.10 Zero-delay feedback

There is a further problem with the trapezoidal integrator replacement in the TPT method. Replacing the integrators with trapezoidal ones introduces *delayless feedback loops* (that is, feedback loops not containing any delay elements) into the structure. E.g. consider the structure in Fig. 3.12. Carefully examining this structure, we find that it has a feedback loop which doesn't contain any unit delay elements. This loop goes from the leftmost summator through the gain, through the upper path of the integrator to the filter's output and back through the large feedback path to the leftmost summator.

Why is this delayless loop a problem? Let's consider for example the naive lowpass filter structure in Fig. 3.5. Suppose we don't have the respective program code representation and wish to obtain it from the block diagram. We could do it in the following way. Consider Fig. 3.15, which is the same as

Fig. 3.5, except that it labels all signal points. At the beginning of the computation of a new sample the signals A and B are already known. $A = x[n]$ is the current input sample and B is taken from the internal state memory of the z^{-1} element. Therefore we can compute $C = A - B$. Then we can compute $D = (\omega_c T)C$ and finally $E = D + B$. The value of E is then stored into the internal memory of the z^{-1} element (for the next sample computation) and is also sent to the output as the new $y[n]$ value. Easy, right?

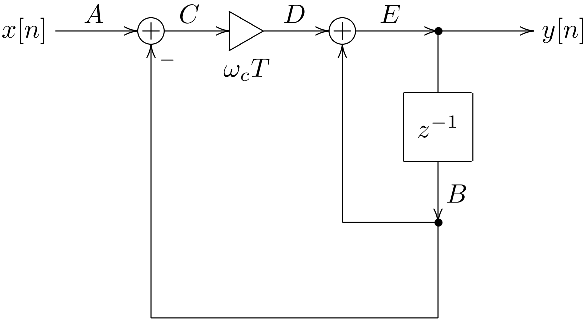


Figure 3.15: Naive 1-pole lowpass filter and the respective signal computation order.

Now the same approach doesn't work for the structure in Fig. 3.12. Because there is a delayless loop, we can't find a starting point for the computation within that loop.

The classical way of solving this problem is exactly the same as what we had in the naive approach: introduce a z^{-1} into the delayless feedback, turning it into a feedback containing a unit delay (Fig. 3.16). Now there are no delayless feedback paths and we can arrange the computation order in a way similar to Fig. 3.15. This however destroys the resulting frequency response, because the transfer function is now different. In fact the obtained result is not significantly better (if better at all) than the one from the naive approach. There are some serious artifacts in the frequency response closer to the Nyquist frequency, if the filter cutoff is sufficiently high.

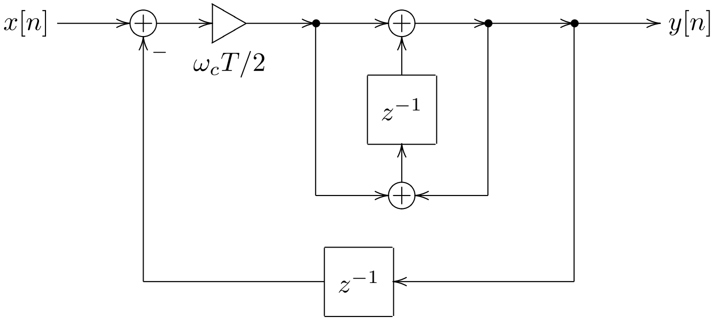


Figure 3.16: Digital 1-pole lowpass filter with a trapezoidal integrator and an extra delay in the feedback.

Therefore we shouldn't introduce any modifications into the structure and solve the *zero-delay feedback* problem instead. The term “zero-delay feedback” originates from the fact that we avoid introducing a one-sample delay into the feedback (like in Fig. 3.16) and instead keep the feedback delay equal to zero.

So, let’s solve the zero-delay feedback problem for the structure in Fig. 3.12. Notice that this structure simply consists of a negative feedback loop around a trapezoidal integrator, where the trapezoidal integrator structure is exactly the one from Fig. 3.11. We will now introduce the concept of the *instantaneous response* of this integrator structure.

So, consider the integrator structure in Fig. 3.11 and let $u[n]$ denote the input signal of the z^{-1} element, respectively its output will be $u[n-1]$. Since there are no delayless loops in the integrator, it’s not difficult to obtain the following expression for $y[n]$:

$$y[n] = \frac{\omega_c T}{2} x[n] + u[n-1] \tag{3.11}$$

Notice that, at the time $x[n]$ arrives at the integrator’s input, all values in the right-hand side of (3.11) are known (no unknown variables). Introducing notation

$$\begin{gathered} g = \frac{\omega_c T}{2} \\ s[n] = u[n-1] \end{gathered}$$

we have

$$y[n] = gx[n] + s[n]$$

or, dropping the discrete time argument notation for simplicity,

$$y = gx + s$$

That is, at any given time moment n, the output of the integrator y is a linear function of its input x, where the values of the parameters of this linear function are known. The g parameter doesn't depend on the internal state of the integrator, while the s parameter does depend on the internal state of the integrator. We will refer to the linear function $f(x) = gx + s$ as the *instantaneous response* of the integrator at the respective implied time moment n. The coefficient g can be referred to as the *instantaneous response gain* or simply *instantaneous gain.* The term s can be referred to as the *instantaneous response offset* or simply *instantaneous offset.*

Let's now redraw the filter structure in Fig. 3.12 as in Fig. 3.17. We have changed the notation from x to ξ in the $gx + s$ expression to avoid the confusion with the input signal $x[n]$ of the entire filter.

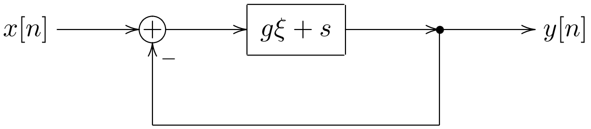


Figure 3.17: 1-pole TPT lowpass filter with the integrator in the instantaneous response form.

Now we can easily write and solve the zero-delay feedback equation. Indeed, suppose we already know the filter output $y[n]$. Then the output signal of the feedback summator is $x[n] - y[n]$ and the output of the integrator is respectively $g(x[n] -$

$y[n]) + s$. Thus

$$y[n] = g(x[n] - y[n]) + s$$

or, dropping the time argument notation for simplicity,

$$y = g(x - y) + s \tag{3.12}$$

The equation (3.12) is the zero-delay feedback equation for the filter in Fig. 3.17 (or, for that matter, in Fig. 3.12). Solving this equation, we obtain

$$y(1 + g) = gx + s$$

and respectively

$$y = \frac{gx + s}{1 + g} \tag{3.13}$$

Having found y (that is, having predicted the output $y[n]$), we can then proceed with computing the other signals in the structure in Fig. 3.12, beginning with the output of the leftmost summator.[11]

It's worth mentioning that (3.13) can be used to obtain the instantaneous response of the entire filter from Fig. 3.12. Indeed, rewriting (3.13) as

$$y = \frac{g}{1 + g}x + \frac{s}{1 + g}$$

[11] Notice that the choice of the signal point for the prediction is rather arbitrary. We could have chosen any other point within the delayless feedback loop.

and introducing notations

$$\begin{aligned}G &= \frac{g}{1+g}\\ S &= \frac{s}{1+g}\end{aligned}$$

we have

$$y = Gx + S \tag{3.14}$$

So, the instantaneous response of the entire lowpass filter in Fig. 3.12 is again a linear function of the input. We could use the expression (3.14) e.g. to solve the zero-delay feedback problem for some larger feedback loop containing a 1-pole lowpass filter.

Let's now convert the structure in Fig. 3.12 into a piece of code. We already know y from (3.14). Let's notice that the output of the $\omega_c T/2$ gain is used twice in the structure. Let v denote the output of this gain. Since $g = \omega_c T/2$, we have

$$\begin{aligned}v = g(x-y) = g\left(x - Gx - S\right) = g\left(x - \frac{g}{1+g}x - \frac{s}{1+g}\right) =\\ = g\left(\frac{1}{1+g}x - \frac{s}{1+g}\right) = g\frac{x-s}{1+g}\end{aligned} \tag{3.15}$$

Recall that s is the output value of the z^{-1} element and let u denote its input value. Then

$$y = v + s \tag{3.16}$$

and

$$u = y + v \tag{3.17}$$

The equations (3.15), (3.16) and (3.17) can be directly expressed in program code:

```
// perform one sample tick of the lowpass filter
v := (x-z1_state)*g/(1+g);
y := v + z1_state;
z1_state := y + v;
```

or instead expressed in a block diagram form (Fig. 3.18). Notice that the block diagram doesn't contain any delayless loops anymore.

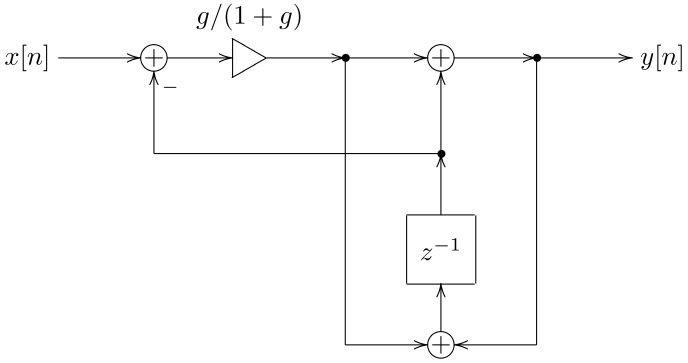


Figure 3.18: 1-pole TPT lowpass filter with resolved zero-delay feedback.

3.11 Direct forms

Consider again the equation (3.6), which describes the application of the bilinear transform to convert an analog transfer function to a digital one. There is a classical method of digital filter design which is based directly on this transformation, without using any integrator replacement techniques. In the author's experience, for music DSP needs this method typically has a largely inferior quality, compared to the TPT. Nevertheless we will describe it here for completeness and for a couple of other reasons. Firstly, it would be nice to try to analyse and understand the reasons for the problems of this method. Secondly, this method could be useful once in a while. Particularly, its deficiencies mostly disappear in the time-invariant (unmodulated, or sufficiently slowly modulated) case, while the implementation sometimes could be slightly more efficient.

Having obtained a digital transfer function from (3.6), we could observe, that, since the original analog transfer function was a rational function of s, the resulting digital transfer function will necessarily be a rational function of z. E.g. using the familiar 1-pole lowpass transfer function

$$H_a(s) = \frac{\omega_c}{s + \omega_c}$$

we obtain

$$H_d(z) = H_a\left(\frac{2}{T}\cdot\frac{z-1}{z+1}\right) = \frac{\omega_c}{\frac{2}{T}\cdot\frac{z-1}{z+1} + \omega_c} =$$

$$= \frac{\frac{\omega_c T}{2}(z+1)}{(z-1)+\frac{\omega_c T}{2}(z+1)} = \frac{\frac{\omega_c T}{2}(z+1)}{\left(1+\frac{\omega_c T}{2}\right)z - \left(1-\frac{\omega_c T}{2}\right)}$$

Now, there are standard discrete-time structures allowing an implementation of any given nonstrictly proper rational transfer function. It is easier to use these structures, if the transfer function is expressed as a rational function of z^{-1} rather than the one of z. In our particular example, we can multiply the numerator and the denominator by z^{-1}, obtaining

$$H_d(z) = \frac{\frac{\omega_c T}{2}(1+z^{-1})}{\left(1+\frac{\omega_c T}{2}\right) - \left(1-\frac{\omega_c T}{2}\right)z^{-1}}$$

The further requirement is to have the constant term in the denominator equal to 1, which can be achieved by dividing everything by $1+\omega_c T/2$:

$$H_d(z) = \frac{\frac{\frac{\omega_c T}{2}}{1+\frac{\omega_c T}{2}}(1+z^{-1})}{1-\frac{1-\frac{\omega_c T}{2}}{1+\frac{\omega_c T}{2}}z^{-1}} \tag{3.18}$$

Now suppose we have an arbitrary rational nonstrictly proper transfer function of z, expressed via z^{-1} and having the constant term in the denominator equal to 1:

$$H(z) = \frac{\displaystyle\sum_{n=0}^{N} b_n z^{-n}}{1-\displaystyle\sum_{n=1}^{N} a_n z^{-n}} \tag{3.19}$$

This transfer function can be implemented by the structure in Fig. 3.19 or by the structure in Fig. 3.20. One can verify (by computing the transfer functions of the respective structures) that they indeed implement the transfer function (3.19). There are also transposed versions of these structures, which the readers should be able to construct on their own.

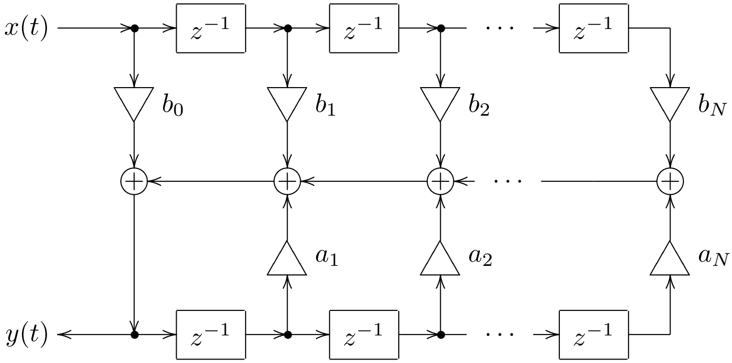


Figure 3.19: Direct form I (DF1).

Let's use the direct form II to implement (3.18). Apparently, we have

$$N = 1$$

$$b_0 = b_1 = \frac{\frac{\omega_c T}{2}}{1 + \frac{\omega_c T}{2}}$$

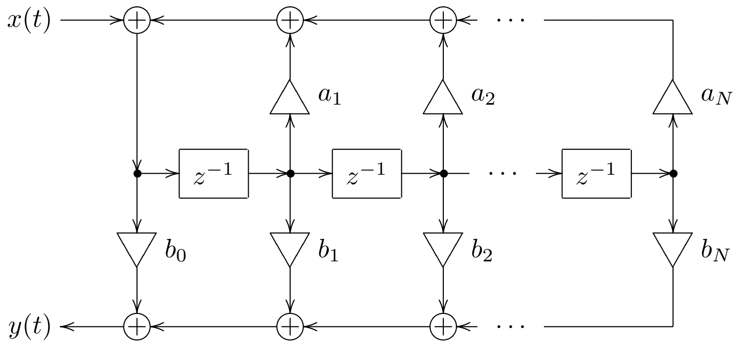


Figure 3.20: Direct form II (DF2), a.k.a. canonical form.

$$a_1 = \frac{1 - \frac{\omega_c T}{2}}{1 + \frac{\omega_c T}{2}}$$

and the direct form implementation itself is the one in Fig. 3.21 (we have merged the b_0 and b_1 coefficients into a single gain element).

In the time-invariant (unmodulated) case the performance of the direct form filter in Fig. 3.21 should be identical to the TPT filter in Fig. 3.12, since both implement the same bilinear-transformed analog transfer function (2.2). When the cutoff is modulated, however, the performance will be different.

This is very easy to understand intuitively. First, consider

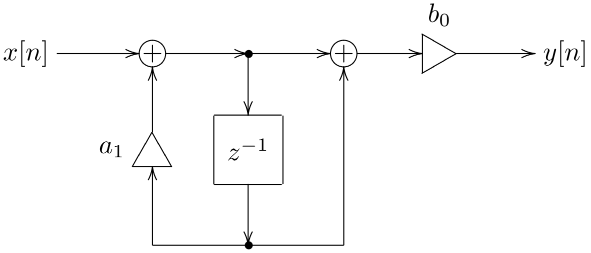
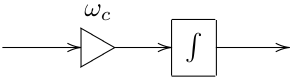
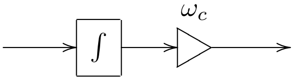


Figure 3.21: Direct form II 1-pole lowpass filter.

the two following analog structures, implementing two different ways of combining a cutoff gain with an integrator:



and



Suppose the input signal is a sine and there is a sudden jump in the cutoff parameter. In this case there will also be a sudden jump in the input of the first integrator, however the jump will be converted into a break by the integration. In the second case the jump will not be converted, because it appears after the integrator. Ignoring the problem of a DC offset possibly introduced by such jump in the first structure (because in real stable filters this DC offset will quickly disappear with time), we should say that the first structure has a better modulation performance, since the cutoff jumps do not produce so audible clicks in the output.

Apparently the two structures behave differently in the time-varying case, even though both have the same transfer function ω_c/s. We say that the two structures have the same transfer function but different *topology* (the latter term referring to the components used in the block diagram and the way they are connected to each other). As mentioned, the transfer function is applicable only to the time-invariant case. No wonder its possible to have structures with identical transfer functions, but different time-varying behavior.

Now, returning to the comparison of implementations in Fig. 3.21 and Fig. 3.12, we notice that the structure in Fig. 3.21 contains a gain element at the output, the value of this gain being approximately proportional to the cutoff (at low cutoffs). This will particularly produce unsmoothed jumps in the output in response to jumps in the cutoff value. In the structure in Fig. 3.12, on the other hand, the cutoff jumps will be smoothed by the integrator. Thus, the difference between the two structures is similar to the just discussed effect of the cutoff gain placement with the integrator.

We should conclude that, other things being equal, the structure in Fig. 3.21 is inferior to the one in Fig. 3.12 (or Fig. 3.18). In this respect consider that Fig. 3.12 is trying to explicitly emulate the analog integration behavior, *preserving the topology* of the original analog structure, while Fig. 3.21 is concerned solely with implementing a correct transfer function. Since Fig. 3.21 implements a classical approach to the bilinear transform application for digital filter design (which ignores the filter topology) we'll refer to the trapezoidal integration replacement technique as the *topology-preserving bilinear transform* (or, shortly, TP-

BLT). Or, even shorter, we can refer to this technique as simply the *topology-preserving transform* (TPT), implicitly assuming that the bilinear transform is being used.[12]

In principle, sometimes there are possibilities to "manually fix" the structures such as in Fig. 3.21. E.g. the time-varying performance of the latter is drastically improved by moving the b_0 gain to the input. The problem however is that this kind of fixing quickly gets more complicated (if being possible at all) with larger filter structures. On the other hand, the TPT method explicitly aims at emulating the time-varying behavior of the analog prototype structure, which aspect is completely ignored by the classical transform approach. Besides, if the structure contains nonlinearities, preserving the topology becomes absolutely critical, because otherwise these nonlinearites can not be placed in the digital model.[13] Also, the direct forms suffer from precision loss issues, the problem growing bigger with the order of the system. For that reason in practice the direct forms of orders higher than 2 are rarely used,[14] but even 2nd-

[12] Apparently, naive filter design techniques also preserve the topology, but they do a rather poor job on the transfer functions. Classical bilinear transform approach does a good job on the transfer function, but doesn't preserved the topology. The topology-preserving transform achieves both goals simultaneously.

[13] This is related to the fact that transfer functions can be defined only for linear time-invariant systems. Nonlinear cases are obviously not linear, thus some critical information can be lost, if the conversion is done solely based on the transfer functions.

[14] A higher-order transfer function is typically decomposed into a product of transfer functions of 1st- and 2nd-order rational functions (with real coefficients!). Then it can be implemented by a serial connection of the respective 1st- and 2nd-order direct form filters.

order direct forms could already noticeably suffer from precision losses.

3.12 Other replacement techniques

The trapezoidal integrator replacement technique can be seen as a particular case of a more general set of replacement techniques. Suppose we have two filters, whose frequency response functions are $F_1(\omega)$ and $F_2(\omega)$ respectively. The filters do not need to have the same nature, particularly one can be an analog filter while the other can be a digital one. Suppose further, there is a frequency axis mapping function $\omega' = \mu(\omega)$ such that

$$F_2(\omega) = F_1(\mu(\omega))$$

Typically $\mu(\omega)$ should map the entire domain of $F_2(\omega)$ onto the entire domain of $F_1(\omega)$ (however the exceptions are possible).

To make the subsequent discussion more intuitive, we will assume that $\mu(\omega)$ is monotone, although this is absolutely not a must.[15] In this case we could say that $F_2(\omega)$ is obtained from $F_1(\omega)$ by a frequency axis warping. Particularly, this is exactly what happens in the bilinear transform case (the mapping $\mu(\omega)$ is then defined by the equation (3.7)). One cool thing about

[15] Strictly speaking, we don't even care whether $\mu(\omega)$ is single-valued. We could have instead required that

$$F_2(\mu_2(\omega)) = F_1(\mu_1(\omega))$$

for some $\mu_1(\omega)$ and $\mu_2(\omega)$.

the frequency axis warping is that it preserves the relationship between the amplitude and phase.

Suppose that we have a structure built around filters of frequency response $F_1(\omega)$, and the rest of the structure doesn't contain any memory elements (such as integrators or unit delays). Then the frequency response $F(\omega)$ of this structure will be a function of $F_1(\omega)$:

$$F(\omega) = \Phi(F_1(\omega))$$

where the specifics of the function $\Phi(w)$ will be defined by the details of the container structure. E.g. if the building-block filters are analog integrators, then $F_1(\omega) = 1/j\omega$. For the filter in Fig. 2.2 we then have

$$\Phi(w) = \frac{w}{w+1}$$

Indeed, substituting $F_1(\omega)$ into $\Phi(w)$ we obtain

$$F(\omega) = \Phi(F_1(\omega)) = \Phi(1/j\omega) = \frac{1/j\omega}{1 + 1/j\omega} = \frac{1}{1 + j\omega}$$

which is the already familiar to us frequency response of the analog lowpass filter.

Now, we can view the trapezoidal integrator replacement as a substitution of F_2 instead of F_1, where $\mu(\omega)$ is obtained from (3.7):

$$\omega_a = \mu(\omega_d) = \frac{2}{T}\tan\frac{\omega_d T}{2}$$

The frequency response of the resulting filter is obviously equal to $\Phi(F_2(\omega))$, where $F_2(\omega)$ is the frequency response of the trapezoidal integrators (used in place of analog ones). But since $F_2(\omega) = F_1(\mu(\omega))$.

$$\Phi(F_2(\omega)) = \Phi(F_1(\mu(\omega)))$$

which means that the frequency response $\Phi(F_2(\cdot))$ of the structure with trapezoidal integrators is obtained from the frequency response $\Phi(F_1(\cdot))$ of the structure with analog integrators simply by warping the frequency axis. If the warping is not too strong, the frequency responses will be very close to each other. This is exactly what is happening in the trapezoidal integrator replacement and generally in the bilinear transform.

Differentiator-based filters

We could have used some other two filters, with their respective frequency responses F_1 and F_2. E.g. we could consider continuous-time systems built around differentiators rather than integrators.[16] The transfer function of a differentiator is apparently simply $H(s) = s$, so we could use (3.5) to build a discrete-time "trapezoidal differentiator". Particularly, if we use the direct form II approach, it could look similarly to the integrator in Fig. 3.9. When embedding the cutoff control into a differentiator (in the form of a $1/\omega_c$ gain), it's probably bet-

[16] The real-world analog electronic circuits are "built around" integrators rather than differentiators. Therefore the differentiator-based filters have rather theoretical significance in the VA filter design.

ter to position it after the differentiator, to avoid the unnecessary "de-smoothing" of the control modulation by the differentiator. Replacing the analog differentiators in a structure by such digital trapezoidal differentiators we effectively perform a differentiator-based TPT.

E.g. if we replace the integrator in the highpass filter in Fig. 2.8 by a differentiator, we essentially perform a $1/s \leftarrow s$ substitution, thus we should have obtained a (differentiator-based) lowpass filter. Remarkably, if we perform a differentiator-based TPT on such filter, the obtained digital structure is fully equivalent to the previously obtained integrator-based TPT 1-pole lowpass filter.

Allpass substitution

One particularly interesting case occurs when F_1 and F_2 define two different allpass frequency responses. That is $|F_1(\omega)| \equiv 1$ and $|F_2(\omega)| \equiv 1$. In this case the mapping $\mu(\omega)$ is always possible. Especially since the allpass responses (defined by rational transfer functions of analog and digital systems) always cover the entire phase range from $-\pi$ to π.[17] In intuitive terms it means: for a filter built of identical allpass elements, we can always replace those allpass elements with an arbitrary other type of allpass elements (provided all other elements are memoryless, that is there are only gains and summators). We will refer to this process as *allpass substitution.* Whereas in the trapezoidal integrator replacement we have replaced analog integrators by

[17] Actually, for $-\infty < \omega < +\infty$, they cover this range exactly N times, where N is the order of the filter.

digital trapezoidal integrators, in the allpass substitution we replace allpass filters of one type by allpass filters of another type.

We can even replace digital allpass filters with analog ones and vice versa. E.g., noticing that z^{-1} elements *are* allpass filters, we could replace them with analog allpass filters. One particularly interesting case arises out of the inverse bilinear transform (3.10). From (3.10) we obtain

$$z^{-1} = \frac{1 - \frac{sT}{2}}{1 + \frac{sT}{2}} \tag{3.20}$$

The right-hand side of (3.20) obviously defines a stable 1-pole allpass filter, whose cutoff is $2/T$. We could take a digital filter and replace all z^{-1} elements with an analog allpass filter structure implementing (3.20). By doing this we would have performed a topology-preserving inverse bilinear transform.

We could then apply the cutoff parametrization to these underlying analog allpass elements:

$$\frac{sT}{2} \leftarrow \frac{s}{\omega_c}$$

so that we obtain

$$z^{-1} = \frac{1 - s/\omega_c}{1 + s/\omega_c}$$

The expression s/ω_c can be also rewritten as $sT/2\alpha$, where α is the cutoff scaling factor:

$$z^{-1} = \frac{1 - sT/2\alpha}{1 + sT/2\alpha} \tag{3.21}$$

Finally, we can apply the trapezoidal integrator replacement to the cutoff-scaled analog filter, converting it back to the digital domain. By doing so, we have applied the cutoff scaling in the digital domain! On the transfer function level this is equivalent to applying the bilinear transform to (3.21), resulting in

$$\begin{aligned} z^{-1} &= \frac{1 - sT/2\alpha}{1 + sT/2\alpha} \leftarrow \frac{1 - \frac{z-1}{\alpha(z+1)}}{1 + \frac{z-1}{\alpha(z+1)}} = \\ &= \frac{\alpha(z+1) - (z-1)}{\alpha(z+1) + (z-1)} = \frac{(\alpha-1)z + (\alpha+1)}{(\alpha+1)z + (\alpha-1)} \end{aligned}$$

That is, we have obtained a discrete-time allpass substitution

$$z^{-1} \leftarrow \frac{(\alpha-1)z + (\alpha+1)}{(\alpha+1)z + (\alpha-1)}$$

which applies cutoff scaling in the digital domain.[18] The allpass filter

$$H(z) = \frac{(\alpha-1)z + (\alpha+1)}{(\alpha+1)z + (\alpha-1)}$$

should have been obtained, as described, by the trapezoidal integrator replacement in an analog implementation of (3.21), alternatively we could use a direct form implementation. Notice that this filter has a pole at $z = (\alpha-1)/(\alpha+1)$. Since $|\alpha-1| < |\alpha+1|$ $\forall\alpha > 0$, the pole is always located inside the unit circle, and the filter is always stable.

[18] Differently from the analog domain, the digital cutoff scaling doesn't exactly shift the response along the frequency axis in a logarithmic scale, as some frequency axis warping is involved. The resulting frequency response change however is pretty well approximated as shiting in the lower frequency range.

3.13 Instantaneously unstable feedback

Writing the solution (3.13) for the zero-delay feedback equation (3.12) we in fact have slightly jumped the gun. Why? Let's consider once again the structure in Fig. 3.17 and suppose g gets negative and starts growing in magnitude further in the negative direction.[19] When g becomes equal to -1, the denominator of (3.13) turns into zero. Something bad must be happening at this moment.

In order to understand the meaning of this situation, let's consider the delayless feedback path as if it was an analog feedback. An analog signal value can't change instantly. It can change very quickly, but not instantly, it's always a continuous function of time. We could imagine there is a smoother unit somewhere in the feedback path (Fig. 3.22). This smoother unit has a very very fast response time. We introduce the notation $\bar{y}$ for the output of the smoother.

So, suppose we wish to compute a new output sample $y[n]$ for the new input sample $x[n]$. At the time $x[n]$ "arrives" at the filter's input, the smoother still holds the old output value $y[n-1]$. Let's freeze the discrete time at this point (which formally means we simply are not going to update the internal state of the z^{-1} element). At the same time we will let the continuous time t run, formally starting at $t = 0$ at the discrete time moment n.

[19] Of course, such lowpass filter formally has a negative cutoff value. It is also unstable. However unstable circuits are very important as the linear basis for the analysis and implementation of e.g. nonlinear self-oscillating filters. Therefore we wish to be able to handle unstable circuits as well.

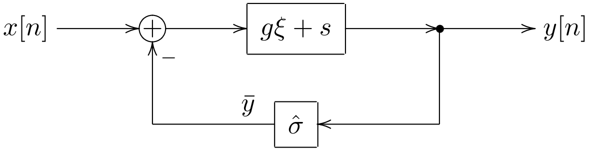


Figure 3.22: Digital 1-pole lowpass filter with a trapezoidal integrator in the instantaneous response form and a smoother unit $\hat{\sigma}$ in the delayless feedback path.

In this time-frozen setup we can choose arbitrary units for the continuous time t. The smoother equation can be written as

$$\operatorname{sgn}\dot{\bar{y}}(t) = \operatorname{sgn}\bigl(y(t) - \bar{y}(t)\bigr)$$

That is, we don't specify the details of the smoothing behavior, however the smoother output always changes in the direction from $\bar{y}$ towards y at some (not necessarily constant) speed.[20] Particularly, we can simply define a constant speed smoother:

$$\dot{\bar{y}} = \operatorname{sgn}(y - \bar{y})$$

or we could use a 1-pole lowpass filter as a smoother:

$$\dot{\bar{y}} = y - \bar{y}$$

[20] We also assume that the smoothing speed is sufficiently large to ensure that the smoothing process will converge at all cases where it potentially can converge (this statement should become clearer as we discuss more details).

The initial value of the smoother is apparently $\bar{y}(0) = y[n-1]$.

Now consider that

$$\begin{aligned}\operatorname{sgn}\dot{\bar{y}}(t) &= \operatorname{sgn}\bigl(y(t) - \bar{y}(t)\bigr) = \operatorname{sgn}\bigl(g(x[n] - \bar{y}(t)) + s - \bar{y}(t)\bigr) = \\ &= \operatorname{sgn}\bigl((gx[n] + s) - (1+g)\bar{y}(t)\bigr) = \operatorname{sgn}\bigl(a - (1+g)\bar{y}(t)\bigr)\end{aligned}$$

where $a = gx[n] + s$ is constant in respect to t. First, assume $1 + g > 0$. Further, suppose $a - (1+g)\bar{y}(0) > 0$. Then $\dot{\bar{y}}(0) > 0$ and then the value of the expression $a - (1+g)\bar{y}(t)$ will start decreasing until it turns to zero at some t, at which point the smoothing process converges. On the other hand, if $a - (1 + g)\bar{y}(0) < 0$, then $\dot{\bar{y}}(0) < 0$ and the value of the expression $a - (1+g)\bar{y}(t)$ will start increasing until it turns to zero at some t, at which point the smoothing process converges. If $a-(1+g)\bar{y}(0) = 0$ then the smoothing is already in a stable equilibrium state.

So, in case $1 + g > 0$ the instantaneous feedback smoothing process always converges. Now assume $1 + g \le 0$. Further, suppose $a-(1+g)\bar{y}(0) > 0$. Then $\dot{\bar{y}}(0) > 0$ and then the value of the expression $a - (1 + g)\bar{y}(t)$ will start further increasing (or stay constant if $1+g = 0$). Thus, $\bar{y}(t)$ will grow indefinitely. Respectively, if $a - (1+g)\bar{y}(0) < 0$, then $\bar{y}(t)$ will decrease indefinitely. This indefinite growth/decrease will occur within the frozen discrete time. Therefore we can say that $\bar{y}$ grows infinitely in an instant. We can refer to this as to an *instantaneously unstable* zero-delay feedback loop.

The analysis of the instantaneous stability can also be done using the analog filter stability analysis means. Let the smoother be an analog 1-pole lowpass filter with a unit cutoff (whose transfer function is $\frac{1}{s+1}$)[21] and notice that in that case the structure in Fig. 3.22 can be redrawn as in Fig. 3.23. This filter has two formal inputs $x[n]$ and s and one output $y[n]$.

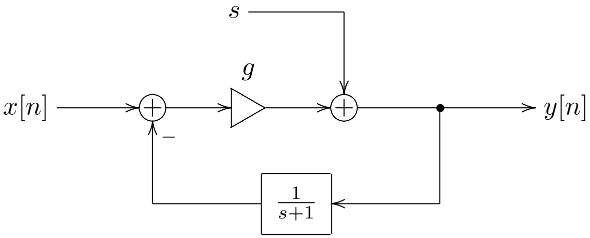


Figure 3.23: An instantaneous representation of a digital 1-pole lowpass filter with a trapezoidal integrator and an analog lowpass smoother.

We can now e.g. obtain a transfer function from the $x[n]$ input to the $y[n]$ output. Ignoring the s input signal (assuming it to be zero), for a continuous-time complex exponential input signal arriving at the $x[n]$ input, which we denote as $x[n](t)$, we have a respective continuous-time complex exponential signal at the $y[n]$ output, which we denote as $y[n](t)$:

$$y[n](t) = g\left(x[n](t) - \frac{1}{s+1}y[n](t)\right)$$

[21] Apparently, the variable s used in the transfer function $\frac{1}{s+1}$ is a different s than the one used in the instantaneous response expression for the integrator. The author apologizes for the slight confusion.

from where

$$y[n](t) = \frac{g}{1+g\frac{1}{s+1}}x[n](t)$$

that is

$$H(s) = \frac{g}{1+g\frac{1}{s+1}} = g\frac{s+1}{s+(1+g)}$$

This transfer function has a pole at $s = -(1+g)$. Therefore, the structure is stable if $1+g > 0$ and not stable otherwise.

The same transfer function analysis could have been done between the s input and the $y[n]$ output, in which case we would have obtained

$$H(s) = \frac{s+1}{s+(1+g)}$$

The poles of this transfer function however, are exactly the same, so it doesn't matter.[22]

Alright, so we have found out that the filter in Fig. 3.12 is instantaneously unstable if $g \leq -1$, but what can we do about it? Firstly, the problem typically doesn't occur, as normally $g > 0$ (not only in the 1-pole lowpass filter case, but also in other cases). Even if it can occur in principle, one can consider, whether these extreme parameter settings are so necessary to support, and possibly simply clip the filter parameters in such a way that the instantaneous instability doesn't occur.

[22] This is a common rule: the poles of a system with multiple inputs and/or multiple outputs are always the same regardless of the particular input-output pair for which the transfer function is being considered (exceptions in signular cases, arising out of pole/zero cancellation are possible, though).

Secondly, let's notice that $g = \omega_c T/2$. Therefore another solution could be to increase the sampling rate (and respectively reduce the sampling period T).[23]

Unstable bilinear transform

There is yet another idea, which hasn't been tried out in practice yet.[24] We are going to discuss it anyway.

The instantaneous instability is occurring at the moment when one of the analog filter's poles hits the pole of the inverse bilinear transform (3.10), which is located at $s = 2/T$. On the other hand, recall that the bilinear transform is mapping the imaginary axis to the unit circle, thus kind-of preserving the frequency response. If the system is not stable, then the frequency response doesn't make sense. Formally, the reason for

[23] Actually, the instantaneous instability has to do with the fact that the trapezoidal integration is not capable of producing reasonable approximation of the continuous-time integration, due to too extreme parameter values. Increasing the sampling rate obviously increases the precision of the trapezoidal integration as well.

The same idea can also be used to easily and reliably find out, whether the positive value of the denominator of the feedback equation's solution corresponds to the instantaneously stable case or vice versa. The sign which the denominator has for $T \to 0$ corresponds to the instantaneously stable case.

[24] The author just got the idea while writing the book and didn't find the time yet to properly experiment with it. Sufficient theoretical analysis is not possible here due to the fact that practical applications of instantaneously unstable (or any unstable, for that matter) filters occur typically for nonlinear filters, and there's not much theoretical analysis means for the latter. Hopefully there are no mistakes in the theoretical transformations, but even if there are mistakes, at least the idea itself could maybe work.

this is that the inverse Laplace transform of transfer functions only converges for $\sigma > \max\{\operatorname{Re} p_n\}$ where p_n are the poles of the transfer function, and respectively, if $\max\{\operatorname{Re} p_n\} \geq 0$, it doesn't converge on the imaginary axis ($\sigma = 0$). However, instead of the imaginary axis $\operatorname{Re} s = \sigma = 0$, let's choose some other axis $\operatorname{Re} s = \sigma > \max\{\operatorname{Re} p_n\}$ and use it instead of the imaginary axis to compute the "frequency response".

We also need to find a discrete-time counterpart for $\operatorname{Re} s = \sigma$. Considering that $\operatorname{Re} s$ defines the magnitude growth speed of the exponentials e^{st} we could choose a z-plane circle, on which the magnitude growth speed of z^n is the same as for $e^{\sigma t}$. Apparently, this circle is $|z| = e^{\sigma T}$. So, we need to map $\operatorname{Re} s = \sigma$ to $|z| = e^{\sigma T}$. Considering the bilinear transform equation (3.5), we divide z by $e^{\sigma T}$ to make sure $ze^{-\sigma T}$ has a unit magnitude and shift the s-plane result by σ:

$$s = \sigma + \frac{2}{T} \cdot \frac{ze^{-\sigma T} - 1}{ze^{-\sigma T} + 1} \tag{3.22}$$

We can refer to (3.22) as the *unstable bilinear transform*, where the word "unstable" refers not to the instability of the transform itself, but rather to the fact that it is designed to be applied to unstable filters.[25] Notice that at $\sigma = 0$ the unstable bilinear transform turns into an ordinary bilinear transform. The inverse transform is obtained by

$$\frac{(s-\sigma)T}{2}(ze^{-\sigma T} + 1) = ze^{-\sigma T} - 1$$

[25] Apparently, the unstable bilinear transform defines the same relationship between $\operatorname{Im} s$ and $\arg z$ as the ordinary bilinear transform. Therefore the standard prewarping formula applies.

from where

$$ze^{-\sigma T}\left(1-\frac{(s-\sigma)T}{2}\right)=1+\frac{(s-\sigma)T}{2}$$

and

$$z=e^{\sigma T}\frac{1+\frac{(s-\sigma)T}{2}}{1-\frac{(s-\sigma)T}{2}}\tag{3.23}$$

Apparently the inverse unstable bilinear transform (3.23) has a pole at $s=\sigma+\frac{2}{T}$. In order to avoid hitting that pole by the poles of the filter's transfer function (or maybe even generally avoid the real parts of the poles to go past that value) we could e.g. simply let

$$\sigma=\max\left\{0,\ \operatorname{Re}p_n\right\}$$

or we could position σ midways:

$$\sigma=\max\left\{0,\ \operatorname{Re}p_n-\frac{1}{T}\right\}$$

In order to construct an integrator defined by (3.22) we first need to obtain the expression for $1/s$ from (3.22):

$$\begin{aligned}\frac{1}{s}&=\frac{1}{\sigma+\frac{2}{T}\cdot\frac{ze^{-\sigma T}-1}{ze^{-\sigma T}+1}}=T\frac{ze^{-\sigma T}+1}{\sigma T(ze^{-\sigma T}+1)+2(ze^{-\sigma T}-1)}=\\&=T\frac{ze^{-\sigma T}+1}{(\sigma T+2)e^{-\sigma T}z+(\sigma T-2)}=\\&=T\frac{1+e^{\sigma T}z^{-1}}{(\sigma T+2)-(2-\sigma T)e^{\sigma T}z^{-1}}=\end{aligned}$$

$$= \frac{T}{2+\sigma T} \cdot \frac{1+e^{\sigma T}z^{-1}}{1-\frac{2-\sigma T}{2+\sigma T}e^{\sigma T}z^{-1}}$$

That is

$$\frac{1}{s} = \frac{T}{2+\sigma T} \cdot \frac{1+e^{\sigma T}z^{-1}}{1-\frac{2-\sigma T}{2+\sigma T}e^{\sigma T}z^{-1}} \tag{3.24}$$

A discrete-time structure implementing (3.24) could be e.g. the one in Fig. 3.24. Yet another approach could be to convert the right-hand side of (3.24) to the analog domain by the inverse bilinear transform, construct an analog implementation of the resulting transfer function and apply the trapezoidal integrator replacement to convert back to the digital domain. It is questionable, whether this produces better (or even different) results than Fig. 3.24.

SUMMARY

We have considered three essentially different approaches to applying time-discretization to analog filter models: naive, TPT (by trapezoidal integrator replacement), and the classical bilinear transform (using direct forms). The TPT approach combines the best features of the naive implementation and the classical bilinear transform.

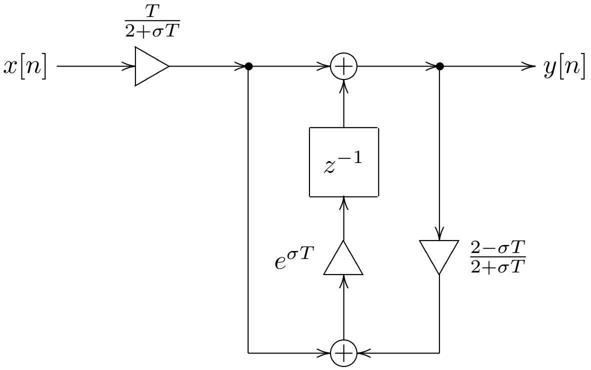


Figure 3.24: Transposed direct form II-style "unstable" trapezoidal integrator.

Chapter 4

Ladder filter

In this chapter we are going to discuss the most classical analog filter model: the transistor ladder filter. We will also discuss to an extent the diode ladder version.

4.1 Linear analog model

The analog transistor ladder filter is an essentially nonlinear structure. However, as the first approximation (and actually a quite good one) we will use its linearized model (Fig. 4.1). The LP_1 blocks denote four identical (same cutoff) 1-pole lowpass filters (Fig. 2.2). The k coefficient controls the amount of negative feedback, which affects the filter resonance. Typically

$k \geq 0$, although $k < 0$ is also sometimes used.[1]

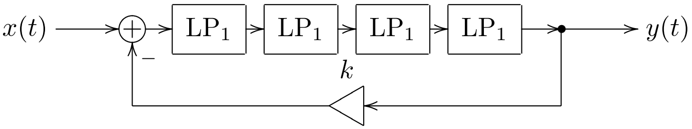


Figure 4.1: Transistor ladder filter.

Let

$$H_1(s) = \frac{1}{1+s}$$

be the 1-pole lowpass transfer function. Assuming complex exponential x and y we write

$$y = H_1^4(s) \cdot (x - ky)$$

from where

$$y(1 + kH_1^4(s)) = H_1^4(s) \cdot x$$

[1] The reason for the negative (rather than positive) feedback is actually quite intuitively simple. Considering the phase response of four serially connected 1-pole lowpass filters at the cutoff:

$$\left.\left(\frac{1}{1+s}\right)^4\right|_{s=j} = \frac{1}{(1+j)^4} = -\frac{1}{4}$$

we notice that the signal phase at the cutoff is inverted. Therefore we have to invert it once again in the feedback to achieve the resonance effect.

and the transfer function of the ladder filter is

$$H(s) = \frac{y}{x} = \frac{H_1^4(s)}{1 + kH_1^4(s)} = \frac{\frac{1}{(1+s)^4}}{1 + k\frac{1}{(1+s)^4}} = \frac{1}{k + (1+s)^4} \qquad (4.1)$$

At $k = 0$ the filter behaves as 4 serially connected 1-pole lowpass filters.

The poles of the filter are respectively

$$s = -1 + (-k)^{1/4}$$

where the raising to the 1/4th power is understood in the complex sense, therefore giving 4 different values:

$$s = -1 + \frac{\pm 1 \pm j}{\sqrt{2}} k^{1/4} \qquad (k \geq 0)$$

(this time $k^{1/4}$ is understood in the real sense). Therefore, at $k = 0$ all poles are located at $s = -1$, as k grows they move apart in 4 straight lines (all going at "45° angles"). As k grows from 0 to 4 the two of the poles (at $s = -1 + \frac{1 \pm j}{\sqrt{2}} k^{1/4}$) are moving towards the imaginary axis, producing a resonance peak in the amplitude response (Fig. 4.2). At $k = 4$ they hit the imaginary axis:

$$\operatorname{Re}\left(-1 + \frac{1 \pm j}{\sqrt{2}} 4^{1/4}\right) = 0$$

and the filter becomes unstable.

In Fig. 4.2 one could notice that, as the resonance increases, the filter gain at low frequencies begins to drop. Indeed, substituting $s = 0$ into (4.1) we obtain

$$H(0) = \frac{1}{1 + k}$$

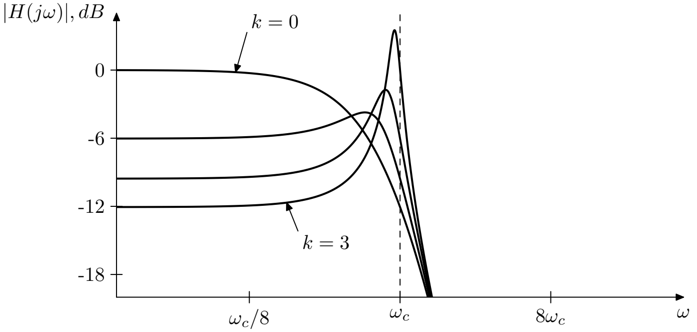


Figure 4.2: Amplitude response of the ladder filter for various k.

This is a general issue with ladder filter designs.

4.2 Linear digital model

A naive digital implementation of the ladder filter shouldn't pose any problems. We will therefore immediately skip to the TPT approach.

Recalling the instantaneous response of a single 1-pole lowpass filter (3.14), we can construct the instantaneous response of a serial connection of four of such filters. Indeed, let's denote the instantaneous responses of the respective 1-poles as $f_n(\xi) = g\xi + s_n$ (obviously, the coefficient g is identical for all four, whereas s_n depends on the filter state and therefore cannot be assumed identical). Combining two such filters in series we have

$$f_2(f_1(\xi)) = g(g\xi + s_1) + s_2 = g^2\xi + gs_1 + s_2$$

Adding the third one:

$$f_3(f_2(f_1(\xi))) = g(g^2\xi + gs_1 + s_2) + s_3 = g^3\xi + g^2 s_1 + gs_2 + s_3$$

and the fourth one:

$$\begin{aligned} f_4(f_3(f_2(f_1(\xi)))) &= g(g^3\xi + g^2 s_1 + gs_2 + s_3) = \\ &= g^4\xi + g^3 s_1 + g^2 s_2 + gs_3 + s_4 = G\xi + S \end{aligned}$$

where

$$\begin{aligned} G &= g^4 \\ S &= g^3 s_1 + g^2 s_2 + gs_3 + s_4 \end{aligned}$$

Using the obtained instantaneous response $G\xi+S$ of the series of 4 1-poles, we can redraw the ladder filter structure as in Fig. 4.3.

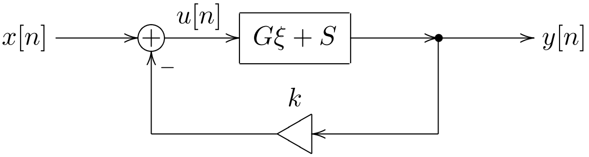


Figure 4.3: TPT ladder filter in the instantaneous response form.

Rather than solving for y, let's solve for the signal u at the feedback point. From Fig. 4.3 we obtain

$$u = x - ky = x - k(Gu + S)$$

from where

$$u = \frac{x - kS}{1 + kG} \tag{4.2}$$

We can then use the obtained value of u to process the 1-pole lowpasses one after the other, updating their state, and computing $y[n]$ as the output of the fourth lowpass.

Notice that for positive cutoff values of the underlying 1-pole lowpasses we have $g > 0$. Respectively $G = g^4 > 0$. This means that for $k \geq 0$ the denominator of (4.2) is always positive and never turns to zero, so we should be safe regarding the instantaneously unstable feedback.[2]

For $k < 0$ the situation is however different. Since $0 < g < 1$ (for $\omega_c > 0$), it follows that $0 < G < 1$. Thus $1 + kG > 0$ $\forall k \geq -1$, however at $k < -1$ we can get into an instantaneously unstable feedback case.

[2] Strictly speaking, we should have checked the instantaneous stability using the feedback smoother approach. However typically a positive denominator of the form $1 + g$ or $1 + kG$ implies that everything is fine.

A quicker way to check for the instantaneous feedback would be to let the sampling rate infinitely grow ($T \to 0$) and then check, whether the denominator changes its sign along the way. In our case $G = g^4 = (\omega_c T/2)^4$, which means the denominator is always larger than 1 (under the assumption $k \geq 0$), regardless of T.

4.3 Feedback shaping

We have observed that at high resonance settings the amplitude gain of the filter at low frequencies drops (Fig. 4.2). An obvious way to fix this problem would be e.g. to boost the input signal by the $(1+k)$ factor.[3] However there's another way to address the same issue. We could "kill" the resonance at the low frequencies by introducing a highpass filter in the feedback (Fig. 4.4). In the simplest case this could be a 1-pole highpass.

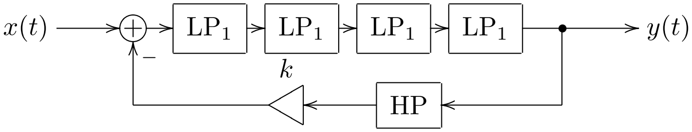


Figure 4.4: Transistor ladder filter with a highpass in the feedback.

The cutoff of the highpass filter can be static or vary along with the cutoff of the lowpasses. The static version has a nice feature that it kills the resonance effect at low frequencies regardless of the master cutoff setting, which may be desirable if the resonance at low frequencies is considered rather unpleasant (Fig. 4.5).

In principle one can also use other filter types in the feedback

[3] We boost the input rather than the output signal for the same reason as when preferring to place the cutoff gains in front of the integrators.

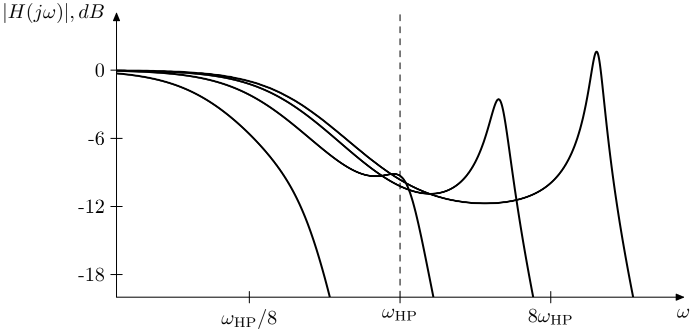


Figure 4.5: Amplitude response of the ladder filter with a static-cutoff highpass in the feedback for various lowpass cutoffs.

shaping. One has to be careful though, since this changes the positions of the filter poles. Particularly, inserting a lowpass into the feedback can easily destabilize an otherwise stable filter.

4.4 Multimode ladder filter

Warning! *The multimode functionality of the ladder filter is a rather exotic feature. If you're looking for the bread-and-butter bandpass, highpass, notch etc. filters, you should first take a look at the multimode 2-pole state-variable filter discussed later in the book.*

By picking up intermediate signals of the ladder filter as in

Fig. 4.6 we obtain the multimode version of this filter. We then can use linear combinations of signals y_n to produce various kinds of filtered signal.[4]

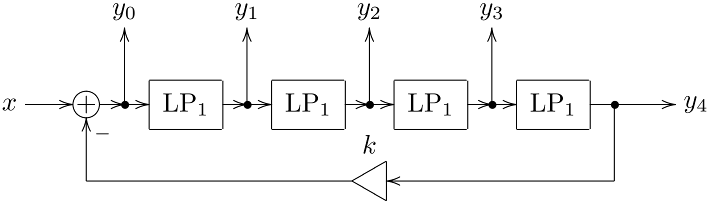


Figure 4.6: Multimode ladder filter.

Suppose $k = 0$. Apparently, in this case, the respective transfer functions associated with each of the y_n outputs are

$$H_n(s) = \frac{1}{(1+s)^n} \qquad (n = 0, \ldots, 4)$$

If $k \neq 0$ then from

$$H_4(s) = \frac{1}{k + (1+s)^4}$$

[4] Actually, instead of y_0 we could have used the input signal x for these linear combinations. However, it doesn't matter. Since $y_0 = x - ky_4$, we can express x via y_0 or vice versa. It's just that some useful linear combinations have simpler (independent of k) coefficients if y_0 rather than x is being used.

using the obvious relationship $H_{n+1}(s) = H_n(s)/(s+1)$ we obtain

$$H_n(s) = \frac{(1+s)^{4-n}}{k+(1+s)^4}$$

Let's say we want to build a 4th-order highpass filter. Considering that the 4th order lowpass transfer function (under the assumption $k = 0$) is built as a product of four 1st order lowpass transfer functions $1/(1+s)$

$$H_{\text{LP}}(s) = \frac{1}{(1+s)^4}$$

we might decide to build the 4th order highpass transfer function as a product of four 1st order highpass transfer functions $s/(1+s)$:

$$H_{\text{HP}}(s) = \frac{s^4}{(1+s)^4}$$

Let's attempt to build $H_{\text{HP}}(s)$ as a linear combination of $H_n(s)$. Apparently, a linear combination of $H_n(s)$ must have the denominator $k+(1+s)^4$, so let's instead construct

$$H_{\text{HP}}(s) = \frac{s^4}{k+(1+s)^4} \tag{4.3}$$

which at $k = 0$ will turn into $s^4/(1+s)^4$:

$$\frac{s^4}{k+(1+s)^4} =$$

$$= \frac{a_0(1+s)^4 + a_1(1+s)^3 + a_2(1+s)^2 + a_3(1+s) + a_4}{k+(1+s)^4} =$$

$$
\begin{aligned}
&= \frac{1}{(1+s)^4}\bigl(a_0s^4 + (a_1+4a_0)s^3 + (a_2+3a_1+6a_0)s^2 + \\
&\qquad + (a_3+2a_2+3a_1+4a_0)s + (a_0+a_1+a_2+a_3+a_4)\bigr)
\end{aligned}
$$

from where

$$
\begin{aligned}
&a_0 = 1 \\
&a_1 + 4a_0 = 0 \\
&a_2 + 3a_1 + 6a_0 = 0 \\
&a_3 + 2a_2 + 3a_1 + 4a_0 = 0 \\
&a_0 + a_1 + a_2 + a_3 + a_4 = 0
\end{aligned}
$$

from where $a_0 = 1$, $a_1 = -4$, $a_2 = 6$, $a_3 = -4$, $a_4 = 1$. The amplitude response corresponding to (4.3) is plotted in Fig. 4.7.[5]

A bandpass filter can be built as

$$H_{\text{BP}}(s) = \frac{s^2}{k + (1+s)^4} \tag{4.4}$$

The two zeros at $s = 0$ will provide for a -12dB/oct rolloff at low frequencies and will reduce the -24dB/oct rolloff at high frequencies to the same -12dB/oct. Notice that the phase response at the cutoff is zero:

$$H_{\text{BP}}(j) = \frac{-1}{k + (1+j)^4} = \frac{1}{4-k}$$

[5] We could have also constructed a true ladder highpass filter by using four 1-pole highpass filters instead of four 1-pole lowpass filters as the fundamental blocks of the ladder filter.

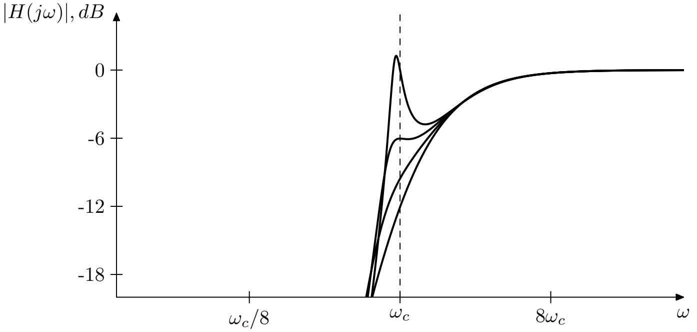


Figure 4.7: Amplitude response of the highpass mode of the ladder filter for various k.

Finding the coefficients is left as an exercise for the reader. The amplitude response corresponding to (4.3) is plotted in Fig. 4.8.[6]

Further filter types can be built in a similar way.

4.5 Simple nonlinear model

At $k \geq 4$ the ladder filter becomes unstable, as the resonance becomes too strong. We could however prevent the signal level

[6] We could have also constructed a true ladder bandpass filter by using two 2-pole bandpass filters (e.g. SVF bandpasses, or serial combinations of a 1-pole lowpass and a 1-pole highpass at the same cutoff) instead of four 1-pole lowpass filters as the fundamental blocks of the ladder filter. Notice that in the bandpass case, we need a positive rather than negative (inverted) feedback, since the phase shift at the bandpass' cutoff is zero.

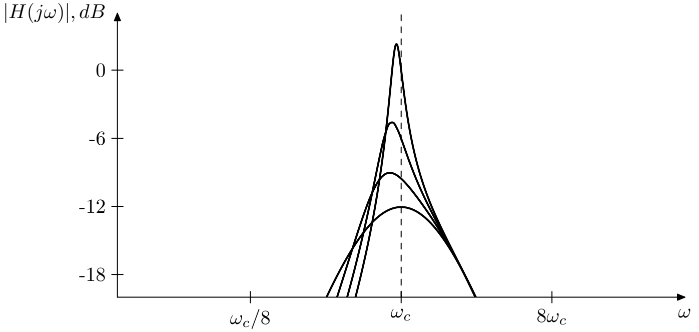


Figure 4.8: Amplitude response of the bandpass mode of the ladder filter for various k.

from growing infinitely by putting a saturator into the feedback path. This will allow the filter to go into *selfoscillation* at $k > 4$. The best place for such saturator is probably at the feedback point, since then it will process both the input and the feedback signals simultaneously, applying a nice overdrive-like saturation to the input signal. A hyperbolic tangent function should provide a nice saturator (Fig. 4.9). It is transparent at low signal levels, therefore at low signal levels the filter behaves as a linear one.

The introduction of the nonlinearity in the feedback path poses no problems for a naive digital model. In the TPT case however this complicates the things quite a bit. Consider Fig. 4.3 redrawn to contain the feedback nonlinearity (Fig. 4.10).

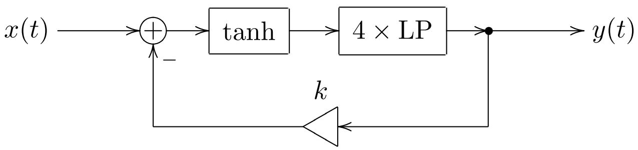


Figure 4.9: Transistor ladder filter with a saturator at the feedback point.

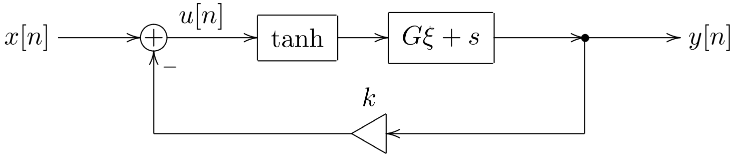


Figure 4.10: Nonlinear TPT ladder filter in the instantaneous response form.

Writing the zero-delay feedback equation we obtain

$$u = x - k(G \tanh u + s) \tag{4.5}$$

Apparently, the equation (4.5) is a transcendental one. It can be solved only using numerical methods. Also, a linear zero-delay feedback equation had only one solution, but how many solutions does (4.5) have? In order to answer the latter question, let's rewrite (4.5) as

$$(x - ks) - u = kG \tanh u \tag{4.6}$$

If $k \geq 0$ and $G > 0$, then the right-hand side of (4.6) is a nonstrictly increasing function of u, while the left-hand side of (4.6) is a strictly decreasing function of u. Thus, (4.6) and respectively (4.5) have a single solution in this case. If $k < 0$, (4.5) can have multiple solutions (up to three). One could use the smoother paradigm introduced in the instantaneously unstable feedback discussion to find out the applicable one.

It is possible to avoid the need of solving the transcendental equation by using a saturator function which still allows analytic solution. This is particularly the case with second-order curves, such as hyperbolas. E.g. one could consider a saturator function $f(x) = x/(1 + |x|)$ consisting of two hyperbolic segments, which turns (4.6) into

$$(x - ks) - u = kG\frac{u}{1 + |u|} \tag{4.7}$$

In order to solve (4.7) one first has to check whether the solution occurs at $u > 0$ or $u < 0$, which can be done by simply evaluating the left-hand side of (4.7) at $u = 0$. Then one can replace $|u|$ by u or $-u$ respectively and solve the resulting quadratic equation.[7]

[7] The same kind of quadratic equation appears for any other second-order curve (hyperbola, ellipse, parabola, including their rotated versions). In solving the quadratic equation one has not only to choose the right one of the two roots of the equation. One also has to choose the right one of

Yet another approach (which also works for multiple nonlinearities!) is to first solve the feedback equation for the linear case, and then apply the nonlinearites "on top". E.g. we use the structure in Fig. 4.3 to obtain the value of u. However then we pretend that we have found the value of u for Fig. 4.10 (or Fig. 4.9, for that matter) and proceed accordingly, putting u through the hyperbolic tangent shaper and then further through the 1-pole lowpasses. We refer to this approach as the "cheap method" of applying the nonlinearities to the TPT structures. It is intuitively clear, that the cheap method is more likely to produce "wrong" results at high cutoff values.

No matter, which approach we chose to compute nonlinearities, one shouldn't forget that nonlinear shaping introduces overtones (usually an infinite amount of those) into the signal, which in turn introduces aliasing. Meaning: the stronger are the nonlinearities in your structure, the more you might need to oversample. If the oversampling is extreme anyway, there might be little difference in quality between the naive and the TPT approach.[8]

the two solution formulas for the quadratic equation $Ax^2 - 2Bx + C = 0$:

$$x = \frac{B \pm \sqrt{B^2 - AC}}{A} \qquad \text{or} \qquad x = \frac{C}{B \mp \sqrt{B^2 - AC}}$$

(where the choice of the formula is based on the sign of B and on the choice of the "±" sign, corresponding to choosing between the two solutions). The reason to use these two different formulas is that they can become ill-conditioned, resulting in computation precision losses. More details on this approach are discussed in the author's article *"Computational optimization of nonlinear zero-delay feedback by second-order piecewise approximation"*.

[8] Before making a final decision, it might be worth asking a few mu-

4.6 Advanced nonlinear model

The nonlinearity introduced in Fig. 4.9 does a good job and sounds reasonably close to a hardware analog transistor ladder filter, however this is not how the nonlinearities in the hardware ladder filter "really" work. In order to describe a closer to reality nonlinear model of the ladder filter, we need to start by introducing nonlinearities into the underlying 1-pole lowpass filters (Fig. 4.11).

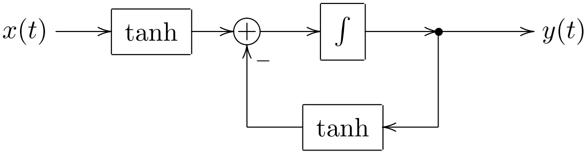


Figure 4.11: A nonlinear 1-pole lowpass element of the ladder filter.

So the equation (2.1) is transformed into

$$y = y(t_0) + \int_{t_0}^{t} \omega_c\bigl(\tanh x(\tau) - \tanh y(\tau)\bigr)\,\mathrm{d}t$$

Which effect does this have? Apparently, the presence of the tanh function reduces the absolute value of the difference $\tanh x-$

sicians with an ear for analog sound to perform a listening test, whether the differences between the naive and TPT models of a particular filter are inaudible and uncritical.

$\tanh y$, if the level of one or both of the signals is sufficiently high. If both x and y have large values of the same sign, it's possible that the difference $\tanh x - \tanh y$ is close to zero, even though the difference $x - y$ is very large. This means that the filter will update its state slower than in (2.1). Intuitively this feels a little bit like "cutoff reduction" at large signal levels.

We can then connect the 1-pole models from Fig. 4.11 into a series of four 1-poles and put a feedback around them, exactly the same way as in Fig. 4.1. Notice that when connecting Fig. 4.11 filters in series, one could use a common tanh module between each of them, thereby optimizing the computation (Fig. 4.12).[9]

One could further enhance the nonlinear behavior of the ladder filter model by putting another saturator (possibly of a different type, or simply differently scaled) into the feedback path.

4.7 Diode ladder

In the diode ladder filter the serial connection of four 1-pole lowpass filters (implemented by the transistor ladder) is replaced by a more complicated structure of 1-pole filters (implemented by the diode ladder). The equations of the diode ladder itself

[9] There is an issue which may appear when using simple tanh approximations having a fully horizontal saturation curve. If both the input and the output signals of a 1-pole are having large values of the same sign, the tanh approximations will return two identical values and the difference $\tanh x - \tanh y$ will be approximated by zero. This might result in the filter getting "stuck" (the cutoff effectively reduced to zero).

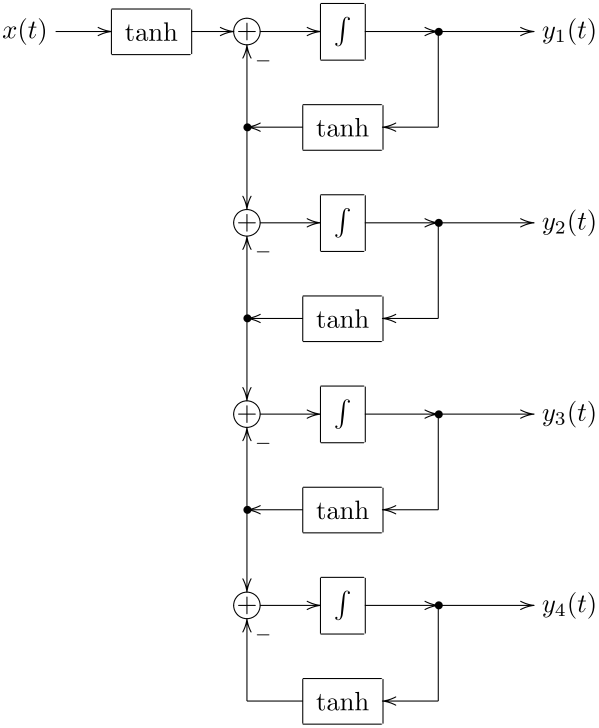


Figure 4.12: Advanced nonlinear transistor ladder (the main feedback path of the ladder filter not shown).

(without the feedback path) are

$$\begin{aligned}
\dot{y}_1 &= \omega_c\bigl(\tanh x - \tanh(y_1 - y_2)\bigr)\\
\dot{y}_2 &= \frac{\omega_c}{2}\bigl(\tanh(y_1 - y_2) - \tanh(y_2 - y_3)\bigr)\\
\dot{y}_3 &= \frac{\omega_c}{2}\bigl(\tanh(y_2 - y_3) - \tanh(y_3 - y_4)\bigr)\\
\dot{y}_4 &= \frac{\omega_c}{2}\bigl(\tanh(y_3 - y_4) - \tanh y_4\bigr)
\end{aligned}$$

which is implemented by the structure in Fig. 4.13 (compare to Fig. 4.12). The diode ladder itself is then built by providing the feedback path around the diode ladder, where the fourth output of the diode ladder is fed back into the diode ladder's input (Fig. 4.14).

The linearized form of the diode ladder equations is obtained by assuming $\tanh\xi \approx \xi$, resulting in

$$\begin{aligned}
\dot{y}_1 &= \omega_c\bigl((x + y_2) - y_1\bigr)\\
\dot{y}_2 &= \omega_c\bigl((y_1 + y_3)/2 - y_2\bigr)\\
\dot{y}_3 &= \omega_c\bigl((y_2 + y_4)/2 - y_2\bigr)\\
\dot{y}_4 &= \omega_c\bigl(y_3/2 - y_4\bigr)
\end{aligned}$$

which apparently is representable as a serial connection of four identical 1-pole lowpass filters (all having the same cutoff ω_c) with some extra gains and feedback paths (Fig. 4.15).

These more complicated connections between the lowpasses "destroy" the frequency response of the ladder in a remarkable form, responsible for the characteristic diode ladder filter

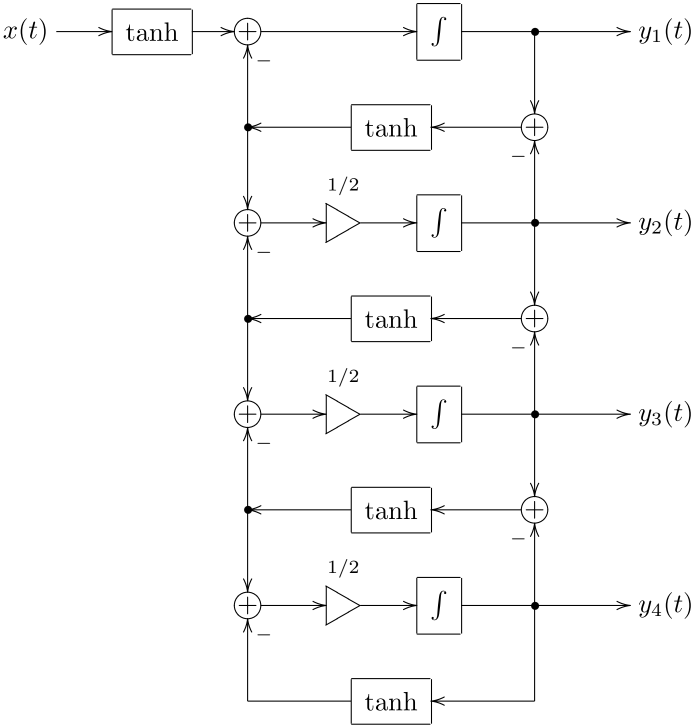


Figure 4.13: Diode ladder.

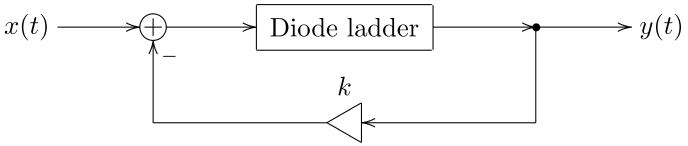


Figure 4.14: Diode ladder filter.

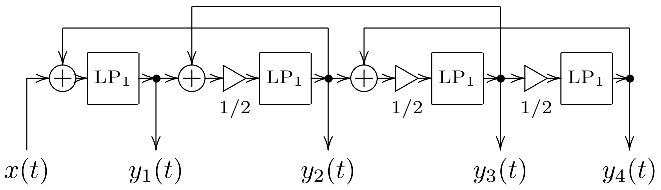


Figure 4.15: Linearized diode ladder.

sound. Generally, the behavior of the diode ladder filter is less “straightforward” than the one of the transistor ladder filter.

Transfer function

In order to obtain the transfer function of the structure in Fig. 4.15 let

$$G(s) = \frac{1}{1+s} \tag{4.8}$$

be the transfer function of the underlying lowpass filters. Let $F(s) = 2G(s)$, or, simplifying the notation, $F = 2G$. Assuming complex exponential signals e^{st}, we obtain from Fig. 4.15

$$y_4 = Fy_3$$

Respectively

$$y_3 = F \cdot (y_2 + y_4)$$

Multiplying both sides by F we have

$$Fy_3 = F^2 \cdot (y_2 + y_4)$$

from where, recalling that $y_4 = Fy_3$:

$$y_4 = F^2 \cdot (y_2 + y_4)$$

that is

$$y_4 = \frac{F^2}{1 - F^2} y_2$$

Further,

$$y_2 = F \cdot (y_1 + y_3) = Fy_1 + y_4$$

Multiplying both sides by $F^2/(1 - F^2)$:

$$y_4 = \frac{F^3}{1 - F^2} y_1 + \frac{F^2}{1 - F^2} y_4$$

from where

$$y_4 \left(1 - \frac{F^2}{1 - F^2}\right) = \frac{F^3}{1 - F^2} y_1$$

and respectively

$$y_4 = \frac{F^3}{1-2F^2} y_1$$

And finally,

$$y_1 = 2F \cdot (x + y_2)$$

Multiplying both sides by $F^3/(1-2F^2)$:

$$\begin{aligned} y_4 &= \frac{2F^4}{1-2F^2}\left(x + \frac{1-F^2}{F^2}\cdot\frac{F^2}{1-F^2}y_2\right) = \\ &= \frac{2F^4}{1-2F^2}\left(x + \frac{1-F^2}{F^2}y_4\right) = \\ &= \frac{2F^4}{1-2F^2}x + 2F^2\frac{1-F^2}{1-2F^2}y_4 \end{aligned}$$

from where

$$y_4\left(1 - 2F^2\frac{1-F^2}{1-2F^2}\right) = \frac{2F^4}{1-2F^2}x$$

from where

$$y_4\left(1 - 4F^2 + 2F^4\right) = 2F^4 x$$

and

$$y_4 = \frac{2F^4}{1-4F^2+2F^4}x = \frac{G^4/8}{1-G^2+G^4/8}x$$

That is, the transfer function $\Delta(s)$ of the diode ladder is

$$\Delta(s) = \frac{G^4/8}{G^4/8 - G^2 + 1} \qquad \text{where } G(s) = \frac{1}{1+s}$$

from where we obtain the transfer function of the entire diode ladder filter as

$$H(s) = \frac{\Delta}{1 + k\Delta}$$

The corresponding amplitude response is plotted in Fig. 4.16.

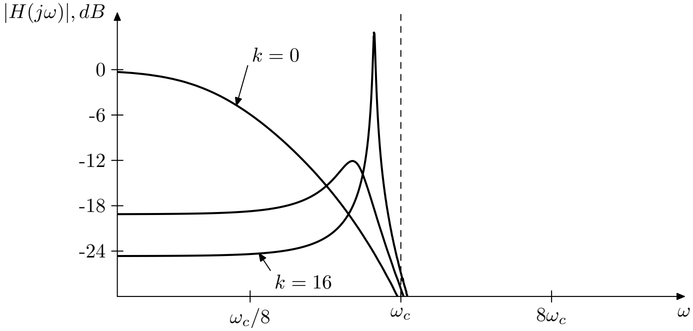


Figure 4.16: Amplitude response of the diode ladder filter for various k.

Let's find the positions of the poles of the diode ladder filter. Equating the denominator to zero:

$$1 + k\Delta = 0$$

we have

$$1 = -k\Delta = \frac{-kG^4/8}{G^4/8 - G^2 + 1}$$

that is

$$-k\frac{G^4}{8} = \frac{G^4}{8} - G^2 + 1$$

that is

$$\frac{1+k}{8}G^4 - G^2 + 1 = 0$$

Solving for G^2:

$$G^2 = \frac{1 \pm \sqrt{1 - \frac{1+k}{2}}}{\frac{1+k}{4}} = \frac{1 \pm \sqrt{\frac{1-k}{2}}}{\frac{1+k}{4}} \tag{4.9}$$

Equating (4.9) and the squared form of (4.8) we have

$$\frac{1}{(1+s)^2} = \frac{1 \pm \sqrt{\frac{1-k}{2}}}{\frac{1+k}{4}}$$

that is

$$(s+1)^2 = \frac{\frac{1+k}{4}}{1 \pm \sqrt{\frac{1-k}{2}}} = \frac{\frac{1+k}{4}\left(1 \mp \sqrt{\frac{1-k}{2}}\right)}{1 - \frac{1-k}{2}} =$$
$$= \frac{\frac{1+k}{4}\left(1 \mp \sqrt{\frac{1-k}{2}}\right)}{\frac{1+k}{2}} = \frac{1 \mp \sqrt{\frac{1-k}{2}}}{2}$$

that is

$$(s+1)^2 = \frac{1}{2} \pm \frac{1}{2}\sqrt{\frac{1-k}{2}} \tag{4.10}$$

The equation (4.10) defines the positions of the poles of the diode ladder filter. Apparently, it's easily solvable. We would be interested to find out, at which k does the selfoscillation start.

If the poles are to be on the imaginary axis, then $s = j\omega$. Substituting $s = j\omega$ into (4.10) we get

$$(1-\omega^2) + 2j\omega = \frac{1}{2} \mp \frac{j}{2}\sqrt{\frac{k-1}{2}} \tag{4.11}$$

Equating the real parts of (4.11) we obtain $1-\omega^2 = \frac{1}{2}$ and $\omega = \pm\frac{1}{\sqrt{2}}$. Equating the imaginary parts of (4.11) and substituting $\omega = \pm\frac{1}{\sqrt{2}}$ we obtain

$$\pm\frac{2}{\sqrt{2}} = \pm\frac{1}{2}\sqrt{\frac{k-1}{2}}$$

from where

$$\pm 4 = \pm\sqrt{k-1}$$

and, since $k \in \mathbb{R}$, we have $k = 17$.

That is, given the unit cutoff of the underlying one-pole lowpass filters, the selfoscillation starts at $k = 17$, where the resonance peak is located at $\omega = 1/\sqrt{2}$.

TPT model

Converting Fig. 4.15 to the instantaneous response form we obtain the structure in Fig. 4.17. From Fig. 4.17 we wish to obtain the instantaneous response of the entire diode ladder. Then we could use this response to solve the zero-delay feedback equation for the main feedback loop of Fig. 4.14.

From Fig. 4.17 we have $y_4 = gy_3 + s_4$. We can rewrite it as

$$y_4 = G_4 y_3 + S_4 \qquad \text{where } G_4 = g,\ S_4 = s_4$$

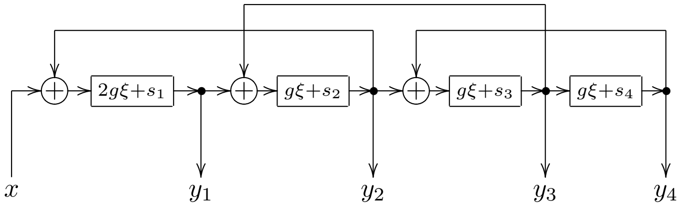


Figure 4.17: Linearized diode ladder in the instantaneous response form.

Further, from Fig. 4.17 we also have

$$y_3 = g(y_2 + y_4) + s_3 = g(y_2 + G_4 y_3 + S_4) + s_3$$

from where

$$y_3 = \frac{gy_2 + gS_4 + s_3}{1 - gG_4} = G_3 y_2 + S_3$$

$$\text{where } G_3 = \frac{g}{1 - gG_4},\ S_3 = \frac{gS_4 + s_3}{1 - gG_4}$$

Further,

$$y_2 = g(y_1 + y_3) + s_2 = g(y_1 + G_3 y_2 + S_3) + s_2$$

from where

$$y_2 = \frac{gy_1 + gS_3 + s_2}{1 - gG_3} = G_2 y_1 + S_2$$

$$\text{where } G_2 = \frac{g}{1-gG_3},\ S_2 = \frac{gS_3+s_2}{1-gG_3}$$

And ultimately

$$y_1 = 2g(x+y_2)+s_1 = 2g(x+G_2y_1+S_2)+s_1$$

from where

$$y_1 = \frac{2gx+2gS_2+s_1}{1-2gG_2} = G_1x+S_1$$

$$\text{where } G_1 = \frac{2g}{1-2gG_2},\ S_1 = \frac{2gS_2+s_1}{1-2gG_2}$$

Thus, we have

$$y_n = G_ny_{n-1}+S_n \qquad (\text{where } y_0 = x)$$

from where it's easy to obtain the instantaneous response of the entire diode ladder as

$$\begin{aligned}
y_4 &= G_4y_3+S_4 = G_4(G_3y_2+S_3)+S_4 = \\
&= G_4G_3y_2+(G_4S_3+S_4) = \\
&= G_4G_3(G_2y_1+S_2)+(G_4S_3+S_4) = \\
&\qquad = G_4G_3G_2y_1+(G_4G_3S_2+G_4S_3+S_4) = \\
&= G_4G_3G_2(G_1x+S_1)+(G_4G_3S_2+G_4S_3+S_4) = \\
&\qquad = G_4G_3G_2G_1x+(G_4G_3G_2S_1+G_4G_3S_2+G_4S_3+S_4) = \\
&= Gx+S
\end{aligned}$$

Notice, that we should have checked that we don't have instantaneously unstable feedback problems within the diode ladder.

That is, we need to check that all denominators in the expressions for G_n and S_n don't turn to zero or to negative values. Considering that $0 < g < \frac{1}{2}$ and that $G_4 = g$, we have

$$0 < G_4 < \frac{1}{2}$$

and

$$1 - gG_4 = 1 - g^2 > \frac{3}{4}$$

Respectively

$$0 < G_3 = g/(1 - gG_4) < \frac{1/2}{3/4} = \frac{2}{3}$$

and

$$1 - gG_3 > 1 - \frac{1}{2} \cdot \frac{2}{3} = 1 - \frac{1}{3} = \frac{2}{3}$$

Then

$$0 < G_2 = g/(1 - gG_3) < \frac{1/2}{2/3} = \frac{3}{4}$$

and

$$1 - 2gG_2 > 1 - 2\frac{1}{2}\frac{3}{4} = 1 - \frac{3}{4} = \frac{1}{4}$$

and thus all denominators are always positive.

Also

$$0 < G_1 = \frac{2g}{1 - 2gG_2} < \frac{1}{1/4} = 4$$

Thus

$$0 < G = G_4G_3G_2G_1 < \frac{1}{2} \cdot \frac{2}{3} \cdot \frac{3}{4} \cdot 4 = 1$$

Using the obtained instantaneous response of the entire diode ladder we now can solve the main feedback equation for Fig. 4.14.

For a nonlinear diode ladder model we could use the structure in Fig. 4.13. However, it might be too complicated to process. Even the application of the “cheap” TPT nonlinear processing approach is not fully trivial.

One can therefore use simpler nonlinear structures instead, e.g. the one from Fig. 4.9. Also, the other ideas discussed for the transistor ladder can be applied. In regards to the multimode diode ladder filter, notice that the transfer functions corresponding to the $y_n(t)$ outputs are different from the ones of the transistor ladder, therefore the mixing coefficients which worked for the modes of the transistor ladder filter, are not going to work the same for the diode ladder.

SUMMARY

The transistor ladder filter model is constructed by placing a negative feedback around a chain of four identical 1-pole lowpass filters. The feedback amount controls the resonance. A nonlinearity in the feedback path (e.g. at the feedback point) could be used to contain the signal level, so that selfoscillation becomes possible.

Chapter 5

2-pole filters

The other classical analog filter model is the 2-pole filter design commonly referred to in the music DSP field as the *state-variable filter* (SVF). It can also serve as a generic analog model for building 2-pole filters, similarly to previously discussed 1-pole RC filter model.

5.1 Linear analog model

The block diagram of the state-variable filter is shown in Fig. 5.1. The three outputs are the highpass, bandpass and lowpass signals.[1][2]

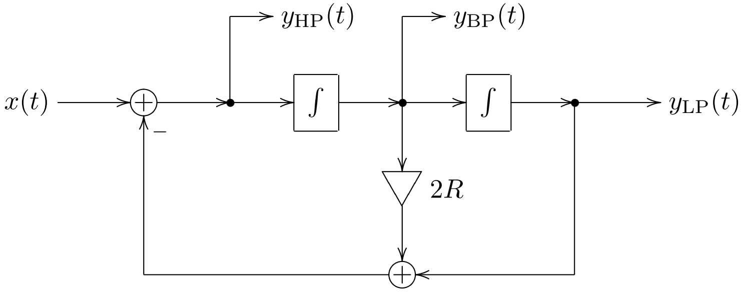


Figure 5.1: 2-pole multimode state-variable filter.

From Fig. 5.1 one can easily obtain the transfer functions for the respective signals. Assume complex exponential signals. Then, assuming unit cutoff,

$$y_{\text{HP}} = x - 2Ry_{\text{BP}} - y_{\text{LP}}$$
$$y_{\text{BP}} = \frac{1}{s}y_{\text{HP}}$$

[1] One can notice that the filter in Fig. 5.1 essentially implements an analog-domain canonical form, similar to the one in Fig. 3.20. Indeed let's substitute in Fig. 3.20 the z^{-1} elements by s^{-1} elements (integrators) and let $a_1 = -2R$, $a_2 = -1$. Then the gains b_0, b_1 and b_2 are simply picking up the highpass, bandpass and lowpass signals respectively.

[2] As usual, one can apply transposition to obtain a filter with highpass, bandpass and lowpass inputs.

$$y_{\mathrm{LP}} = \frac{1}{s} y_{\mathrm{BP}}$$

from where

$$y_{\mathrm{HP}} = x - 2R \cdot \frac{1}{s} y_{\mathrm{HP}} - \frac{1}{s^2} y_{\mathrm{HP}}$$

from where

$$\left(1 + \frac{2R}{s} + \frac{1}{s^2}\right) y_{\mathrm{HP}} = x$$

and

$$H_{\mathrm{HP}}(s) = \frac{y_{\mathrm{HP}}}{x} = \frac{1}{1 + \frac{2R}{s} + \frac{1}{s^2}} = \frac{s^2}{s^2 + 2Rs + 1}$$

Thus

$$\begin{aligned} H_{\mathrm{HP}}(s) &= \frac{s^2}{s^2 + 2Rs + 1} \\ H_{\mathrm{BP}}(s) &= \frac{s}{s^2 + 2Rs + 1} \\ H_{\mathrm{LP}}(s) &= \frac{1}{s^2 + 2Rs + 1} \end{aligned}$$

The respective amplitude responses are plotted in Fig. 5.2, Fig. 5.3 and Fig. 5.4. One could observe that the highpass response is a mirrored version of the lowpass response,[3] while the band-

[3] As with the 1-pole filters, the highpass transfer function can be obtained from the lowpass transfer function by the $s \leftarrow 1/s$ (LP to HP) substitution.

pass response is symmetric by itself.[4] It can also be easily shown mathematically. The slope rolloff speed is apparently -12dB/oct for the low- and highpass, and -6dB/oct for the bandpass.

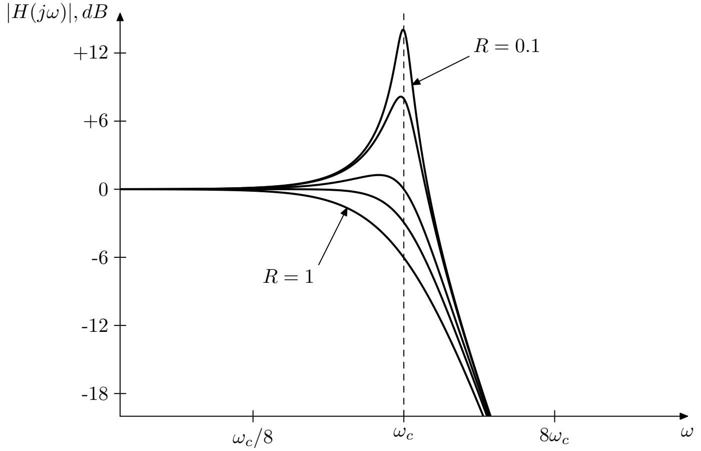


Figure 5.2: Amplitude response of a 2-pole lowpass filter.

[4] The bandpass response can be obtained from the lowpass response by a so-called *LP to BP* (lowpass to bandpass) substitution:

$$s \leftarrow \frac{1}{2R} \cdot \left(s + \frac{1}{s}\right)$$

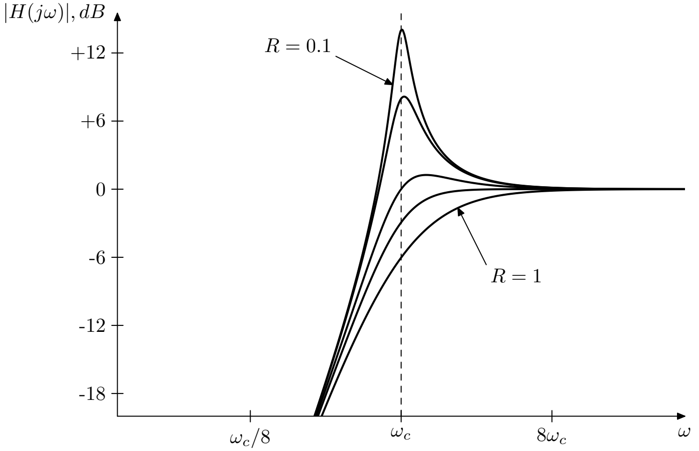


Figure 5.3: Amplitude response of a 2-pole highpass filter.

Notice that $y_{\text{LP}}(t) + 2Ry_{\text{BP}}(t) + y_{\text{HP}}(t) = x(t)$, that is, the input signal is split into lowpass, bandpass and highpass components. The same can be expressed in the transfer function form:

$$H_{\text{LP}}(s) + 2RH_{\text{BP}}(s) + H_{\text{HP}}(s) = 1$$

The resonance of the filter is controlled by the R parameter. Contrarily to the ladder filter, where the resonance increases with the feedback amount, in the state-variable filter the bandpass signal feedback serves as a damping means for the resonance. In the absence of the bandpass signal feedback the filter

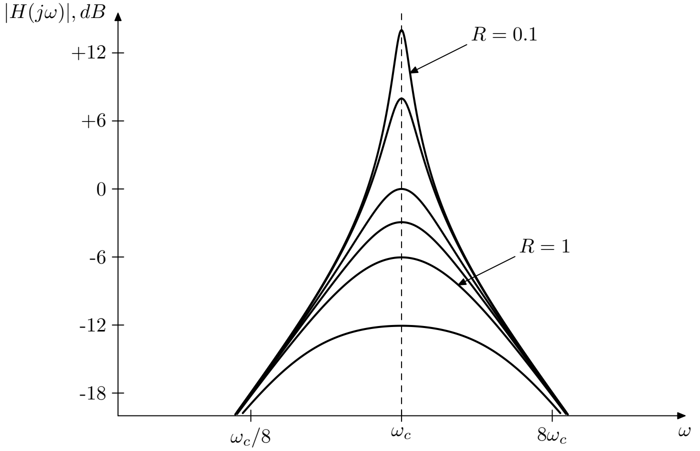


Figure 5.4: Amplitude response of a 2-pole bandpass filter.

will get unstable. The R parameter therefore may be referred to as the *damping* parameter.

Solving $s^2 + 2Rs + 1 = 0$ we obtain the poles of the filter at

$$s = -R \pm \sqrt{R^2 - 1} = \begin{cases} -R \pm \sqrt{R^2 - 1} & \text{if } R \geq 1 \\ -R \pm j\sqrt{1 - R^2} & \text{if } -1 \leq R \leq 1 \end{cases}$$

Without trying to give a precise definition of the resonance concept, we could say that at $R = 1$ there is no resonance (there are two real poles at $s = -1$). As R starts decreasing from 1 towards 0 there appear two mutually conjugate complex poles moving along the unit circle towards the imaginary axis, so the resonance slowly appears. At $R = 0$ the filter becomes unstable.

The amplitude response at the cutoff ($\omega = 1$) is $1/2R$ for all three filter types. Except for the bandpass, the point $\omega = 1$ is not exactly the peak location but it's pretty close (the smaller the value of R, the closer is the true peak to $\omega = 1$). The phase response at the cutoff is $-90°$ for lowpass, $0°$ for bandpass and $+90°$ for highpass.

5.2 Linear digital model

Skipping the naive implementation, which the readers should be perfectly capable of creating and analysing themselves by now, we proceed with the discussion of the TPT model.

Assuming $g\xi + s_n$ instantaneous responses for the two trapezoidal integrators one can redraw Fig. 5.1 to obtain the discrete-time model in Fig. 5.5.

Picking y_{HP} as the zero-delay feedback equation's unknown[5] we obtain from Fig. 5.5:

$$y_{\text{HP}} = x - 2R(gy_{\text{HP}} + s_1) - g(gy_{\text{HP}} + s_1) - s_2$$

from where

$$\left(1 + 2Rg + g^2\right) y_{\text{HP}} = x - 2Rs_1 - gs_1 - s_2$$

[5] The state-variable filter has two feedback paths sharing a common path segment. In order to obtain a single feedback equation rather than an equation system we should pick a signal on this common path as the unknown variable.

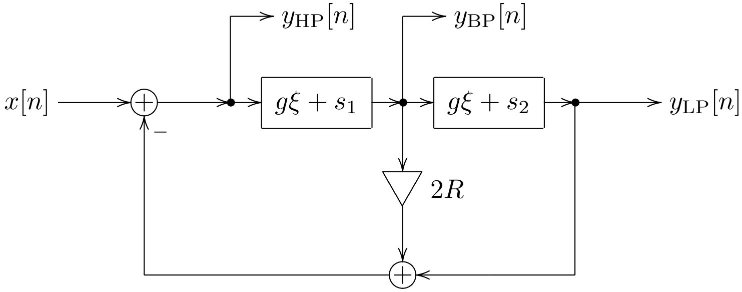


Figure 5.5: TPT 2-pole multimode state-variable filter in the instantaneous response form.

from where

$$y_{\mathrm{HP}} = \frac{x - 2Rs_1 - gs_1 - s_2}{1 + 2Rg + g^2} \tag{5.1}$$

Using y_{HP} we can proceed defining the remaining signals in the structure.[6]

5.3 Further filter types

By mixing the lowpass, bandpass and highpass outputs one can obtain further filter types.

[6] Apparently, $1 + 2Rg + g^2 > 0\ \forall g > 0$, provided $R > -1$. Thus, instantaneously unstable feedback may appear only if $R \leq -1$.

Unit gain bandpass

$$H_{\mathrm{BP1}}(s) = 2RH_{\mathrm{BP}}(s) = \frac{2Rs}{s^2 + 2Rs + 1}$$

This version of the bandpass filter has a unit gain at the cutoff (Fig. 5.6). Notice that the unit gain bandpass signal can be directly picked up at the output of the $2R$ gain element in Fig. 5.1.

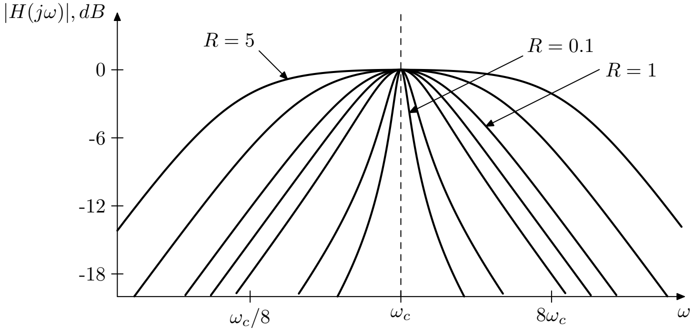


Figure 5.6: Amplitude response of a 2-pole unit gain bandpass filter.

Band-shelving filter

By adding/subtracting the unit gain bandpass signal to/from the input signal one obtains the band-shelving filter (Fig. 5.7):[7]

$$H_{\text{BS}}(s) = 1 + 2RKH_{\text{BP}}(s) = 1 + \frac{2RKs}{s^2 + 2Rs + 1}$$

Similarly to the other shelving filter types we can specify the mid-slope requirement $|H_{\text{BS}}(j\omega)| = \sqrt{1+K}$ (for some ω), from where we obtain

$$R = \frac{\left|\omega - \omega^{-1}\right|}{2\sqrt{1+K}} = \frac{\left|2^{\Delta/2} - 2^{-\Delta/2}\right|}{2\sqrt{1+K}}$$

where Δ is the desired mid-slope bandwidth (in octaves) of the peak.

Notch filter

At $K = -1$ the band-shelving filter turns into a notch (or band-stop) filter (Fig. 5.8):

$$H_{\text{N}}(s) = 1 - 2RH_{\text{BP}}(s) = \frac{s^2 + 1}{s^2 + 2Rs + 1}$$

[7] In the same way one could obtain 2-pole low- and high-shelving filters, however one has to be careful, because the phase response of the underlying low- and high-pass filters might produce weird peaks and dips in the resulting shelving filter's response.

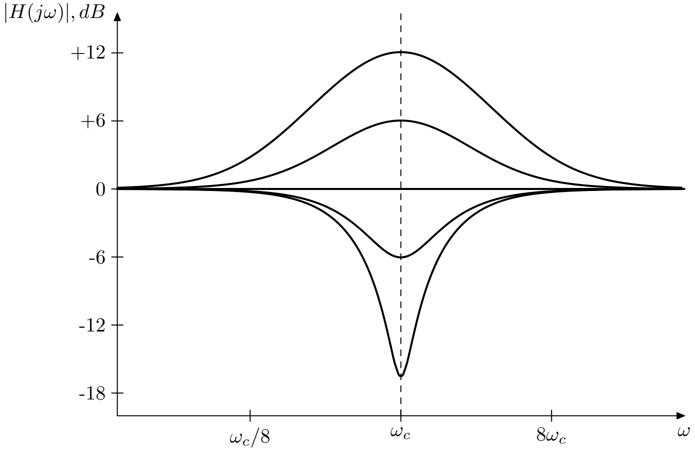


Figure 5.7: Amplitude response of a 2-pole band-shelving filter for $R = 1$ and varying K.

Allpass filter

At $K = -2$ the band-shelving filter turns into an allpass filter (Fig. 5.9):

$$H_{\text{AP}}(s) = 1 - 4RH_{\text{BP}}(s) = \frac{s^2 - 2Rs + 1}{s^2 + 2Rs + 1}$$

Notice how the damping parameter affects the phase response slope.

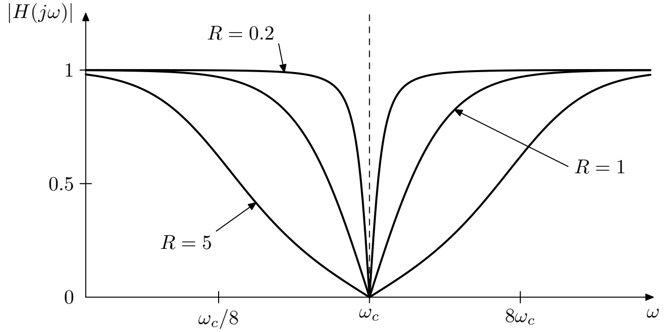


Figure 5.8: Amplitude response of a 2-pole notch filter. The amplitude scale is linear.

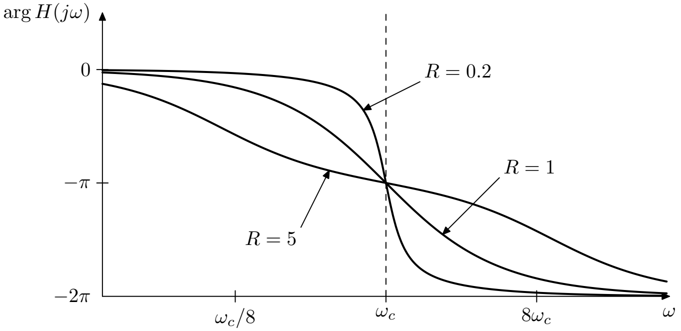


Figure 5.9: Phase response of a 2-pole allpass filter.

Peaking filter

By subtracting the highpass signal from the lowpass signal (or also vice versa) we obtain the peaking filter (Fig. 5.10):

$$H_{\text{PK}}(s) = H_{\text{LP}}(s) - H_{\text{HP}}(s) = \frac{1 - s^2}{s^2 + 2Rs + 1}$$

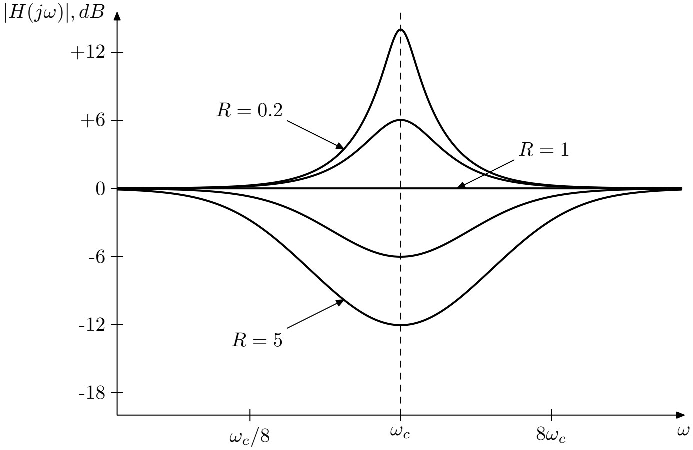


Figure 5.10: Amplitude response of a 2-pole peaking filter.

5.4 Nonlinear model

In the ladder filter the resonance was created as the result of the feedback. Therefore by limiting the feedback level (by a saturator) we could control the resonance amount and respectively prevent the filter from becoming unstable.

The feedback in the SVF has a more complicated structure. Particularly, the bandpass path is responsible for damping the resonance. We could therefore apply an inverse idea: try boosting the bandpass signal in case the signal level becomes too strong. This can be achieved e.g. by placing an inverse hyperbolic tangent nonlinearity into the bandpass feedback path (Fig. 5.11).[8] The idea is that at low signal levels

$$\tanh^{-1} y_{\mathrm{BP}'} + (R-1)y_{\mathrm{BP}'} \approx Ry_{\mathrm{BP}'}$$

at higher signal levels the nonlinearity boosts the bandpass signal. Notice that the bandpass signal should be rather picked up at the output of the nonlinearity than at the $y_{\mathrm{BP}'}$ point to make sure it has a similar level as the lowpass and highpass.

The nonlinear feedback equation can be obtained using y_{HP} as the unknown:

$$y_{\mathrm{HP}} = x - 2\tanh^{-1}(gy_{\mathrm{HP}}+s_1) - 2(R-1)(gy_{\mathrm{HP}}+s_1) - g(gy_{\mathrm{HP}}+s_1) - s_2$$

Instead of using the hyperbolic tangent function, one could also use a hyperbolic shaper of a similar shape $f(x) = x/(1-|x|)$

[8] Since the domain of the inverse hyperbolic tangent is restricted to $(-1, 1)$, the “cheap” TPT nonlinearity application method doesn’t work in this case, at least not in a straightforward manner.

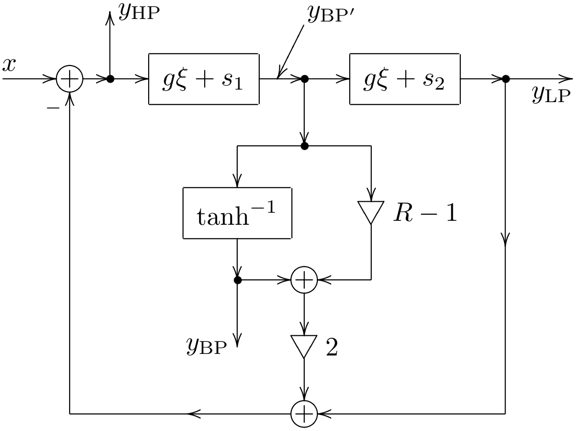


Figure 5.11: 2-pole state-variable filter with a nonlinearity.

to allow analytical solution of the nonlinear zero-delay feedback equation.

A more straightforward possibility is to introduce saturation into the integrators, or at least into the first of them. In the TPT approach one could use the direct form I-style integrator (Fig. 3.8) resulting in the integrator in Fig. 5.12. Or one could equivalently use the transposed direct form II-style integrator (Fig. 3.10) resulting in the integrator in Fig. 5.13.

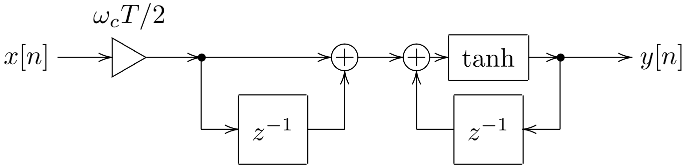


Figure 5.12: Saturating direct form I trapezoidal integrator.

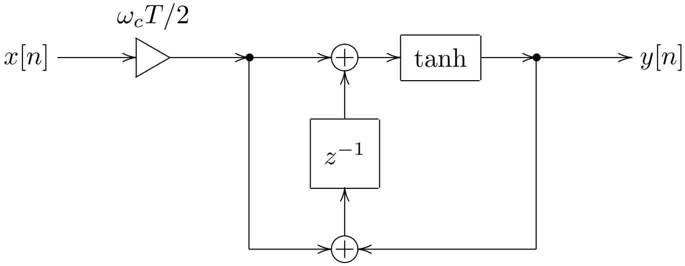


Figure 5.13: Saturating transposed direct form II trapezoidal integrator.

5.5 Cascading decomposition

First, recall that a 1-pole multimode filter can be used to implement any 1st-order rational transfer function. Similarly, a multimode SVF can be used to implement practically any 2nd-order rational transfer function. Indeed, consider

$$H(s)=\frac{b_2s^2+b_1s+b_0}{s^2+a_1s+a_0}$$

where we assume $a_0>0$.[9]

Let's perform the cutoff scaling substitution $s\leftarrow s\sqrt{a_0}$, which turns $H(s)$ into something of the form

$$H(s)=\frac{b_2s^2+b_1s+b_0}{s^2+a_1s+1}$$

(the coefficient values are now different in the above). Now simply let $R=a_1/2$ and use b_2, b_1 and b_0 as the mixing coefficients for the highpass, bandpass and lowpass signals.

This further allows to implement practically any given stable transfer function by a serial connection of a number of 2-pole

[9] If $a_0<0$, this means that $H(s)$ has two real poles of different signs. If $a_0=0$ then at least one of the poles is at $s=0$. In either case, this filter is already unstable, which means, if we are practically interested in its implementation, most likely there is a nonlinear analog prototype, and we simply can apply TPT to this prototype to obtain a digital structure. If we insist on using the SVF structure (why would we?), we can also extend it to the canonical form by introducing a gain element into the lowpass feedback path.

(and possibly 1-pole) filters. Indeed, simply factor the numerator and the denominator into 2nd- and possibly 1st-order factors (where the 2nd-order real factors will necessarily appear for complex conjugate pairs of roots and optionally for pairs of real roots). Any pair of 2nd-order factors (one in the numerator, one in the denominator) can be implemented by a 2-pole multimode SVF. Any pair of 1st-order factors can be implemented by a 1-pole multimode. If there are not enough 2nd-order factors in the numerator or denominator, a pair of 1st order factors in the numerator or denominator can be combined into a 2nd-order factor.

SUMMARY

The state-variable filter has the structure shown in Fig. 5.1. Contrarily to the ladder filter, the resonance strength in the SVF is controlled by controlling the damping signal. The multimode outputs have the transfer functions

$$\begin{aligned}
H_{\mathrm{HP}}(s) &= \frac{s^2}{s^2+2Rs+1} \\
H_{\mathrm{BP}}(s) &= \frac{s}{s^2+2Rs+1} \\
H_{\mathrm{LP}}(s) &= \frac{1}{s^2+2Rs+1}
\end{aligned}$$

and can be combined to build further filter types.

Chapter 6

Allpass-based effects

Phasers are essentially ladder filters built around allpass filters instead of lowpass filters. Flangers can be obtained from phasers by an allpass substitution. For these reasons both types belong to the VA filter discussion.

6.1 Phasers

The simplest *phaser* is built by mixing the unmodified (dry) input signal with an allpass-filtered (wet) signal as in Fig. 6.1, where the allpass filter's cutoff is typically modulated by an LFO.[1] The allpass filter can be rather arbitrary, except than it

[1] In the absence of LFO modulation the structure should be rather referred to as a (multi-) notch filter.

has to be a differential filter.[2]

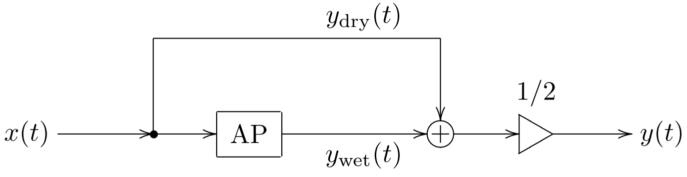


Figure 6.1: The simplest phaser.

At the points where the allpass filter's phase response is 180°, the wet and the dry signals will cancel each other, producing a notch. At the points where the allpass filter's phase response is 0° the wet and the dry signals will boost each other, producing a peak (Fig. 6.2).

The phaser structure in Fig. 6.1 contains no feedback, therefore there is no difference between naive and TPT digital implementations (except that the underlying allpass filters should be better constructed in a TPT way).

Mixing at arbitrary ratios

Instead of mixing at the 50/50 ratio we can mix at any other ratio, where the sum of the dry and wet mixing gains should

[2] Phasers typically use differential allpass filters or their digital counterparts. If e.g. a delay (which is not a differential filter, but *is* an allpass) is used as the allpass, the structure should be rather referred to as a *flanger*.

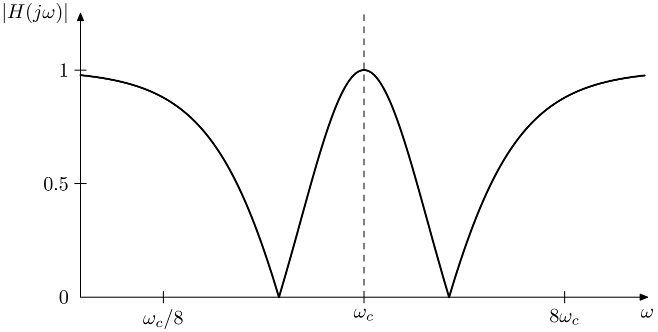


Figure 6.2: Amplitude response of the simplest phaser from Fig. 6.1 (using four identical 1-pole allpass filters with the same cutoff as the allpass core of the phaser).

amount to unity. This will affect the depth of the notches and the height of the peaks. For the phaser in Fig. 6.1 the mixing ratio higher than 50/50 (where the wet signal amount is more than 50%) hardly makes sense.

Wet signal inversion

By inverting the wet signal, one swaps the peaks and the notches. Notice that the phase response of differential allpasses at $\omega = 0$ can be either 0° or 180°, the same holds for the phase response at $\omega = +\infty$. For that reason the possibility to swap the peaks and the notches might be handy.

Notch spacing

In the simplest case one uses a series of identical 1-pole allpasses inside a phaser. In order to control the notch spacing in an easy and nice way, one should rather use a series of identical 2-pole allpasses. As mentioned earlier, by changing the resonance amount of the 2-pole allpasses one controls the phase slope of the filters. This affects the spacing of the notches (Fig. 6.3).

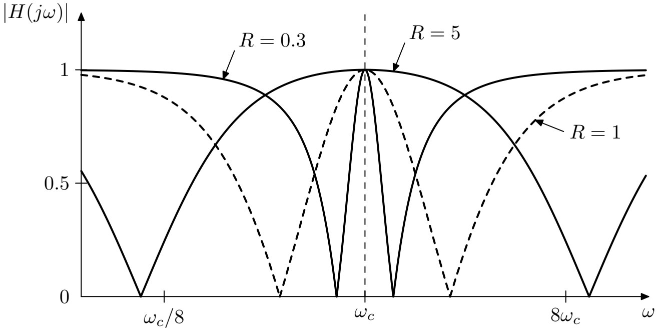


Figure 6.3: Effect of the allpass resonance on the notch spacing (using two 2-pole allpass filters as the allpass core of the phaser).

Feedback

We can also introduce feedback into the phaser. Similarly to the case of the ladder filter modes, the dry signal is better picked up after the feedback point (Fig. 6.4) The feedback changes the shape of the peaks and notches (Fig. 6.5).

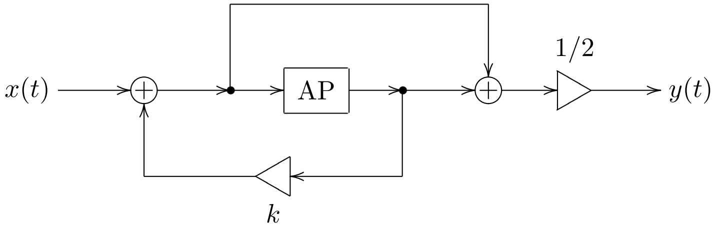


Figure 6.4: Phaser with feedback.

With the introduction of feedback we have a zero-delay feedback loop in the phaser structure. It can be solved using typical TPT means.[3]

6.2 Flangers

A delay is a linear time-invariant allpass. It even has a transfer function $H(s) = e^{-sT}$ where T is the delay time. Obviously $|H(s)| = |e^{-sT}| = 1$. However it is not a differential filter, for that reason the transfer function is not a rational function of

[3] Inserting a unit delay in the feedback produces subtle but rather unpleasant artifacts in the phasing response, one should better use the TPT approach here.

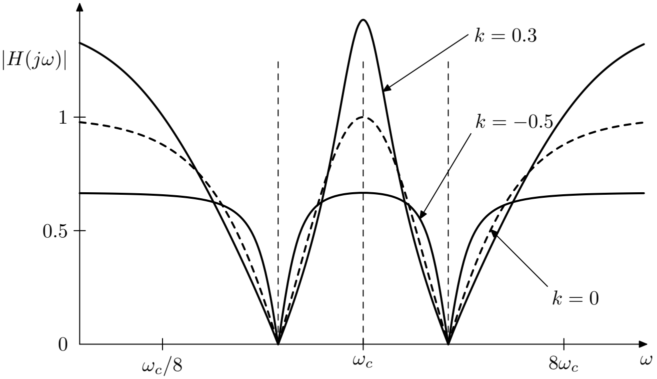


Figure 6.5: Effect of the feedback amount in Fig. 6.4 on the notch and peak shapes.

s. Digital delay models are typically built using interpolation techniques, the details of which fall outside the scope of this book.

Using the allpass substitution principle we can replace the allpass filter chain in a phaser by a delay. This produces a *flanger*. The discussion of the phasers mostly didn't assume any details about the underlying allpass, therefore most of it is applicable to flangers.

The main difference with using a delay is that the 0° and 180° phase response points are evenly spaced in the linear frequency scale (Fig. 6.6), whereas the spacing of the same points in responses of differential allpasses is not that regular. Also, a delay's phase response can easily have lots of 0° and 180° points (the larger the delay time is, the more of those points it has within the audible frequency range), while the number of those points in a differential allpass filter's phase response is limited by the filter's order.

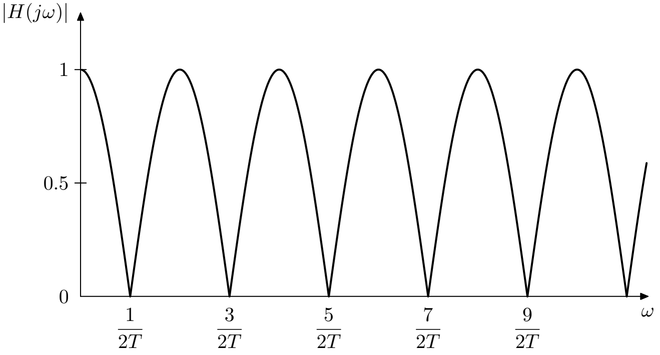


Figure 6.6: Amplitude response of the simplest flanger using the structure from Fig. 6.1.

Rather than modulating the delay time linearly by an LFO, one should consider that a filter's cutoff should be typically modulated in the logarithmic frequency scale (a.k.a. the pitch scale), therefore one in principle should do the same for the delay in a flanger. The delay's cutoff for that purpose can be simply defined as $\omega_c = 2\pi/T$, where T is the delay time.

SUMMARY

A phaser is made of an allpass differential filter connected in parallel with the dry signal path. This creates notches at the points of 180° phase difference and peaks at 0° points. The allpass cutoff should be modulated by an LFO. Using a delay instead of a differential allpass creates a flanger. Feedback can be used to change the shape of the peaks and notches in the amplitude response.

Index