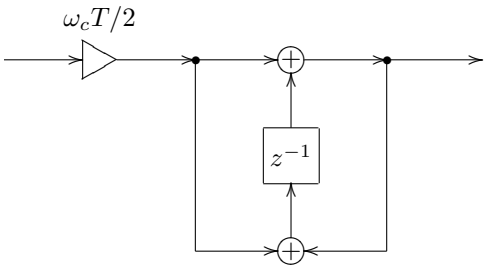
THE ART OF VA FILTER DESIGN



Vadim Zavalishin

rev. 2.0.0alpha (May 28, 2018)

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 main purpose of this book is not to explain how to build high-quality emulations of analog hardware (although the techniques explained in the book can be an important and valuable tool for building VA emulations). Rather it is about how to build high-quality time-varying digital filters. The author hopes that these techniques will be used to construct new digital filters, rather than only to build emulations of existing analog structures.

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 apologizes for possible mistakes and messy explanations, as the book didn't go through any serious proofreading.

Preface to revision 2.0.0alpha

This preface starts with an excuse. With revision 2.0.0 the book receives a major update, where the new material roughly falls into two different categories: the practical side of VA DSP and a more theoretical part. The latter arose from the desire to describe theoretical foundations for the subjects which the book intended to cover. These foundations were not copied from other texts (except where explicitly noted), but were done from scratch, the author trying to present the subject in the most intuitive way.[1] For that reason, especially in the more theoretical part, the book possibly contains mistakes.

Certain pieces of information are simply ideas which the author spontaneously had and tried to describe,[2] not necessarily properly testing all of them. This is another potential source of mistakes. One option would have been not rushing the book release and making an exhaustive testing of the presented material. During the same time the book text could have gone through a few more polishing runs, possibly restructuring some of the material in an easier to grasp way. However, this probably would have delayed the book's release by half a year or, likely, much more, as after five months of overly intensive work on the book the author (hopefully) deserves some relaxing. On the other hand, the main intention of the book is not to provide a collection of ready to use recipes, but rather to describe one possible way to think about the respective matters and give some key pieces of information. Thus, readers, who understood the text, should be able to correct the respective mistakes, if any, on their own. From that perspective, the book in the present state should fulfill its goal.

Therefore the author decided to release the book in an *alpha* state with the above reservations. Readers looking for a collection of time-proven recipes might want to check other sources.

The author also has recieved a number of complaints in regards to the book having too high requirements on the math side. It just so happens that certain things simply need advanced math to be properly understood. Sacrificing the exactness and the amount of information for the sake of a more accessible text could have definitely been an option, but... that would have been a completely different book. In that regard the new revision contains parts which are even harder on the math side than the previous revisions, the math prerequisites for these parts respectively being generally higher than for the rest of the book. Such parts, however, may simply be skipped by the readers.

In regards to the usage of the math in the book, the author would like to make one more remark. The book uses math notation not simply to provide some calculation formulas or to do formal transformations. The math notation is also used to express information, since quite in some cases it can do this much more exactly than words. In that sense the respective formulas become an integral part of the book's text, rather than some kind of a parallel stream of information. E.g. the formula (2.4), which some readers find daunting, is simply providing a detailed explanation to the statement that each partial can

[1] "Intuitive" here doesn't mean "easy to understand", but rather "when understood, it becomes easy".

[2] It is possible that some of these ideas are not new, but the author at the time of the writing was not aware of that. This might result in a lack of respective credits and in a different terminology, for which, should that happen to be the case, the author apologizes.

be integrated independently.

Certain readers, being initially daunted by the look of the text, also believe that they need to read some other filter DSP text before attempting this one. This is not necessarily so, since this book strongly deviates in its presentation from the classical DSP texts and this might create a collision in the beginner's mind between two very different approaches to the material. Also, chances are, after reading some other classical DSP text first, the reader will only find out that this didn't help much in regards to understanding this book and was simply an additional investment of time.

The part of DSP knowledge which is more or less required (although a pretty surface level should suffice) is a basic understanding of discrete time sampling. Also basic knowledge of Fourier theory could be helpful, but probably even that is not a must, as the book introduces it in a, however condensed, but sufficient for the understanding of the the further text form. No preliminary knowledge of filters is needed. Also, in author's impression, often the real problem is possibly an insufficient level of math knowledge or experience, which then leads to a reader believing that some additional filter knowledge is needed first, whereas what's lacking is rather purely the math skills. In this case, if the gap is not very large, one could try to simply read through anyway, it might become progressively better, or the part of the math which is not being understood may happen to be not essential for practical application of the materials.

Acknowledgements

The author would like to express his gratitude to a number of people who 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, Mike Daliot and Jelena Mičetić Krowarz. Special thanks to Stephan Schmitt, Egbert Jürgens, Tobias Baumbach, Steinunn Arnardottir, Eike Jonas, Maximilian Zagler, Marin Vrbica and Philipp Dransfeld.

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. Particularly 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, to the forum member "neotec" for the introduction of the transposed direct form II BLT integrator in the TPT context, to Teemu Voipio and Max Mikhailov for their active involvement into the related discussions and research and to Urs Heckmann for being an active proponent of the ZDF techniques and actually (as far as the author knows) starting the whole avalanche of their usage. Thanks to Robin Schmidt, Richard Hoffmann and Francisco Garcia 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 initial idea of writing this book. Thanks to Julian Parker for productive discussions, which stimulated the creation of the book's next revision.

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

Prior work credits

Various flavors and applications of delayless feedback techniques were in prior use for quite a while. Particularly there are works by A.Härmä, F.Avancini, G.Borin, G.De Poli, F.Fontana, D.Rocchesso, T.Serafini and P.Zamboni, although reportedly this subject has been appearing as far ago as in the 70s of the 20th century.

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.

1.1 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:

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

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] More precisely we will develop a number of definitions.

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.

The set of partials contained in a signal (including the DC term) is referred to as the signal's *spectrum*. Respectively, a periodic signal can be specified by specifying its spectrum.

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 referred to as *complex-form Fourier series* and respectively we can talk of a *complex spectrum*. 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

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

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.

[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.

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 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.

Even though the set of partials is uncountable this time, we still refer to it as a *spectrum* of the signal. Thus, while periodic signals had discrete spectra (consisting of a set of discrete partials at the harmonically related frequencies), nonperiodic signals have continuous spectra.

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]

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.

[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 (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:

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

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 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.

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):

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

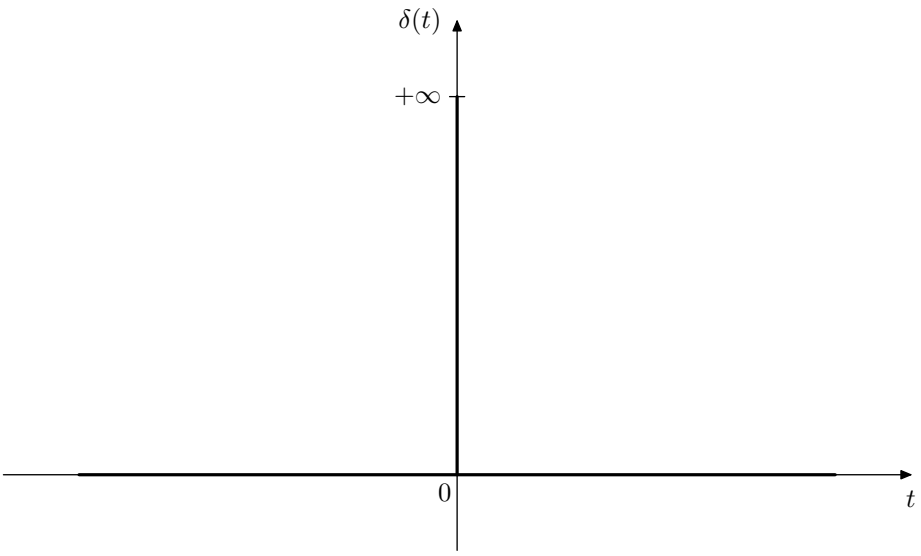


Figure 1.1: Dirac delta function.

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

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

Notice that thereby the spectrum $X(\omega)$ is discrete, even though being formally notated as a continuous function. 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

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

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 time-limited signals the function $X(s)$ 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 $\operatorname{Re} s = 0$.

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.

[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.

[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 $X(s)$).

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". The full set of these sinusoids, including their amplitudes and phases, is refereed to as the spectrum of 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.

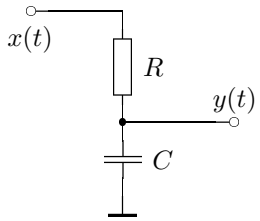


Figure 2.1: A simple RC lowpass filter.

Writing the equations for that circuit we have:

$$\begin{aligned} x &= U_R + U_C \\ y &= U_C \\ U_R &= RI \\ I &= \dot{q}_C \\ q_C &= CU_C \end{aligned} \tag{2.1}$$

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

$$\dot{y} = \frac{1}{RC}(x - y) \tag{2.2}$$

or, integrating with respect to time:

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

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

$$y = y(t_0) + \int_{t_0}^{t} \omega_c\big(x(\tau) - y(\tau)\big)\,\mathrm{d}\tau \tag{2.3}$$

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 (2.3) can be expressed in the block diagram form (Fig. 2.2).

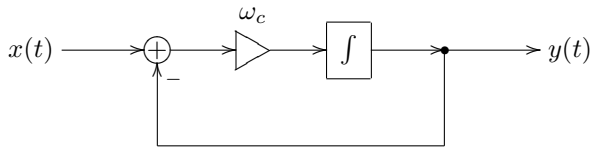


Figure 2.2: A 1-pole RC lowpass filter in the block diagram form.

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:

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

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

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

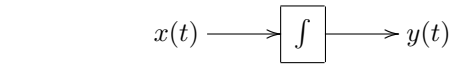
One can observe from Fig. 2.2 that the output signal is always trying to "reach" the input signal. Indeed, the difference $x - y$ is always "directed" from

[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.

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

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

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[2]. It can be shown that in this case the effect of the extra DC term on the output signal is negligible.[3] 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![4]

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,[5] not only are we able to factor $X(s)$ out of the integration:

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

[2] 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

[3] We will discuss the mechanisms behind that fact when we talk about *transient response*.

[4] 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 $t \gg t_0$).

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

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

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

$$\int \left(\int_{\sigma-j\infty}^{\sigma+j\infty} X(s)e^{s\tau} \frac{\mathrm{d}s}{2\pi j} \right) \mathrm{d}\tau = \int_{\sigma-j\infty}^{\sigma+j\infty} \left(\int X(s)e^{s\tau} \,\mathrm{d}\tau \right) \frac{\mathrm{d}s}{2\pi j} = \\ = \int_{\sigma-j\infty}^{\sigma+j\infty} \frac{X(s)}{s} e^{s\tau} \frac{\mathrm{d}s}{2\pi j} \tag{2.4}$$

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).[6]

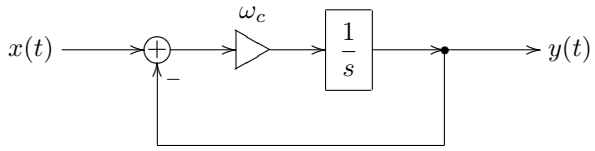


Figure 2.3: A 1-pole RC lowpass filter in the block diagram form with a $1/s$ notation for the integrator.

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*!!!)[7] 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

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

[6] 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 $1/s$.

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

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.

Let's introduce the notation

$$H(s) = \frac{\omega_c}{s+\omega_c} \tag{2.5}$$

Then

$$Y(s) = H(s)X(s) \tag{2.6}$$

$H(s)$ 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*[8] 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

$$y(t) = \int_{\sigma-j\infty}^{\sigma+j\infty} H(s)X(s)e^{st} \frac{\mathrm{d}s}{2\pi j} \tag{2.7}$$

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 differential systems, whose block diagrams consist of integrators, summators and fixed gains, the transfer function is always a *non-strictly proper*[9] 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

[8] 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.

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

during a 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".

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

$$I(t) = I(s)e^{st}$$
$$U(t) = U(s)e^{st}$$

(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 (2.5).

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[10] [11]

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

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!

[10] 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 $H(j\omega)$.

[11] 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 $\omega \ll \omega_c$ pass 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.

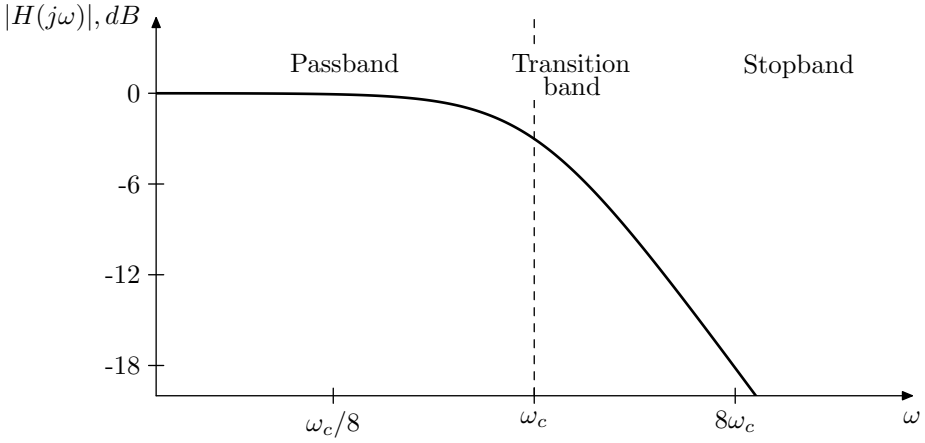


Figure 2.4: Amplitude response of a 1-pole lowpass filter.

The frequency range where $|H(j\omega)| \approx 1$ is referred to as the filter's *passband*. The frequency range where $|H(j\omega)| \approx 0$ is referred to as the filter's *stopband*. The frequency range between the passand and the stopband where $|H(j\omega)|$ is changing from approximately 1 to approximately 0 is referred to as the filter's *transition band*.[12]

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 amplitude

[12] We introduce the concepts of pass-, stop- and transition bands only qualitatively, without attempting to give more exact definitions of the positions of the boundaries between the bands.

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

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

The phase response of the 1-pole lowpass is respectively

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

giving 0 at $\omega = 0$, $-\pi/4$ at the cutoff and $-\pi/2$ at $\omega \to +\infty$. With phase response plots we don't want a logarithmic phase axis, but the logarithmic frequency scale is usually desired. Fig. 2.5 illustrates.

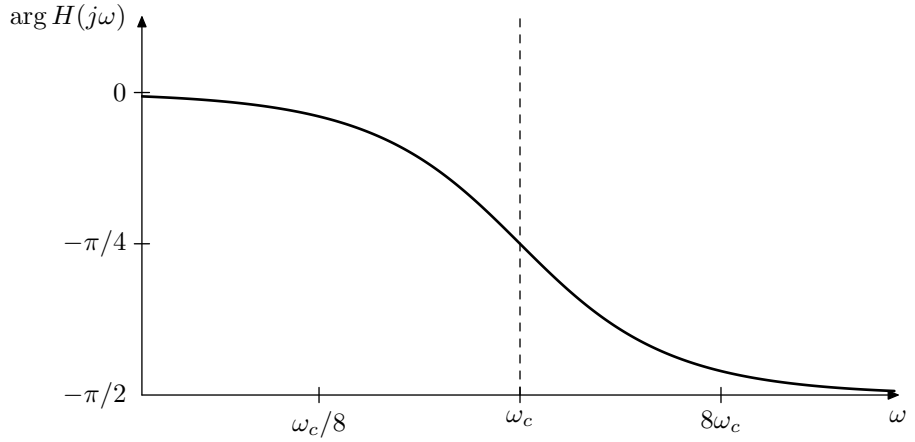


Figure 2.5: Phase response of a 1-pole lowpass filter.

Note that the phase response is close to zero in the passband, this will be a property encountered in most of the filters that we deal with.

2.7 Cutoff parameterization

Suppose $\omega_c = 1$. Then the lowpass transfer function (2.5) 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. 2.6), or of the phase response graph, for that matter (Fig. 2.7).

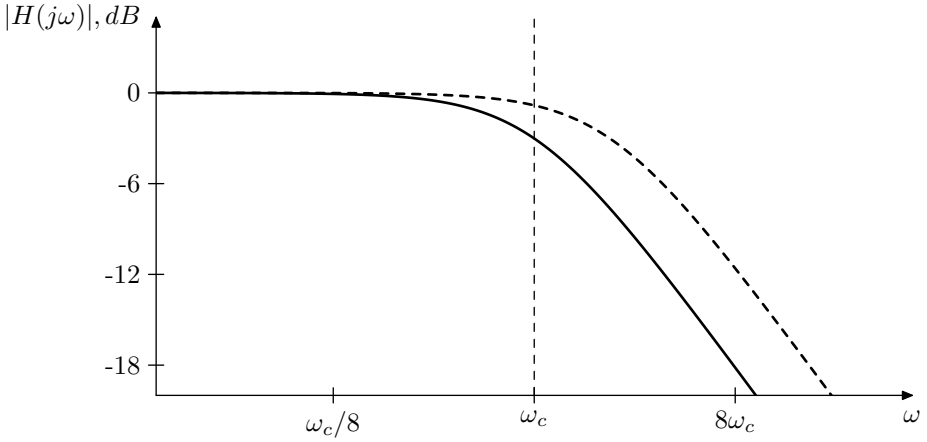


Figure 2.6: 1-pole lowpass filter's amplitude response shift by a cutoff change.

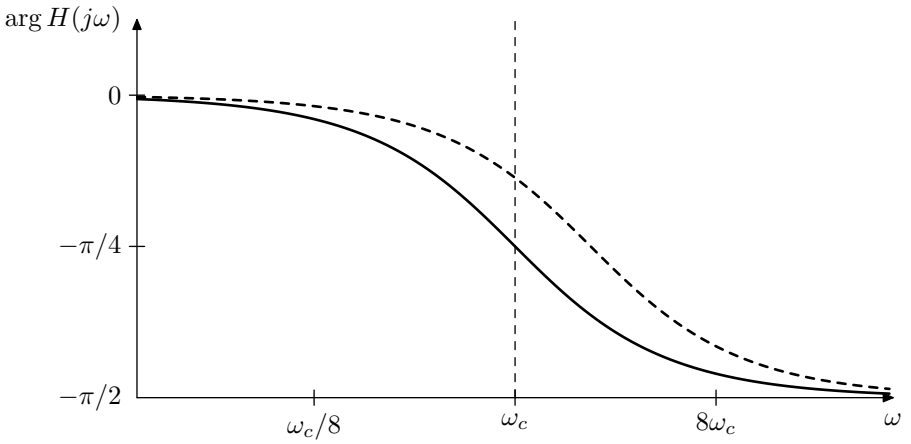
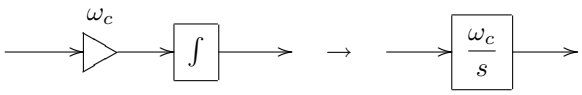


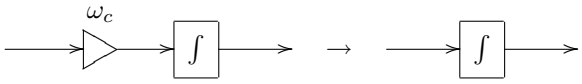
Figure 2.7: 1-pole lowpass filter's phase response shift by a cutoff change.

The substitution $s \leftarrow s/\omega_c$ is a generic way to handle cutoff parameterization for analog filters, because it doesn't change the 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:[13]



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 gain *before* the integrator, because then the integrator will smooth the jumps and further artifacts arising out of the cutoff modulation. Another reason to put the cutoff gain before the integrator is that it has an important impact on the behavior of the filter in the time-varying case. We will discuss this aspect in Section 2.16.

With the cutoff gain implied inside the integrator block, the structure from Fig. 2.2 is further simplified to the one in Fig. 2.8:

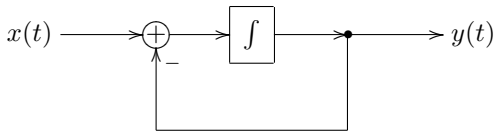


Figure 2.8: A 1-pole RC lowpass filter with an implied cutoff.

Unit-cutoff notation

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.

[13] 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.

To illustrate the convenience of the unit cutoff notation we will obtain the explicit expression for the 1-pole lowpass phase response shown in Fig. 2.5:

$$\arg H(j\omega) = \arg \frac{1}{1+j\omega} = -\arg(1+j\omega) = -\arctan\omega \tag{2.8}$$

The formula (2.8) explains the apparent from Fig. 2.5 symmetry (relative to the point at $\omega = \omega_c$) of the phase response in the logarithmic frequency scale, as this symmetry is simply due to the property of the arctangent function:

$$\arctan x + \arctan \frac{1}{x} = \frac{\pi}{2} \tag{2.9}$$

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. 2.9.

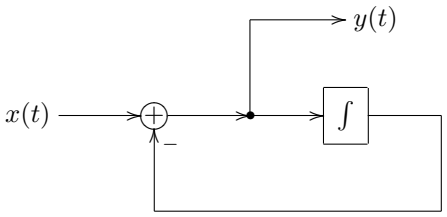


Figure 2.9: A 1-pole highpass filter.

Obtaining the transfer function of this filter we get

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

or, in the unit-cutoff form,

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

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. 2.10 (in the logarithmic scale).

It's not difficult to observe or show that this response is a mirrored version of the one in Fig. 2.4. 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 have a 6dB/oct rolloff.

The phase response of the highpass is a 90° shifted version of the lowpass phase response:

$$\arg \frac{j\omega}{1+j\omega} = \frac{\pi}{2} + \frac{1}{1+j\omega}$$

Fig. 2.11 illustrates. Note that the phase response in the passband is close to zero, same as we had for the lowpass.

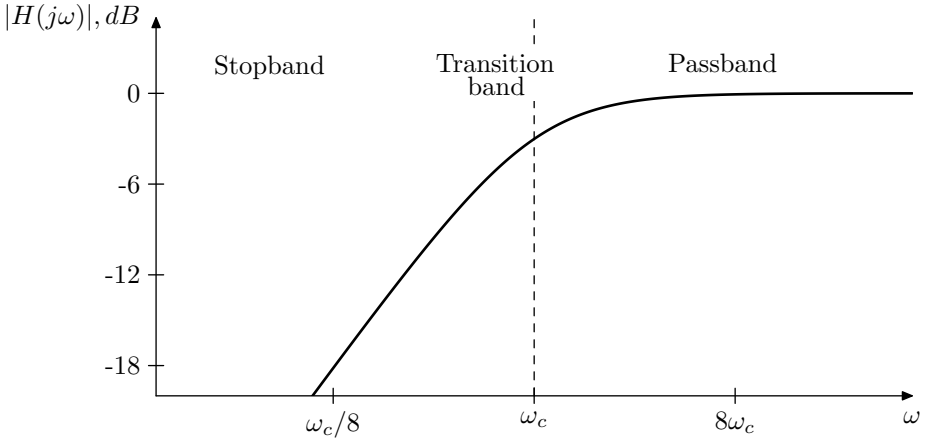


Figure 2.10: Amplitude response of a 1-pole highpass filter.

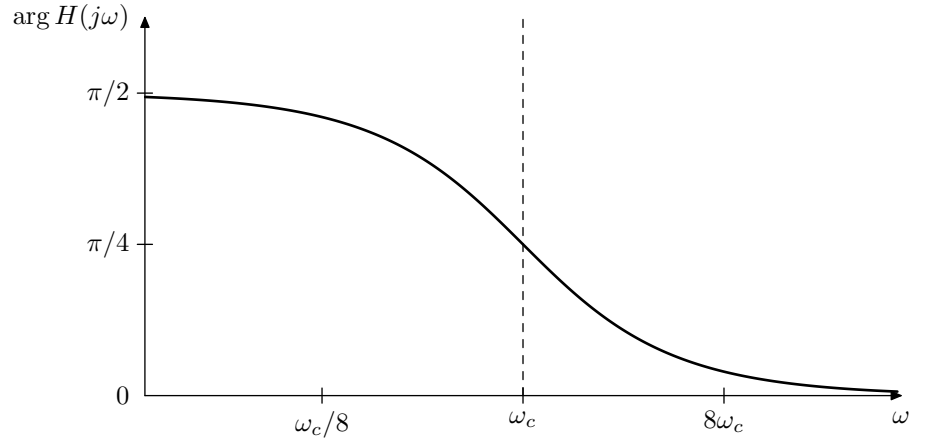


Figure 2.11: Phase response of a 1-pole highpass filter.

2.9 Poles and zeros

Poles and zeros are two very important concepts used in connection with filters. Now might be a good time to introduce them.

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. Writing any such function in the

multiplicative form we obtain

$$H(s) = g \cdot \frac{\prod_{n=1}^{N_z}(s - z_n)}{\prod_{n=1}^{N_p}(s - p_n)} \qquad (N_p \geq N_z \geq 0, \quad N_p \geq 1) \tag{2.10}$$

where N_p stands for the order of the denominator, simultaneously being the number of poles, and N_z stands for the order of the numerator, simultaneously being the number of zeros. Thus such transfer functions always have poles and often have zeros. 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 1-pole 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.[14] Therefore it is common, instead of e.g. a "4th-order filter" to say a "4-pole filter".

The number of poles therefore provides one possible way of classification of filters. It allows to get an approximate idea of how complex the filter is and also often allows to estimate some other filter properties without knowing lots of extra detail. The number of zeros in the filter is usually less important and therefore typically is not used for classification.

Finite and infinite zeros/poles

Equation (2.10) assumes that all p_n and z_n are finite. However often (especially when dealing with complex numbers) it is convenient to include the infinity into the set of "allowed" values. Respectively, if $N_z < N_p$ we will say that $H(s)$ has a zero of order $N_p - N_z$ at the infinity. E.g. the 1-pole lowpass transfer function has a zero of order 1 at the infinity.

Conversely, if $N_p > N_z$ we could say that $H(s)$ has a pole of order $N_p - N_z$ at the infinity, however this situation won't occur for a transfer function of a differential filter, since N_z cannot exceed N_p.

Apparently, zeros at the infinity are not a part of the explicit factoring (2.10) and occur implicity simply due to the difference of the numerator and denominator orders. Even though they don't show up in (2.10) they may occasionally show up in other formulas or transformations. Thus, whether the infinite zeros (or also poles, if we deal with other rational functions) are included into the set of zeros/poles under consideration depends on the context. Unless explicitly

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

mentioned, usually only finite zeros and poles are meant, however the readers are encouraged to use their own judgement in this regard.

Notice that if zeros/poles at the infinity are included, the total number of zeros is always equal to the total number of poles.

Rolloff

In (2.10) let $\omega \to +\infty$. Apparently, this is the same as simply letting $s \to \infty$ and therefore we obtain

$$H(s) \sim \frac{g}{s^{N_p - N_z}} \qquad (s \to \infty)$$

as the asymptotic behavior, which means that the amplitude response rolloff speed at $\omega \to +\infty$ is $6(N_p - N_z)$dB/oct.

Now suppose some of the zeros of $H(s)$ are located at $s = 0$ and let N_{z0} be the number of such zeros. Then, for $\omega \to 0$ we obtain

$$H(s) \sim g \cdot s^{N_{z0}} \qquad (s \to 0)$$

(assuming there are no poles at $s = 0$). Therefore the amplitude response rolloff speed at $\omega \to 0$ is $6N_{z0}$dB/oct. Considering that $0 \le N_{z0} \le N_z \le N_p$, 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.

The case of 0dB/oct rolloff deserves a special attention. The 0dB/oct at $\omega \to +\infty$ occurs when $N_p = N_z$. Respectively $H(s) \to g$ as $s \to \infty$. Since g must be real, it follows that so is $H(\infty)$, thus we arrive at the following statement: if $H(\infty) \neq 0$, then the phase response at the infinity is either $0°$ or $180°$. The same statement applies for $\omega \to 0$ if $N_{z0} > 0$, where we simply notice that $H(0)$ must be real due to $H(j\omega)$ being Hermitian.[15] The close-to-zero phase response in the passbands of 1-pole low- and high-passes is a particular case of this property.

Stability

The other, probably even more important property of the poles (but not zeros) is that they determine the stability of the filter. A filter is said to be *stable* (or, more exactly, BIBO-stable, where BIBO stands for "bounded input bounded output") if for any bounded input signal the resulting output signal is also bounded. In comparison, unstable filters "explode", that is, given a bounded input signal (e.g. a signal with the amplitude not exceeding unity), the output signal of such filter will grow indefinitely.

It is known that a filter[16] is stable if and only if all its poles are located

[15] Of course, $H(0)$ and $H(\infty)$ are real regardless of the rolloff speeds. However zero values of H do not have a defined phase response and can be approached from any direction on the complex plane of values of H. On the other hand a nonzero real value $H(0)$ or $H(\infty)$ means that $H(s)$ must be almost real in some neightborhood of $s = 0$ or $s = \infty$ respectively.

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

in the left complex semiplane (that is to the left of the imaginary axis).[17] For our lowpass and highpass filters this is apparently true, as long as $\omega_c > 0$. If $\omega_c < 0$, the pole is moved to the right semiplane, the filter becomes unstable and will "explode". This behavior can be conveniently explained in terms of the *transient response* of the filters and we will do so later.

We have established by now that if we put a sinusoidal signal through a stable filter we will obtain an amplitude-modified and phase-shifted sinusoidal signal of the same frequency (after the effects of the initial state, if such were initially present, disappear). In an unstable filter the effects of the initial state do not decay with time, but, on the opposite, infinitely grow, thus the output will not be the same kind of a sinusoidal signal and it doesn't make much sense to take of amplitude and phase responses, except maybe formally.

It is possible to obtain an intuitive understanding of the effect of the pole position on the filter stability. Consider a transfer function of the form (2.10) and suppose all poles are initially in the left complex semiplane. Now imagine one of the poles (let's say p_1) starts moving towards the imaginary axis. As the pole gets closer to the axis, the $(s - p_1)$ factor in the denominator becomes smaller around $\omega = \operatorname{Im} p_1$ and thus 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.[18]

It should be stressed once again, that the concepts of poles and zeros are bound to the concept of the transfer function and thus are properly defined only if the filter's parameters are not modulated. Sometimes one could talk about poles and/or zeros moving with time, but this is rather a convenient way to describe particular aspects of the change in the filter's parameters rather than a formally correct way. Although, if the poles and zeros are moving "slowly enough", this way of thinking could provide a good approximation of what's going on.

Cutoff

The cutoff control is defined as $s \leftarrow s/\omega_c$ substitution. Given a transfer function denominator factor $(s - p)$, after the cutoff substitution it becomes $(s/\omega_c - p)$. The pole associated with this factor becomes defined by the equation

$$s/\omega_c - p = 0$$

which gives $s = \omega_c p$. This means that the pole position is changed from p to $\omega_c p$.

Obviously, the same applies for zeros.

[17] The case when some of the poles are exactly on the imaginary axis, while the remaining poles are in the left semiplane is referred to as *marginally stable* case. For some of the marginally stable filters the BIBO property may still theoretically hold. However since in practice (due to noise in analog systems or precision losses in their digital emulations) it's usually impossible to have the pole locations exactly defined and we will not concern ourselves with this boundary case. One additional property of filters with all poles in the left semiplane is that their state decays to zero in the absence of the input signal. Marginally stable filters do not have this property.

[18] The reason, why the stable area is the left (and not the right) complex semiplane, is discussed later in connection with transient response.

Minimum and maximum phase

Consider a change to a filter's transfer function (2.10) where we flip one of the poles or zeros symmetrically with respect to the imaginary axis.[19] E.g. we replace p_1 with $-p_1^*$ or z_1 with $-z_1^*$. Apparently, such change doesn't affect the amplitude response of the filter.

Indeed, a pole's contribution to the amplitude response is, according to (2.10), $|j\omega - p_n|$, which is the distance from the pole p_n to the point $j\omega$. However the distance from the point $-p_n^*$ to $j\omega$ is exactly the same, thus replacing p_n with $-p_n^*$ doesn't change the amplitude response (Fig. 2.12). The same applies to the situation when we change a zero from z_n to $-z_n^*$.

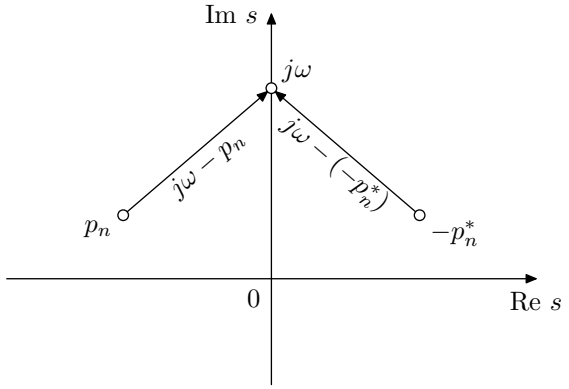


Figure 2.12: Contribution to the amplitude response from two symmetric points.

Flipping a pole symmetrically with respect to the imaginary axis normally doesn't make much sense, since this would turn a previously stable filter into an unstable one. Even though sometimes we will be specifically interested in using unstable filters (particularly if the filter is nonlinear), such flipping is not very useful. The point of the flipping is preserving the amplitude response and, as we mentioned, the concept of the amplitude response doesn't really work in the case of an unstable filter.

The situation is very different with zeros, though. Zeros can be located in both left and right semiplanes without endangering filter's stability. Therefore we could construct filters with identical amplitude responses, differing only in which of the zeros are positioned to the left and which to the right of the imaginary axis. Even though the amplitude response is not affected by this, the phase response apparently is, and this could be the reason to chose between the two possible positions of each (or all) of the zeros.

Qualitatively comparing the effect of the positioning a zero to the left or to the right, consider the following. A zero located to the left of the imaginary axis makes a contribution to the phase response which varies from $-90°$ to $+90°$ as ω goes from $-\infty$ to $+\infty$. A zero located on the right makes a contribution which varies from $+90°$ to $-90°$. That is, in the first case the phase is increasing by

[19] Conjugation p^* flips the pole p symmetrically with respect to the real axis. Now if we additionally flip the result symmetrically with respect to the origin, the result $-p^*$ will be located symmetrically to p with respect to the imaginary axis.

180° as ω goes from $-\infty$ to $+\infty$, in the second case it is decreasing by 180°.

The phase is defined modulo 360° and generally we cannot compare two different values of the phase. E.g. if we have two values $\varphi_1 = +120^\circ$ and $\varphi_2 = -90^\circ$, we can't say for sure, whether φ_1 is larger than φ_2 by 210°, or whether φ_1 is smaller than φ_2 by 150°. So, we only can reliably compare continuous changes to the phase. In the case of comparing the positioning of a zero in the left or right complex semiplane, we can say that in one case the phase will be growing and in the other it will be decreasing.

If all zeros are in the left semiplane, then the phase will be increasing as much as possible, the total contribution of all zeros to the phase variation on $\omega \in (-\infty, +\infty)$ being equal to $+180^\circ \cdot N_z$. If all zeros are in the right semiplane, then the phase will be decreasing as much as possible, the total contribution being $-180^\circ \cdot N_z$. Assuming the filter is stable, all its poles are in the left semiplane. The factors corresponding to the poles are contained in the denominator of the transfer function, therefore left-semiplane poles contribute to the decreasing of the phase, the total contribution being $-180^\circ \cdot N_p$.

If all zeros are positioned in the left semiplane, the total phase variation is $-180^\circ \cdot (N_p - N_z)$. If all zeros are positioned in the right semiplane, the total phase variation is $-180^\circ \cdot (N_p + N_z)$. Since $0 \le N_z \le N_p$, the absolute total phase variation in the second case is as large as possible, whereas in the first case it is as small as possible. For that reason the filters and/or transfer functions having all zeros in the left semiplane are referred to as *minimum phase*, and respectively the filters and/or transfer functions having all zeros in the right semiplane are referred to as *maximum phase*.[20]

2.10 LP to HP substitution

The symmetry between the lowpass and the highpass 1-pole amplitude responses has an algebraic explanation. The 1-pole highpass transfer function can be obtained from the 1-pole lowpass transfer function by the *LP to HP* (lowpass to highpass) *substitution*:

$$s \leftarrow 1/s$$

Applying the same substitution to a highpass 1-pole we obtain a lowpass 1-pole. The name "LP to HP substitution" originates from the fact that a number of filters are designed as lowpass filters and then are being transformed to their highpass versions. Occasionally we will also refer to the LP to HP substitution as *LP to HP transformation*, where essentially there won't be a difference between the two terms.

Recalling that $s = j\omega$, the respective transformation of the imaginary axis is $j\omega \leftarrow 1/j\omega$ or, equivalently

$$\omega \leftarrow -1/\omega$$

Recalling that the amplitude responses of real systems are symmetric between positive and negative frequencies ($|H(j\omega)| = |H(-j\omega)|$) we can also write

$$\omega \leftarrow 1/\omega \qquad \text{(for amplitude response only)}$$

[20] The only filter which we discussed so far which was having a zero was the 1-pole highpass. It has the zero right on the imaginary axis and thus we can't really say whether it's minimum or maximum phase or "something in between". However later we will encounter some filters with zeros located off the imaginary axis and in some cases the choice between minimum and maximum phase will become really important.

Taking the logarithm of both sides gives:

$$\log \omega \leftarrow -\log \omega \qquad \text{(for amplitude response only)}$$

Thus, the amplitude response is flipped around $\omega = 1$ in the logarithmic scale.

The LP to HP substitutions also transforms the filter's poles and zeros by the same formula:

$$s' = 1/s$$

where we substitute pole and zero positions for s. Clearly this transformation maps the complex values in the left semiplane to the values in the left semiplane and the values in the right semiplane to the right semiplane. Thus, the LP to HP substitution exactly preserves the stability of the filters.

Notice that thereby a zero occuring at $s = 0$ will be transformed into a zero at the infinity and vice versa (this is the main example of why we sometimes need to consider zeros at the infinity). Particularly, the zero at $s = \infty$ of the 1-pole lowpass filter is transformed into the zero at $s = 0$ of the 1-pole highpass filter.

The LP to HP substitution can be performed not only algebraically (on a transfer function), but also directly on a block diagram, if we allow the usage of differentiators. Since the differentiator's transfer function is $H(s) = s$, replacing all integrators by differentiators will effectively perform the $1/s \leftarrow s$ substitution, which apparently is the same as the $s \leftarrow 1/s$ substitution. Shall the usage of the differentiators be forbidden, it might still be possible to convert differentiation to the integration by analytical transformations of the equations expressed by the block diagram.

2.11 Multimode filter

Actually, we can pick up the lowpass and highpass signals simultaneously from the same structure (Fig. 2.13). This is referred to as a *multimode filter*.

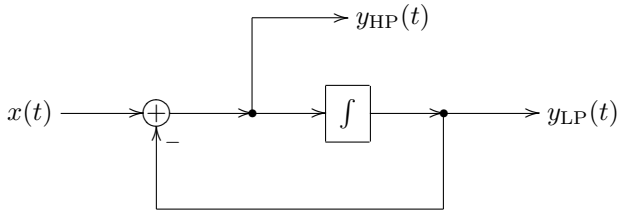


Figure 2.13: A 1-pole multimode filter.

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

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

The multimode filter can be used to implement almost any 1st-order stable differential filter by simply mixing its outputs. Indeed, let

$$H(s) = \frac{b_1 s + b_0}{s + a_0}$$

where we assume $a_0 \neq 0$.[21] Letting $\omega_c = a_0$ we obtain

$$H(s) = \frac{b_1 s + b_0}{s + \omega_c} = b_1 \frac{s}{s + \omega_c} + \frac{b_0}{\omega_c} \cdot \frac{\omega_c}{s + \omega_c} = b_1 H_{\mathrm{HP}}(s) + \left(\frac{b_0}{\omega_c}\right) H_{\mathrm{LP}}(s)$$

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

$$y = b_1 y_{\mathrm{HP}}(t) + \left(\frac{b_0}{\omega_c}\right) y_{\mathrm{LP}}(t)$$

as the output signal.

Normally (although not always) we are interested in the filters whose responses do not change the shape under cutoff variation, but are solely shifted to the left or to the right in the logarithmic frequency scale. Such modal mixtures are easiest written in the unit-cutoff form:

$$H(s) = \frac{b_1 s + b_0}{s + 1} = b_1 \frac{s}{s + 1} + b_0 \frac{1}{s + 1}$$

where we actually imply

$$H(s) = \frac{b_1 (s/\omega_c) + b_0}{(s/\omega_c) + 1}$$

Respectively, the mixing coefficients become independent of the cutoff:

$$y = b_1 y_{\mathrm{HP}}(t) + b_0 y_{\mathrm{LP}}(t)$$

Fig. 2.14 illustrates.

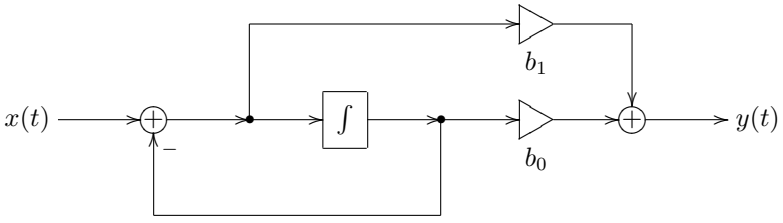


Figure 2.14: Modal mixture with 1-pole multimode filter implementing $H(s) = (b_1 s + b_0)/(s + 1)$.

[21] If $a_0 = 0$, it means that the pole of the filter is exactly at $s = 0$, which is a rather exotic situation to begin with. Even then, chances are that $b_0 = 0$ as well, in which case the filter either reduces to a multiplication by a gain ($H(s) = b_1$) or, if the coefficients vary, we can take the limiting value of b_0/ω_c in the respective formulas.

2.12 Shelving filters

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

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

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

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

The amplitude response is plotted Fig. 2.15. Typically $K \geq -1$. At $K = 0$ the signal is unchanged. At $K = -1$ the filter turns into a highpass.

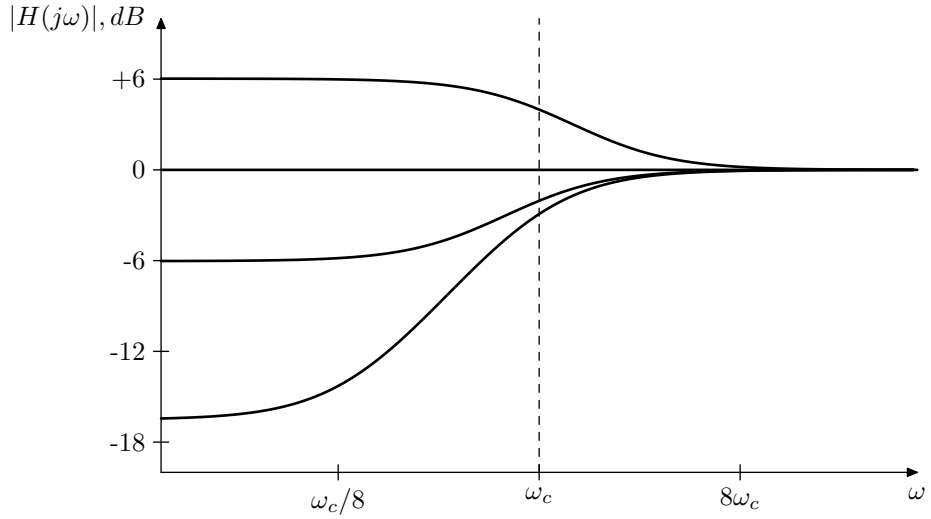


Figure 2.15: Amplitude response of a 1-pole low-shelving filter (for various K).

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

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

and

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

The amplitude response is plotted Fig. 2.16.

Actually, it would be more convenient to specify with the fact that the amplitude boost or drop for the "shelf" in decibels. It's not difficult to realize that the decibel boost is

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

Indeed, e.g. for the low-shelving filter at $\omega = 0$ (that is $s = 0$) we have[22]

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

[22] $H(0) = 1 + K$ 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.

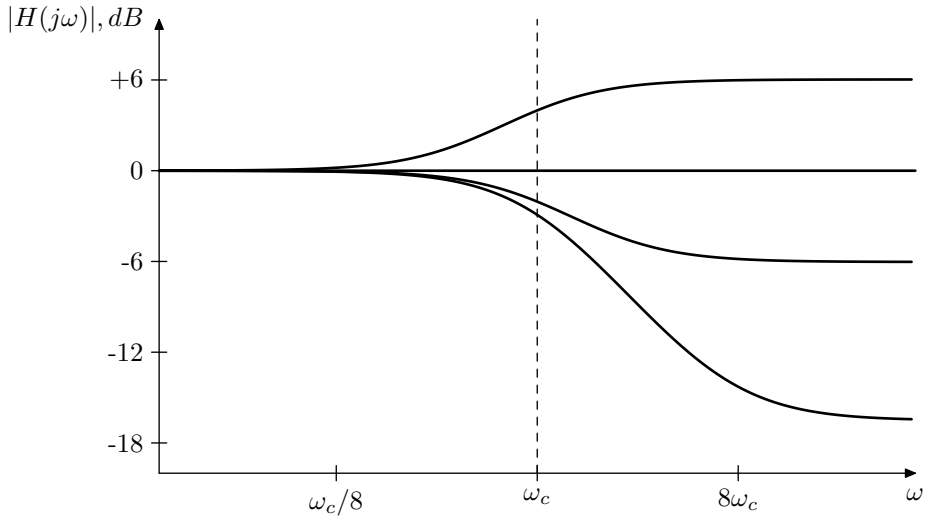


Figure 2.16: Amplitude response of a 1-pole high-shelving filter (for various K).

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

There is, however, a problem with the shelving filters built this way. Even though these filters do work as a shelving filters, the definition of the cutoff at $\omega = 1$ for such filters is not really convenient. Indeed, looking at the amplitude response graphs in Figs. 2.15 and 2.16 we would rather wish to have the cutoff point positioned exactly at the middle of the respective slopes. A solution to this problem will be described in Chapter 10.

2.13 Allpass filter

The ideas explained in the discussion of the minimum and maximum phase properties of a filter can be used to construct an allpass filter. Since in this chapter our focus is on 1-poles, we will construct a 1-pole allpass but the same approach generalizes to an allpass of an arbitrary order.

Starting with an identity 1-pole transfer function

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

and noticing that this is a minimum phase filter, let's flip its zero symmetrically with respect to the imaginary axis, thereby turning it into a maximum phase filter:

$$H(s) = \frac{s-1}{s+1} \tag{2.11}$$

As we discussed before, such change can't affect the amplitude response of the filter and thus

$$|H(j\omega)| = \left|\frac{j\omega - 1}{j\omega + 1}\right| \equiv 1$$

On the other hand the phase response has changed from $\arg H(j\omega) \equiv 0$ to some decreasing function of ω (Fig. 2.17).

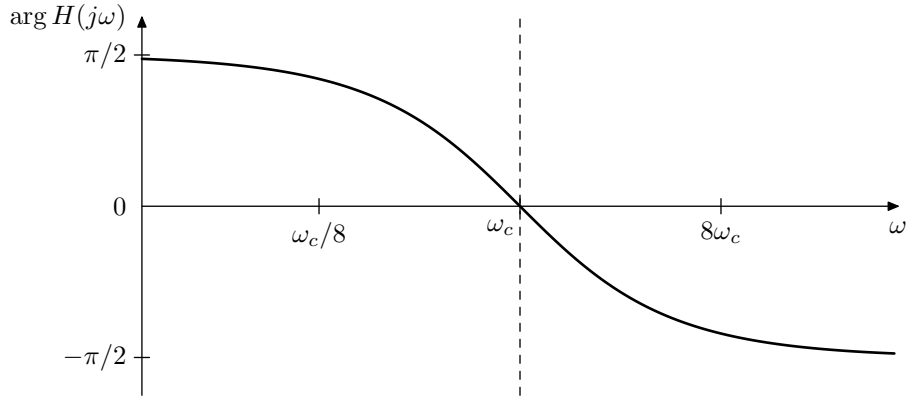


Figure 2.17: Phase response of the 1-pole allpass filter (2.11).

The filters whose purpose is to affect only the phase of the signal, not touching the amplitude part at all, are referred to as *allpass filters.*[23] Obviously, (2.11) is a 1-pole allpass. However it's not the only possible one.

Apparently, multiplying a transfer function by -1 doesn't change the amplitude response. Therefore, multiplying the right-hand side of (2.11) by -1 we obtain another 1-pole allpass.

$$H(s) = \frac{1-s}{1+s} \tag{2.12}$$

This one differs from the one in (2.11) by the fact that the phase response of (2.12) is changing from 0 to $-\pi$ (Fig. 2.18) whereas the phase of (2.11) is changing from $+\pi/2$ to $-\pi/2$. Often it's more convenient, if the allpass filter's phase response starts at zero, which could be a reason for preferring (2.12) over (2.11).

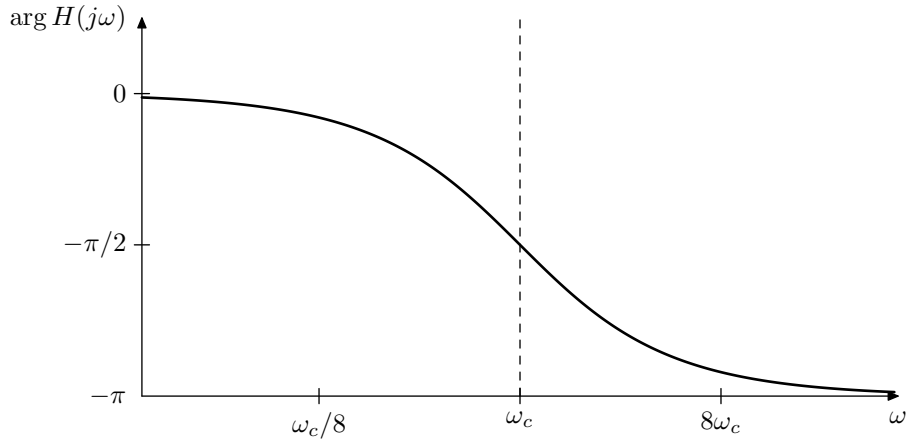


Figure 2.18: Phase response of the 1-pole allpass filter (2.12).

[23] The most common VA use for the allpass filters is probably in phasers.

Notably, the phase response of the allpass (2.12) (Fig. 2.18) is the doubled phase response of the 1-pole lowpass (Fig. 2.7). It is easy to realize that the reason for this is that the numerator $(1-s)$ contributes exactly the same amount to the phase response as the denominator $(1+s)$:

$$\arg\frac{1-j\omega}{1+j\omega} = \arg(1-j\omega) - \arg(1+j\omega) = -2\arg(1+j\omega) = -2\arctan\omega \quad (2.13)$$

where the symmetry of the phase response in Fig. 2.18 is due to (2.9).

Noticing that

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

we find that the allpass (2.12) can be obtained by simply subtracting the highpass output from the lowpass output of the multimode filter, the opposite order of subtraction creating the (2.11) allpass.

As mentioned earlier, the same approach can in principle be used to construct arbitrary allpasses. Starting with a stable filter

$$H(s) = \frac{\prod_{n=1}^{N}(s-p_n)}{\prod_{n=1}^{N}(s-p_n)} \equiv 1$$

we flip all zeros over to the right complex semiplane, turning $H(s)$ into a maximum phase filter:

$$H(s) = \frac{\prod_{n=1}^{N}(s+p_n^*)}{\prod_{n=1}^{N}(s-p_n)}$$

where we might invert the result to make sure that $H(0) = 1$

$$H(s) = (-1)^N \cdot \frac{\prod_{n=1}^{N}(s+p_n^*)}{\prod_{n=1}^{N}(s-p_n)}$$

In practice, however, high order allpasses are often created by simply connecting several of 1- and 2-pole allpasses in series.

2.14 Transposed multimode filter

We could apply the *transposition* to the block diagram in Fig. 2.13. The transposition process is defined as reverting the direction of all signal flow, where

forks turn into summators and vice versa (Fig. 2.19).[24] 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. 2.19 we have a lowpass and a highpass input and a single output.

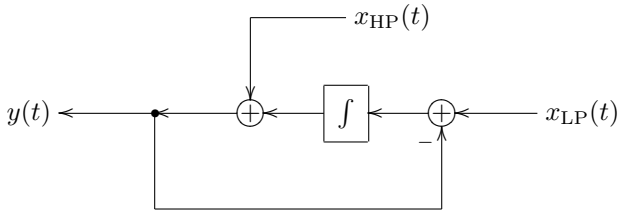


Figure 2.19: A 1-pole transposed multimode filter.

Looking carefully at Fig. 2.19 we would notice that the lowpass part of the structure is fully identical to the non-transposed lowpass. The highpass part differs solely by the relative order of the signal inversion and the integrator in the feedback loop. It might seem therefore that the ability to accept multiple inputs with different corresponding transfer functions is the only essential difference of the transposed filter from the non-transposed one.

This is not fully true, if time-varying usage of the filter is concerned. Note that if the modal mixture is involved, the gains corresponding to the transfer function numerator coefficients will precede the filter (Fig. 2.20). Thus, if the mixing coefficients vary with time, the coefficient variations will be smoothed down by the filter (especially the lowpass coefficient, but also to an extent the highpass one), in a similar way to how the cutoff placement prior to the integrator helps to smooth down cutoff variations. Compare Fig. 2.20 to Fig. 2.14.

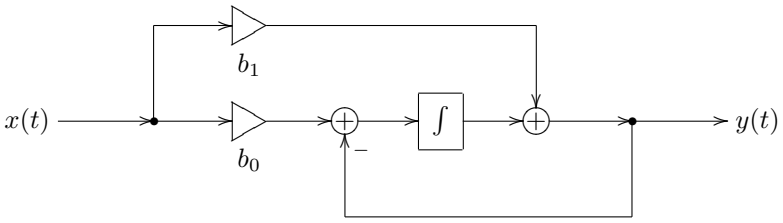


Figure 2.20: 1-pole transposed multimode filter implementing $H(s) = (b_1 s + b_0)/(s + 1)$.

One particularly useful case of the transposed 1-pole's multi-input feature, is feedback shaping. Imagine we are mixing an input signal $x_{\mathrm{in}}(t)$ with a feedback signal $x_{\mathrm{fbk}}(t)$, 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-

[24] The inverting input of the summator in the transposed version was obtained from the respective inverting input of the summator in the non-transposed 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.

pole multimode filter as in Fig. 2.21. The mixing coefficients A, B, C and D define the numerators of the respective two transfer functions.

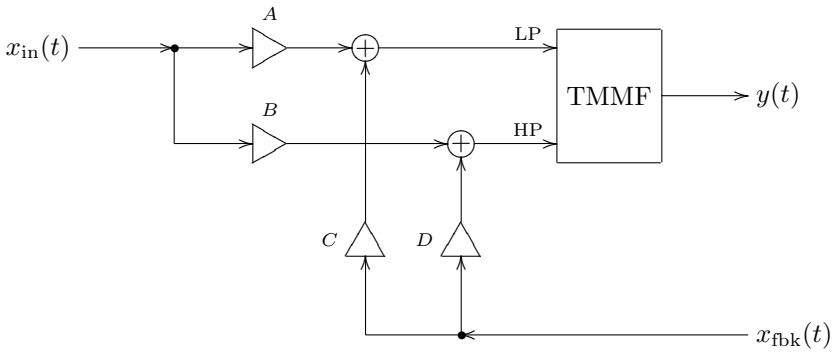


Figure 2.21: A transposed multimode filter (TMMF) used for feedback signal mixing.

2.15 Transient response

For a 1-pole filter it is not difficult to obtain an explicit expression for the filter's output, given the filter's input. Indeed, let's rewrite (2.2) in terms of ω_c:

$$\dot{y}(t) = \omega_c \cdot (x(t) - y(t))$$

We can further express ω_c in terms of the system pole $p = -\omega_c$:

$$\dot{y} = p \cdot (y - x) \tag{2.14}$$

Writing the system equation in terms of the pole will prove to be useful, when we reuse the results obtained in this section in later chapters of the book.

Rewriting (2.14) in a slightly different way we obtain

$$\dot{y} - py = -px \tag{2.15}$$

Multiplying both sides by e^{-pt}:

$$e^{-pt}\dot{y} - pe^{-pt}y = -pe^{-pt}x$$

and noticing that the left-hand size is a derivative of $e^{-pt}y(t)$ we have

$$\frac{\mathrm{d}}{\mathrm{d}t}(e^{-pt}y) = -pe^{-pt}x$$

Integrating both sides from 0 to t with respect to t:

$$e^{-pt}y(t) - y(0) = -p\int_0^t e^{-p\tau}x(\tau)\,\mathrm{d}\tau$$

$$e^{-pt}y(t) = y(0) - p\int_0^t e^{-p\tau}x(\tau)\,\mathrm{d}\tau$$

Multiplying both sides by e^{pt}:

$$y(t) = y(0)e^{pt} - p\int_0^t e^{p(t-\tau)}x(\tau)\,\mathrm{d}\tau \tag{2.16}$$

we obtain a formula which allows us to *explicitly* compute the filter's output, knowing the filter's input and initial state.

Now suppose $x(t) = X(s)e^{st}$. Then (2.16) implies

$$\begin{aligned}
y(t) &= y(0)e^{pt} - pe^{pt}X(s)\int_0^t e^{(s-p)\tau}\,\mathrm{d}\tau = \\
&= y(0)e^{pt} - pe^{pt}X(s)\cdot\frac{e^{(s-p)\tau}}{s-p}\bigg|_{\tau=0}^{t} = \\
&= y(0)e^{pt} - pe^{pt}X(s)\cdot\frac{e^{(s-p)t}-1}{s-p} = \\
&= \left(y(0) - \frac{-p}{s-p}X(s)\right)e^{pt} + \frac{-p}{s-p}X(s)e^{st} = \\
&= \left(y(0) - H(s)X(s)\right)e^{pt} + H(s)X(s)e^{st} = \\
&= \big(y(0) - H(s)x(0)\big)e^{pt} + H(s)x(t) = \\
&= H(s)x(t) + \big(y(0) - H(s)x(0)\big)e^{pt}
\end{aligned} \tag{2.17}$$

where

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

is the filter's transfer function.

Now look at the last expression of (2.17). The first term corresponds to (2.6). This is the output of the filter which we would expect according to our previous discussion. The second term looks new, but, since normally $p < 0$, this term is exponentially decaying with time. Thus at some moment the second term becomes negligible and only the first term remains. We say that the filter has entered a *steady state* and refer to $H(s)x(t)$ as the *steady-state response* of the filter (for the complex exponential input signal $x(t) = X(s)e^{st}$). The other term, which is exponentially decaying and exists only for a certain period of time is called the *transient response*.

Now we would like to analyse the general case, when the input signal is a sum of such exponential signals:

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

First, assuming $y(0) = 0$ and using the linearity of (2.16), we apply (2.17) independently to each partial $X(s)e^{st}$ of $x(t)$, obtaining

$$y(t) = \int H(s)X(s)e^{st}\,\frac{\mathrm{d}s}{2\pi j} - e^{pt}\int H(s)X(s)\,\frac{\mathrm{d}s}{2\pi j} \tag{2.18}$$

Again, the first term corresponds to (2.6) and is the steady-state response. Respectively, the second term, which is exponentially decaying (notice that the

integral in the second term is simply a constant, not changing with t), is the transient response.

Comparing (2.18) to (2.16) we can realize that the difference between $y(0) = 0$ and $y(0) \neq 0$ is simply the addition of the term $y(0)e^{pt}$. Thus we simply add the missing term to (2.18) obtaining

$$\begin{aligned} y(t) &= \int H(s)X(s)e^{st}\,\frac{\mathrm{d}s}{2\pi j} + \left(y(0) - \int H(s)X(s)\,\frac{\mathrm{d}s}{2\pi j}\right)\cdot e^{pt} = \\ &= y_s(t) + (y(0) - y_s(0))\cdot e^{pt} = y_s(t) + y_t(t) \end{aligned} \tag{2.19}$$

where

$$y_s(t) = \int H(s)X(s)e^{st}\,\frac{\mathrm{d}s}{2\pi j} \tag{2.20a}$$

$$y_t(t) = (y(0) - y_s(0))\cdot e^{pt} \tag{2.20b}$$

are the steady-state and transient responses.

Looking at (2.20) we can give the following interpretation to the steady-state and transient responses. Steady-state response is the "expected" response of the filter in terms of the spectrum of $x(t)$ and the transfer function $H(s)$, this is the part of the filter's output that we have been exclusively dealing with until now and this is the part that we will continue being interested in most of the time. Particularly, this is the part of the filter's output for which the terms amplitude and phase response are making sense. However, at the initial time moment the filter's output will usually not match the expected response ($y(0) \neq y_s(0)$), since the initial filter state may be arbitrary. Even if $y(0) = 0$, we still usually have $y_s(0) \neq 0$. But the integrator's state cannot change abruptly[25] and therefore there will be a difference between the actual and "expected" outputs. This difference however decays exponentially as e^{pt}. This exponentially decaying part, caused by a discrepancy between the "expected" output and the actual state of the filter is the transient response (Fig. 2.22).

The origin of the term "steady-state response" should be obvious by now. As for the term "transient response" things might be a bit more subtle, but actually it's also quite simple.

Suppose the input of the filter is receiving a steady signal, e.g. a periodic wave and suppose the filter has entered the steady state by $t = t_0$ (meaning that the transient response became negligibly small). Suppose that at $t = t_0$ a *transient* occurs in the input signal: the filter's input suddenly changes to some other steady signal, e.g. it has a new waveform, or amplitude, or frequency, or all of that. This means that at this moment the definition of the steady state also changes and the filter's output does no longer match the "expected" signal. Thus, at $t = t_0$ we suddenly have $y_s(t) \neq y(t)$ and a decaying transient response impulse is generated. The transient response turns a sudden jump, which would have occured in the filter's output due to the switching of the input signal, into a continuous exponential "crossfade".

[25] Assuming the input signal is finite. In theoretical filter analysis sometimes infinitely large input signals (most commonly $x(t) = \delta(t)$) are used. In such cases the filter state may change abruptly (and this is the whole purposes of using input signals such as $\delta(t)$).

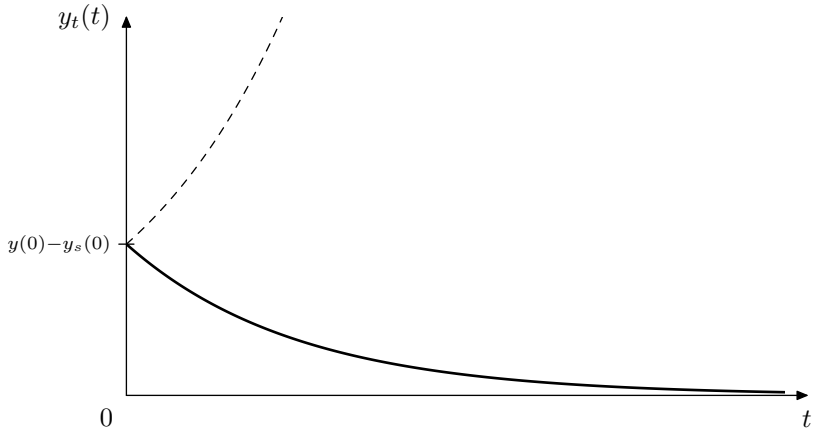


Figure 2.22: Transient response of a 1-pole lowpass filter (dashed line depicts the unstable case).

Highpass transient response

For a highpass, since $y_{\mathrm{HP}}(t) = x(t) - y_{\mathrm{LP}}(t) = x(t) - y$, equation (2.19) converts into

$$y_{\mathrm{HP}}(t) = x(t) - (y_s(t) + y_t(t)) = (x(t) - y_s(t)) - y(t) = y_{\mathrm{HPs}}(t) + y_{\mathrm{HPt}}(t)$$

where the highpass steady-state response is

$$\begin{aligned} y_{\mathrm{HPs}}(t) = x(t) - y_s(t) &= x(t) - \int H(s)X(s)e^{st}\,\frac{\mathrm{d}s}{2\pi j} = \\ &= \int X(s)e^{st}\,\frac{\mathrm{d}s}{2\pi j} - \int H(s)X(s)e^{st}\,\frac{\mathrm{d}s}{2\pi j} = \\ &= \int (1 - H(s))X(s)e^{st}\,\frac{\mathrm{d}s}{2\pi j} = \int H_{\mathrm{HP}}(s)X(s)e^{st}\,\frac{\mathrm{d}s}{2\pi j} \end{aligned}$$

and the highpass transient response is

$$\begin{aligned} y_{\mathrm{HPt}}(t) &= -y_t(t) = -\left(y(0) - y_s(0)\right)\cdot e^{pt} = \\ &= \left((x(t) - y(0)) - (x(t) - y_s(0))\right)\cdot e^{pt} = (y_{\mathrm{HP}}(0) - y_{\mathrm{HPs}}(0))\cdot e^{pt} \end{aligned}$$

That is we are having the same kind of exponentially decaying discrepancy between the output signal and the steady-state signal, where the exponent e^{pt} itself is identical to the one in the lowpass transient response.

Poles and stability

At this point we could get a first hint at the mechanism behind the relationship between the filter poles and filter stability. The transient response of the 1-pole filter decays as e^{pt} (this means it it takes longer time to reach a steady state at lower cutoffs). However, if $p > 0$, the transient response doesn't decay, but instead infinitely grows with time (as shown by the dashed line in Fig. 2.22), and we say that the filter "explodes".

At $p = 0$ the 1-pole lowpass filter doesn't explode, but stays at the same value (since $p = 0$ implies $\dot{y} = 0$ for this filter), corresponding to the marginally stable case. But this actually happens because of the specific form of the transfer function we are using: $H(s) = -p/(s - p)$. Thus, $p = 0$ simultaneously implies a zero total gain, which prevents the explosion.

However, in a more general case, a marginally stable 1-pole filter can explode. We are going to discuss this using Jordan 1-poles.

Steady state

The steady-state response is actually not a precisely defined concept, as it has a subjective element. A bit earler we have been analysing the situation of an abrupt change of the input signal causing a discrepancy between the steady-state response and the actual output signal, this discrepancy being responsible for the appearance of the transient response term. However we don't have to understand this case as an abrupt change of the input signal. Instead we could consider the input signal over the entire time duration as a whole incorporating the abrupt change as an integral part of the signal. E.g. instead of considering the input signal changing from $\sin t$ to $2\sin(4t + 1)$ at some moment $t = t_0$, we would formally consider a non-periodic signal $x(t)$ defined as

$$x(t) = \begin{cases} \sin t & \text{if } t < t_0 \\ 2\sin(4t + 1) & \text{if } t \geq t_0 \end{cases}$$

In that sense there would be just some non-periodic input signal $x(t)$ which doesn't change to some other input signal. Then we would have a different definition of the signal's spectrum, the spectrum being constant all the time, rather than suddenly changing at $t = t_0$, which would mean there is no transient at $t = t_0$. Thus we would also be having a different definition of the steady state response, which wouldn't have a discrepancy with the filter's output signal at $t = t_0$ either. Therefore there wouldn't be a transient response impulse appearing at $t = t_0$. Thus, the definition of the input signal has a subjective element, which results in the same subjectivity of the definition of the steady-state response signal.

The formal definition of the steady-state response is the formula (2.20a). Careful readers who are also familiar with Laplace transform theory might be by now asking themselves the question, whether the multiplication of $X(s)$ by $H(s)$ has any effect on the region of convergence and, if yes, what are the implications of this effect. Surprisingly, this question has a connection to the subjectivity of the steady-state response.

The thing is that due to the subjectivity of the steady-state response, we don't care too much about what the Laplace integral in (2.20a) converges to. Most importantly, it does converge. And normally it will converge for any $\operatorname{Re} s$ (with some additional care being taken in evaluation of (2.20a) if the integration path $\operatorname{Re} s = \text{const}$ contains some poles). It's just that as we horizontally shift the integration path $\operatorname{Re} s = \text{const}$, and this path is thereby traversing through the poles of $H(s)X(s)$, the integral (2.20a) will converge to some other function, but it will converge nevertheless. In fact we even cannot say what the Laplace transform's region of convergence for (2.20a) is. We could say what the region of convergence is for $X(s)$, since we have the original signal $x(t)$, but we cannot

say what is the region of convergence for $H(s)X(s)$, since its original signal would be $y_s(t)$ and we don't have an exact definition of the latter.

Therefore we actually could choose which of the different resulting signals delivered by (2.20a) (for different choices of the "region of convergence" of $H(s)X(s)$) to take as the steady-state response. For one, we probably shouldn't go outside of the region of convergence of $X(s)$, since otherwise we would have a different input signal and the result would be simply wrong. However, other than that we have total freedom. Given that all poles (or actually, the only pole, since so far $H(s)$ is a 1-pole) of $H(s)$ are located to the left of the imaginary axis (which is the case for the stable filter), it probably makes most sense to choose the range of $\operatorname{Re} s$ containing the imaginary axis as the region of convergence of $H(s)X(s)$, because $H(s)$ evaluated on the imaginary axis gives the amplitude and phase responses and thus the steady-state response definition will be in agreement with amplitude and phase responses.

What shall we do, however, if $\operatorname{Re} p > 0$ (where p is the pole of $H(s)$), that is $H(s)$ is unstable? First, let's notice that as we change the integration path in (2.20a) from $\operatorname{Re} s < p$ to $\operatorname{Re} s > p$ the integral (2.20a) changes exactly by the residue of $H(s)X(s)e^{st}$ at $s = p$ (it directly follows from the residue theorem). But this residue is simply

$$\operatorname*{Res}_{s=p} \left(H(s)X(s)e^{st}\right) = \operatorname*{Res}_{s=p} \left(\frac{a}{s-p} \cdot X(s)e^{st}\right) = aX(p)e^{pt} \qquad (\text{where } a = -p)$$

Therefore the steady state response $y_s(t)$ defined by the integral (2.20a) is changing by a term of the form $aX(p)e^{pt}$, which is then added to or subtracted from the transient response to keep the sum $y(t)$ unchanged. But the transient response already consists of a similar term, just with a different amplitude. Thus the change from $\operatorname{Re} s < p$ to $\operatorname{Re} s > p$ simply changes the transient response's amplitude. Therefore, there is not much difference, whether in the unstable case we evaluate (2.20a) for e.g. $\operatorname{Re} s = 0$ or for some $\operatorname{Re} s > p$. It might therefore be simply more consistent to always evaluate it for $\operatorname{Re} s = 0$, regardless of the stability, but, as we just explained, this is not really a must.

Note that thereby, even though amplitude and phase responses make no sense for unstable filters, the equation (2.20a) still applies, therefore the transfer function $H(s)$ itself makes total sense regardless of the filter stability.

Jordan 1-pole

For the purposes of theoretical analysis of systems of higher order it is sometimes helpful to use 1-poles where the input signal is not multiplied by the cutoff $-p$:

$$\dot{y} = py + x \tag{2.21}$$

(Fig. 2.23). We also allow p to take complex values. Such 1-poles are the building elements of the state-space diagonal forms and of the so-called *Jordan chains*. For that reason we will refer to (2.21) as a *Jordan 1-pole*.

One could argue that there is not much difference between the 1-pole equations (2.14) and (2.21) and respectively between Fig. 2.2 and Fig. 2.23, since one could always represent the Jordan 1-pole via the ordinary 1-pole lowpass by dividing the input signal of the latter by the cutoff. Also it would be no problem to allow p to take complex values in (2.14). This approach however

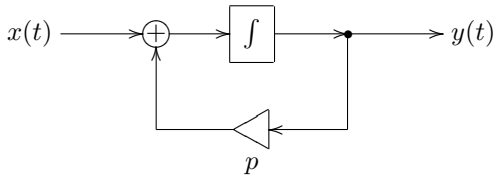


Figure 2.23: Jordan 1-pole. Note that the integrator is not supposed to internally contain the implicit cutoff gain!

won't work if $p = 0$. For that reason, in certain cases it is more conveninent to use a Jordan 1-pole instead.

Changing from (2.14) to (2.21) effectively takes away the $-p$ coefficient in front of x from all formulas derived from (2.14). Particularly, (2.16) turns into

$$y(t) = y(0)e^{pt} + \int_0^t e^{p(t-\tau)}x(\tau)\,\mathrm{d}\tau \tag{2.22}$$

and (2.17) turns into

$$\begin{aligned} y(t) &= y(0)e^{pt} + e^{pt}X(s)\int_0^t e^{(s-p)\tau}\,\mathrm{d}\tau = \\ &= \left(y(0) - \frac{1}{s-p}X(s)\right)e^{pt} + \frac{1}{s-p}X(s)e^{st} \end{aligned} \tag{2.23}$$

where have

$$y_s(t) = \frac{1}{s-p}X(s)e^{st} = H(s)x(t)$$

and

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

From this point on we'll continue the transient response analysis in terms of Jordan 1-poles. The results can be always converted to ordinary 1-poles by multiplying the input signal by $-p$.

Hitting the pole

Suppose the input signal of the filter is $x(t) = X(p)e^{pt}$ (where $X(p)$ is the complex amplitude). In this case (2.23) cannot be applied, because the denominator $s - p$ turns to zero and we have to compute the result differently. From (2.22) we obtain

$$y(t) = y(0)e^{pt} + X(p)\int_0^t e^{p(t-\tau)}e^{p\tau}\,\mathrm{d}\tau = y(0)e^{pt} + X(p)te^{pt} \tag{2.24}$$

Now there doesn't really seem to be a steady-state component in (2.24). The second term might look a bit like the steady-state response. Clearly it's not having the usual steady-state response form $H(p)X(p)e^{pt}$, but that would be impossible since $H(p) = \infty$. Not only that, it's not even proportional to the input signal (or, more precisely, the proportionality coefficient is equal to t,

thereby changing with time), thus not really looking like any kind of a steady state. The first term doesn't work as a steady-state response either, since it depends on the initial state of the system.

Since the idea of the steady-state response is, to an extent, subjective, it means the output which we expect from the system independently of the initial state, we could formally introduce

$$y_s(t) = X(p)te^{pt}$$

as the steady-state response in this case, thereby further transforming (2.24) as

$$y(t) = y(0)e^{pt} + Xte^{pt} = (y(0) - y_s(0))e^{pt} + y_s(t) = y_t(t) + y_s(t)$$

The benefit of this choice is that the transient response still consists of a single e^{pt} partial. The other option is letting

$$y_s(t) \equiv 0$$

which means that (2.24) entirely consists of the transient response.

In either case, the problem is that as $s \to p$ in (2.23), the steady-state response defined by $y_s(t) = H(s)X(s)e^{st}$ becomes infinitely large and we need to switch to a different steady-state response definition. Note, that there is no jump in the output signal $y(t)$, nor does $y(t)$ become infinitely large. The switching is occuring only in the way how we separate $y(t)$ into steady-state and transient parts.

We could further illustrate what is going on by a detailed evaluation of (2.23) at $s \to p$. The part which needs special attention is the integral of $e^{(s-p)\tau}$:

$$\lim_{s\to p} \int_0^t e^{(s-p)\tau}\,\mathrm{d}\tau = \lim_{s\to p} \frac{e^{(s-p)\tau}}{s-p}\bigg|_{\tau=0}^{t} = \lim_{s\to p} \frac{e^{(s-p)t} - 1}{s-p} = t$$

and thus $y(t) = y(0)e^{pt} + X(p)te^{pt}$, which matches our previous result.

In the particular case of $p = 0$ the equation (2.24) turns into

$$y(t) = y(0) + X(0)t$$

thus the marginally stable system to which Fig. 2.23 turns at $p = 0$ explodes if $s = 0$, that is if $x(t)$ is constant.[26]

Jordan chains

Fur the purposes of further analysis of transient responses of systems of higher orders it will be instructive to analyse the transient response generated by serial chains of identical Jordan 1-poles, referred to as *Jordan chains* (Fig. 2.24).

Given a complex exponential input signal $x(t) = X(s)e^{st}$, the output of the first 1-pole will have the form

$$y_1(t) = y_{s1}(t) + y_{t1}(t) = H_1(s)X(s)e^{st} + (y_1(0) - H_1(s)X(s))e^{pt}$$

where

$$H_1(s) = \frac{1}{s-p}$$

[26] It's easy to see that this system is simply an integrator.

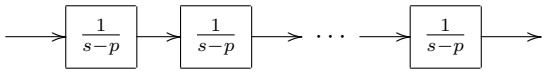


Figure 2.24: Jordan chain

The output of the second 1-pole will be therefore

$$y_2(t) = H_1^2(s)X(s)e^{st} + (y_1(0) - H_1(s)X(s))te^{pt} + (y_2(0) - H_1^2(s)X(s))e^{pt}$$

where we have used (2.23) and (2.24).

Before we obtain the output of the further 1-poles we first need to apply (2.22) to $x(t) = Xt^n e^{pt}$ yielding

$$y(t) = y(0)e^{pt} + X\frac{t^{n+1}}{(n+1)!}e^{pt}$$

Then

$$\begin{aligned} y_3(t) = H_1^3(s)X(s)e^{st} + (y_1(0) - H_1(s)X(s))\frac{t^2}{2}e^{pt} + \\ + (y_2(0) - H_1^2(s)X(s))te^{pt} + (y_3(0) - H_1^3(s)X(s))e^{pt} \end{aligned}$$

and, continuing in the same fashion, we obtain for the n-th 1-pole:

$$y_n(t) = H_1^n(s)X(s)e^{st} + \sum_{\nu=0}^{n-1}(y_{n-\nu}(0) - H_1^{n-\nu}(s)X(s))\frac{t^\nu}{\nu!}e^{pt} \tag{2.25}$$

Apparently the first term $H_1^n(s)X(s)e^{st}$ is the steady-state response whereas the remaining terms are the transient response. In principle, one could argue, that treating the remaining terms as transient response can be questioned, since we have some ambiguity in the definition of the steady-state response of the 1-poles if their poles are hit by their input signals. However, while this argument might be valid in respect to individual 1-poles, from the point of view of the entire Jordan chain all terms $t^\nu e^{pt}/\nu!$ are arising out of the mismatch between the chain's internal state and the input signal, therefore we should stick to the steady-state response definition $H_1^n(s)X(s)e^{st}$. This also matches the fact that the transfer function of the entire Jordan chain is $H_1^N(s) = 1/(s-p)^N$, where N is the number of 1-poles in the chain.

2.16 Cutoff as time scaling

Almost all analysis of the filters which we have done so far applies only to linear time-invariant filters. In practice, however, filter parameters are often being modulated. This means that the filters no longer have the time-invariant property and our analysis does not really apply. In general, the analysis of time-varying filters is a pretty complicated problem. However, in the specific (but pretty common) case of cutoff modulation there is actually a way to apply the results obtained for the time-invariant case.

Imagine a system of an arbitrary order (therefore, containing one or more integrators). Suppose the cutoff gain elements are always preceding the integrators and suppose all integrators have the same cutoff gain (that is, these gains

always have the same value, even when modulated). For each such integrator, given its input signal (which we denote as $x(t)$), its output signal is defined by

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

If cutoffs are synchronously varying with time, we could reflect this explicitly:

$$y(t) = y(t_0) + \int_{t_0}^{t} \omega_c(\tau) x(\tau)\,\mathrm{d}\tau \tag{2.26}$$

We would like to introduce a new time variable $\tilde{\tau}$ defined by

$$\mathrm{d}\tilde{\tau} = \omega_c(\tau)\,\mathrm{d}\tau$$

and respectively write

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

In order to be able to do that, we need to require that $\omega_c(t) > 0\ \forall t$, or, more precisely that there is some $\omega_0 > 0$ such that

$$\omega_c(t) \geq \omega_0 > 0 \quad \forall t$$

In an obvious analogy to the concept of uniform convergence, we could describe this requirement as "$\omega_c(t)$ being *uniformly positive*".

Apparently the requirement on $\omega_c(t)$ to be uniformly positive is quite a reasonable restriction, we just need to put a lower limit on the cutoff. Given such $\omega_c(t)$ we then can introduce the "warped" time $\tilde{t}$, where from

$$\mathrm{d}\tilde{t} = \omega_c(t)\,\mathrm{d}t$$

we obtain

$$\tilde{t} = \int \omega_c(t)\mathrm{d}t \tag{2.27}$$

E.g. we could take

$$\tilde{t} = \int_0^t \omega_c(\tau)\mathrm{d}\tau$$

Since $\omega_c(t)$ is uniformly positive, the function $\tilde{t}(t)$ is monotonically increasing and maps $t \in (-\infty, +\infty)$ to $\tilde{t} \in (-\infty, +\infty)$. We can therefore reexpress the signals $x(t)$ and $y(t)$ in terms of $\tilde{t}$, obtaining some functions $\tilde{x}\left(\tilde{t}\right)$ and $\tilde{y}\left(\tilde{t}\right)$, and ultimately

$$\tilde{y}\left(\tilde{t}\right) = \tilde{y}\left(\tilde{t}_0\right) + \int_{\tilde{t}_0}^{\tilde{t}} \tilde{x}(\tilde{\tau})\,\mathrm{d}\tilde{\tau} \tag{2.28}$$

This means that the variation of ω_c can be equivalently represented as warping of the time axis, the cutoff gains in the warped time scale having a constant unity value.[27]

[27] Instead of unit cutoff we can have any other positive value, by simply linearly stretching the time axis in addition to the warping $\tilde{t}(t)$.

Equivalent topologies

The fact that cutoff modulation can be equivalently represented as warping of the time scale has several implications of high importance. One implication has to do with equivalence of systems with different *topologies*.

The term *topology* in this context simply refers to the components used in the system's block diagram and the way they are connected to each other. Often, the reason we would want to talk about the topology would be to put it against the idea of the transfer function. More specifically: *there can be systems with different topologies implementing the same transfer function.* We have already seen the example of that: there is an ordinary 1-pole multimode and a transposed 1-pole multimode, which both can be used to implement one and the same transfer function.

According to our previous discussion, systems having identical transfer functions will behave identically (at least in the absence of the transient response arising out of a non-zero initial state of the system). However, all of our analysis of system behavior, including the transient response, was done under the assumption of time-invariance. This assumption is actually critical: for a time-varying system the situation is more complicated and two systems may behave differently even if they share the same transfer function.[28] We had a brief example of that in Section 2.7 where we compared different positionings of the cutoff gain relative to the integrator.

However (2.28) means that if the cutoff modulation is compliant to (2.26) (pre-integrator cutoff gain) and if the only time-varying aspect of the system is the cutoff modulation, the systems will behave identically. Indeed, we could use one and the same time-warping (2.27) for both of the systems, thus, if they are identically behaving in the original time-invariant case, so will they in the time-warped case.

This question will be addressed once again from a slightly more detailed point in Section 7.12.

Time-varying stability

A further implication of (2.28) is the fact that the stability of a system cannot be destroyed by the cutoff modulation. This is true for an arbitrary system, given all cutoff gains are preceding the integrators and are having equal values all the time. Indeed, the warping of the time axis (2.27) can't affect the BIBO property of the signals $x(t)$ and $y(t)$, thus stability is unaffected by the time warping.[29]

[28] Of course, strictly speaking time-varying systems do not have a transfer function. But it is intuitive to use the idea of a "time-varying transfer function", understood as the transfer function which is formally evaluated pretending the system's parameters are fixed at each time moment. E.g. if we have a 1-pole lowpass with a varying cutoff $\omega_c(t)$, we would say that its transfer function at each time moment is $H(s) = \omega_c(t)/(\omega_c(t) + s)$. Of course, this is not a true transfer function in the normal sense. Particularly, for an exponential input e^{st} the filter's output is not equal to $y(t) = H(s,t)e^{st}$.

[29] Note that this applies only to the idealized continuous-time systems. After conversion to discrete time the same argument will not automatically hold and the stability of the resulting discrete time system will need to be proven again. However, it is not unreasonable to expect, given a discretization method which preserves time-invariant stability, that it will also at least approximately preserve the time-varying stability.

For the 1-pole filter, however, the time-varying stability can be checked in a much simpler manner. As we should remember, the output signal equation for a 1-pole lowpass, written in the differential form is

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

This means that, as long as $\omega_c > 0$, the value of y always "moves in the direction towards the input signal". In this case, clearly, the absolute value of y can't exceed the maximum of the absolute value of x.[30] On the contrary, imagine $\omega_c < 0$, $x(t) = 0$ and $y(t) \neq 0$ (let's say $x(t)$ was nonzero for a while and then we switched it off, leaving $y(t)$ at a nonzero value). The differential equation turns into $\dot{y} = -\omega_c y = |\omega_c| \cdot y$, which clearly produces an indefinitely growing $y(t)$.

The 1-pole highpass filter's output is simply $x(t) - y(t)$ (where $y(t)$ is the lowpass signal), therefore the highpass filter is stable if and only if the lowpass filter is stable.

We have seen that cutoff is a very special filter parameter, such that its modulation can't destroy the filter's stability (provided some reasonable conditions are met). There are also some trivial cases, when the modulated parameters are not a part of a feedback loop, such as e.g. the mixing gain of a shelving filter. Apparently, such parameters when being varied can't destroy the filter's stability as well. With the filter types which we introduce later in this book there will be other parameters within feedback loops which in principle can be modulated. Unfortunately, for the modulation of such other parameters there is no simple answer (although sometimes the stability can be proven by some means). Respectively there is no easy general criterion for time-varying filter stability as there is for the time-invariant case. Often, we simply hope that the modulation of the filter parameters does not make the (otherwise stable) filter unstable. This is not simply a theoretical statement, on the contrary, such cases, where the modulation destabilizes a filter, do occur in practice.

SUMMARY

The analog 1-pole filter implementations are built around the idea of the multimode 1-pole filter in Fig. 2.13. The transfer functions of the lowpass and highpass 1-pole filters are

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

and

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

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

[30] If $\omega_c = 0$ then $y(t)$ doesn't change. This is the marginally stable case. Particularly, even if $x(t) = 0$, the output $y(t)$ will stay at whatever value it is, rather than decaying towards the zero.

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

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

and its Laplace integral representation is

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

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

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

(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]

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

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

$$z = e^{sT}$$

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

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 $X(z)z^n$.

[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.

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.[5]

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 integrator state and simultaneously serve as the output sample. As we begin to process the next input sample, 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.

[5] Based on the fact that the naive integration introduced above is identical to Euler backward-difference integration, there is an opinion that the naive approach (loosely defined as "take whatever values we have now at the integrator inputs and apply a single naive integration step to those") is identical to the Euler method. This is not 100% so. The readers are encouraged to formally apply backward- and forward-difference Euler methods to $\dot{y} = \omega_c(y - x)$ to convince themselves that there are some differences. Particularly, the backward-difference method is implicit (requires solving an equation), while the forward-difference method produces the "future" value of the output. For more complicated systems the differences could be more drastic, although the author didn't explicitly verify that.

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. 2.13 could have been implemented as:

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

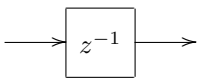
where the integrator state is stored in the *lp* variable.

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.

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

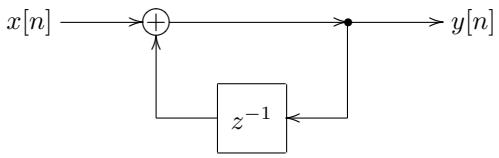


Figure 3.1: Naive integrator for $T = 1$.

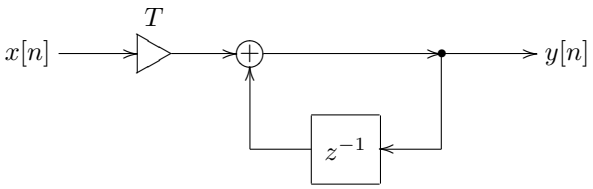


Figure 3.2: Naive integrator for arbitrary T.

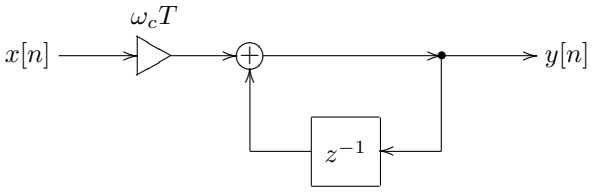


Figure 3.3: Naive integrator with embedded cutoff.

```
// 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 $x - y$.

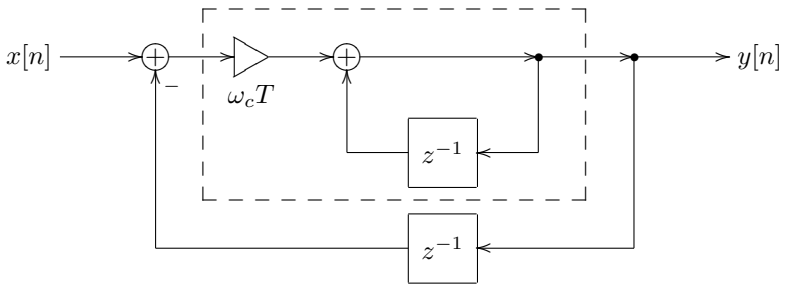


Figure 3.4: Naive 1-pole lowpass filter (the dashed line denotes the integrator).

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).

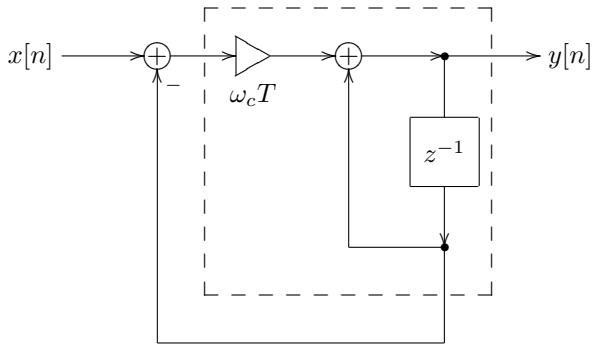
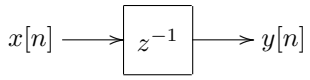


Figure 3.5: Naive 1-pole lowpass filter with just one z^{-1} element (the dashed line denotes the integrator).

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[6] 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[7] $x[n] = X(z)z^n$ and $y[n] = Y(z)z^n$, or shortly, $x = X(z)z^n$ and

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

[7] 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.

$y = Y(z)z^n$. Then the output of the z^{-1} element is yz^{-1}. The output of the summator is then $x + yz^{-1}$, thus

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

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.[8]

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, 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

$$\left(1 - (1 - \omega_c)z^{-1}\right)y = \omega_c x$$

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.

In principle, the amplitude response deviation can be drastically reduced by correcting the filter's cutoff setting. E.g. one could notice that the second

[8] 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 $z = e^{j\omega}$.

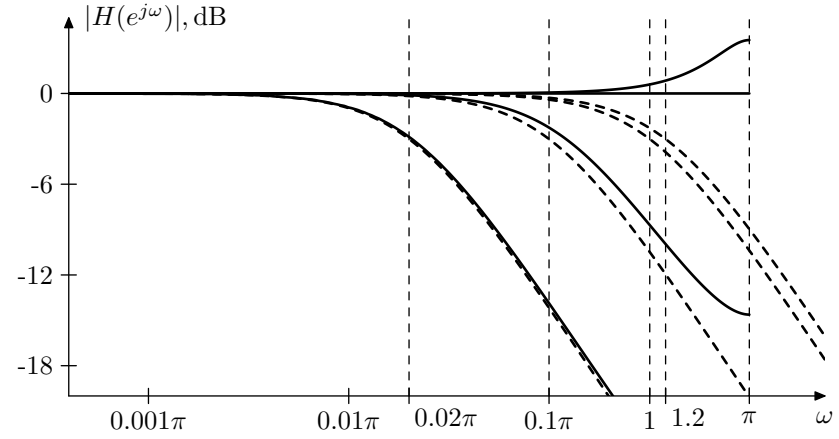


Figure 3.6: Amplitude response of a naive 1-pole lowpass filter for a number of different cutoffs. Dashed curves represent the respective analog filter responses for the same cutoffs.

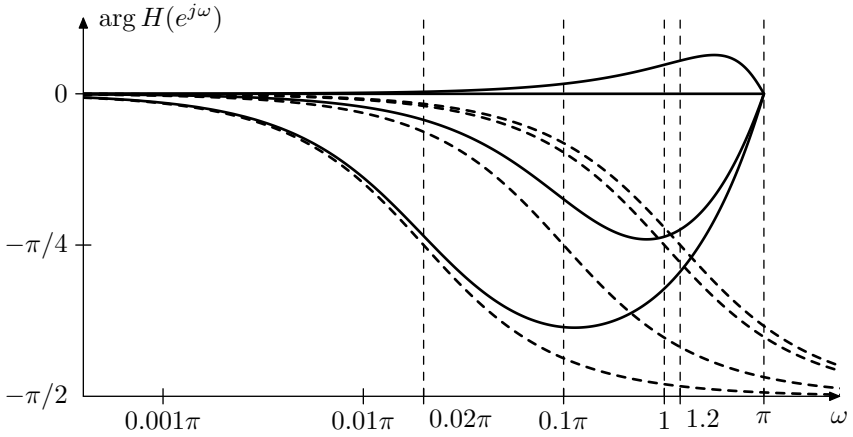


Figure 3.7: Phase response of a naive 1-pole lowpass filter for a number of different cutoffs. Dashed curves represent the respective analog filter responses for the same cutoffs.

of the amplitude responses in Fig. 3.6 is occurring a bit too far to the right, compared to the analog response (which is what we're aiming at). Therefore we could achieve a better matching between the two responses by reducing the cutoff setting of the digital filter by a small amount. Depending on the formal definition of the response matching, one could derive an analytical expression for such cutoff correction. There are two main problems with that, though.

One problem is that many other filters, e.g. a 2-pole resonating lowpass, have more parameters, e.g. not only cutoff but also the resonance, and we potentially may need to correct all of them, which results in much more involved math. This problem is not as critical though, and there are some methods utilizing this approach.

The other problem, though, is the phase response. Looking at Fig. 3.7 it

seems that no matter how we try to correct the filter's cutoff, the phase response will be always zero at Nyquist, whereas we would desire something close to $-\pi/2$. The effects of the deviation of the filter's phase response are mostly quite subtle. Therefore it's somewhat difficult to judge how critical the phase devations might be.[9] However there's one absolutely objective and major issue associated with the phase deviations. Attempting to mix outputs of two filters with some deviations in either or both of the amplitude and phase responses may easily lead to unexpected and undesired results. For that reason in this book we will concentrate on a different method which is much more robust in this respect.

Poles and zeros

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.

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. This is indeed the case, a linear time-invariant difference system[10] is stable if and only if all its poles are located inside the unit circle. We will give more detail about the mechanisms behind this in the discussion of the discrete-time transient response in Sections 3.12 and 7.13.

3.6 Trapezoidal integration

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

```
// 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.

[9] Many engineers seem to believe that the deviations in phase response are quite tolerable acoustically. The author's personal preference is to be on the safe side and not take the risks which are difficult to estimate. At least some caution in this regard would be recommended.

[10] 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.

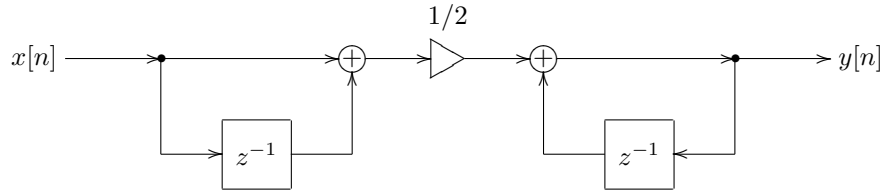


Figure 3.8: Direct form I trapezoidal integrator ($T = 1$).

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.

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

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

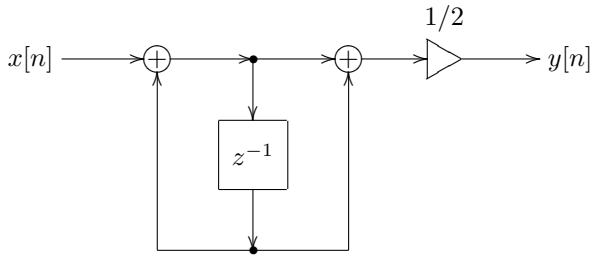


Figure 3.9: Direct form II (canonical) trapezoidal integrator ($T = 1$).

Notice that

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

and

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

Expressing (3.3a) and (3.3b) in a graphical form, we obtain the structure in Fig. 3.10. 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.

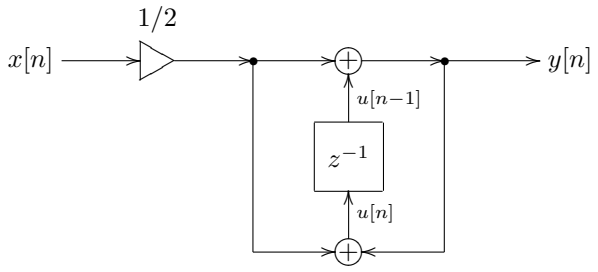


Figure 3.10: Transposed direct form II (transposed canonical) trapezoidal integrator ($T = 1$).

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.

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.

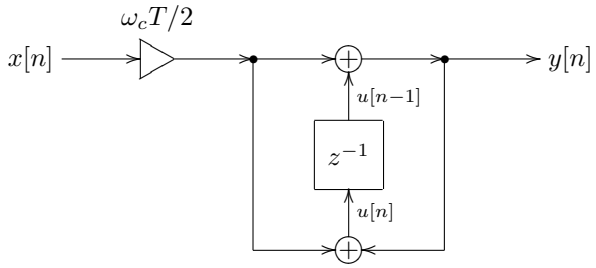


Figure 3.11: Transposed direct form II (transposed canonical) trapezoidal integrator with "embedded" cutoff gain.

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). Assuming signals of the exponential form z^n, we can drop the index $[n]$, understanding it implicity, while the index $[n-1]$ will be replaced by the multiplication by z^{-1}. Then (3.3) turn into

$$y = uz^{-1} + \frac{x}{2}$$
$$u = y + \frac{x}{2}$$

Substituting the second equation into the first one we have

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

$$yz = y + \frac{x}{2} + \frac{x}{2}z$$

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

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.

3.7 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 substitution

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

in the transfer function expression.

It would be more convenient to write this substitution explicitly as

$$s = \frac{2}{T} \cdot \frac{z-1}{z+1} \tag{3.4}$$

The substitution (3.4) 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

$$H_d(z) = H_a\left(\frac{2}{T} \cdot \frac{z-1}{z+1}\right) \tag{3.5}$$

Respectively, the discrete-time frequency response is

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

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 $0 \le \omega < +\infty$ is mapped into the digital frequency range $0 \le \omega T < \pi$ ($0 \le \omega < \pi f_s$), that is from zero to Nyquist![11] 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

$$\omega_a = \frac{2}{T}\tan\frac{\omega_d T}{2} \tag{3.6}$$

or, in a more symmetrical way

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

[11] 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:

$$\begin{aligned} s^{-1} &= \frac{z}{z-1} \\ s &= \frac{z-1}{z} \\ H_d(z) &= H_a\left(\frac{z-1}{z}\right) \\ H_d(e^{j\omega}) &= H_a\left(\frac{e^{j\omega}-1}{e^{j\omega}}\right) = H_a\left(1-e^{-j\omega}\right) \end{aligned} \tag{3.8}$$

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 (3.4) we have

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

from where

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

and

$$z = \frac{1 + \frac{sT}{2}}{1 - \frac{sT}{2}} \tag{3.9}$$

The equation (3.9) 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 (3.8) 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.[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.

3.8 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 $\omega_c T/2$, the frequency response itself and, specifically, its value at the cutoff will be different from the expected one. Fig. 3.13 illustrates. The -3dB level is specifically highlighted in Fig. 3.13, since this is the amplitude response value of the 1-pole lowpass filter at the cutoff, thereby aiding the visual identification of the cutoff point on the response curves.[13]

Apparently, the difference between analog and digital response is occuring due to the warping of the frequency axis (3.6). We would like to estimate the

[12] Note that thereby, should we become interested in the amplitude and phase responses of Fig. 3.12, we don't have to derive the discrete-time transfer function of Fig. 3.12. Instead we can simply take the amplitude and phase responses of the analog 1-pole (which are simpler to compute) and apply the mapping (3.7). This is the reason that we almost exclusively deal with analog transfer functions in this book, we simply don't need digital ones most of the time.

[13] Apparently, the picture in Fig. 3.13 will be the same at any other sampling rate, except that the frequency axis values will need to be relabelled proportionally to the sampling rate change. E.g. at 88.2kHz the labels would be 4, 8, 16, 22.05, 32 and 44.1kHz respectively. We could have labelled the axis in terms of normalized ω instead, but giving the absolute values is more illustrative. Particularly, the audible frequency range is easier to see.

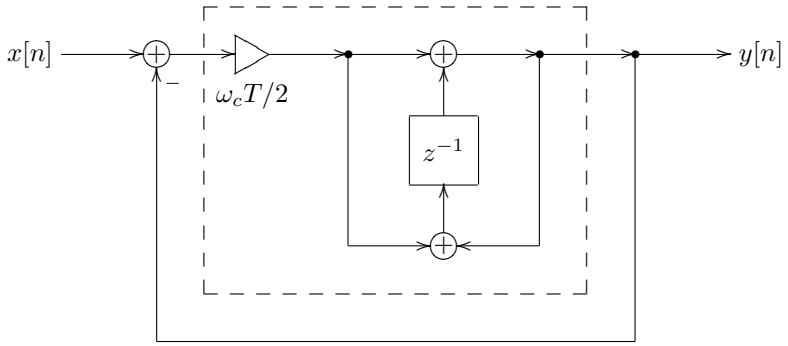


Figure 3.12: 1-pole TPT lowpass filter (the dashed line denotes the trapezoidal integrator).

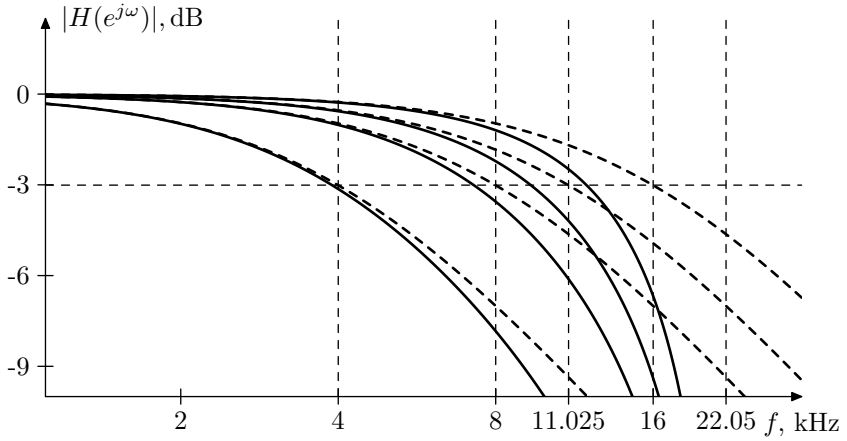


Figure 3.13: Amplitude response of an unprewarped bilinear-transformed 1-pole lowpass filter for a number of different cutoffs. Dashed curves represent the respective analog filter responses for the same cutoffs. Sampling rate 44.1kHz.

frequency error introduced by the warping. To simplify the further discussion let's rewrite (3.6) as a mapping function $\mu(\omega)$:

$$\omega_a = \mu(\omega_d) = \frac{2}{T} \tan \frac{\omega_d T}{2} \tag{3.10}$$

Now, given some desired analog response, we could take some point ω_a on this response and ask ourselves, where is the same point located on the digital response. According to (3.10), it is located at $\mu^{-1}(\omega_a)$ (where μ^{-1} is the function inverse of μ). Thus the ratio of the actual and desired frequencies is $\mu^{-1}(\omega_a)/\omega_a$, or, in the octave scale:

$$\Delta P = \log_2 \frac{\mu^{-1}(\omega_a)}{\omega_a}$$

The solid curve in Fig. 3.14 illustrates (note that Fig. 3.14 labels the ΔP axis

in semitones).

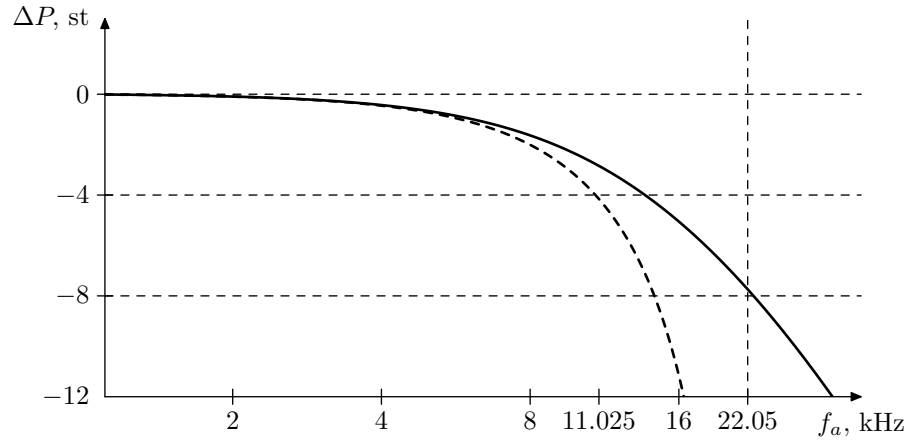


Figure 3.14: Bilinear transform's detuning of analog frequencies plotted against analog frequency (solid curve) or digital frequency (dashed curve). Sampling rate 44.1kHz.

We could also express the detuning of analog frequencies in terms of the corresponding digital frequency. Given the digital frequency response, we take some point ω_d and ask ourselves, where is the same point located on the analog response. According to (3.10), it is located at $\mu(\omega_d)$ and thus the frequency ratio is $\omega_d/\mu(\omega_d)$, respectively

$$\Delta P = \log_2 \frac{\omega_d}{\mu(\omega_d)}$$

The dashed curve in Fig. 3.14 illustrates. Note that we are not talking about how much the specified digital frequency will be detuned (because digital frequencies are not getting detuned, they are already where they are), it's still about how much the corresponding analog frequency will be detuned.

Thus, given an analog filter with a frequency response $H_a(j\omega)$, its digital counterpart will have its frequencies detuned as shown in Fig. 3.14. Particularly, the cutoff point, instead of being at the specified frequency $\omega = \omega_c$, will be at

$$\omega_d = \mu^{-1}(\omega_c) \tag{3.11}$$

In principle, one could argue that the frequency response change in Fig. 3.13 is not that drastic and could be tolerated, especially since the deviation occurs mostly in the high frequency range, which is not the most audible part of the frequency spectrum. This might have been the case with the 1-pole lowpass filter, however for other filters with more complicated amplitude responses it won't be as acceptable. Fig. 3.15 illustrates the frequency error for a 2-pole resonating lowpass filter. The resonance peaks (which occur close to the filter's cutoff) are very audible and so would be their detuning, which according to Fig. 3.14 is in the range of semitones. Particularly, at 16kHz the dashed curve in Fig. 3.14 shows a detuning of ca. 1 octave, meaning that we would have a resonance at this point when it should have been occurring at ca. 32kHz.

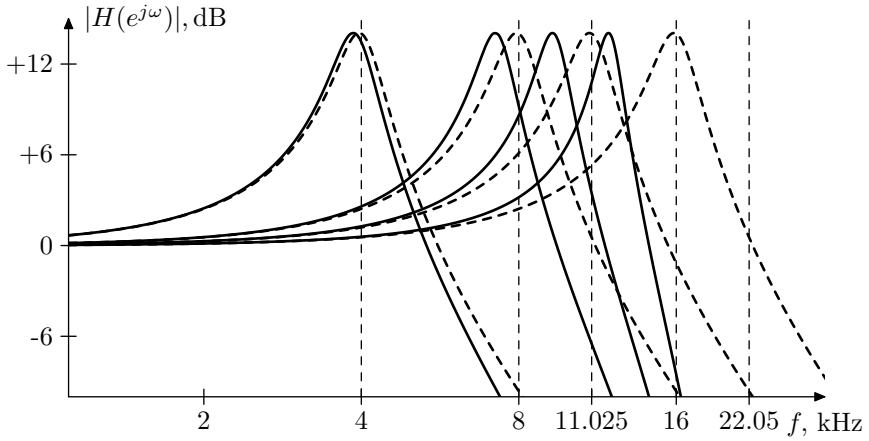


Figure 3.15: Amplitude response of an unprewarped bilinear-transformed resonating 2-pole lowpass filter for a number of different cutoffs. Dashed curves represent the respective analog filter responses for the same cutoffs. Sampling rate 44.1kHz.

Prewarping at cutoff

As a general rule (to which there are exceptions), we would like the cutoff point of the filter to be positioned exactly at the specified cutoff frequency ω_c. In this regard we could notice that if we used a different cutoff value

$$\tilde{\omega}_c = \mu(\omega_c) \tag{3.12}$$

then (3.11) would give

$$\omega_d = \mu^{-1}(\tilde{\omega}_c) = \mu^{-1}(\mu(\omega_c)) = \omega_c$$

and the cutoff point would be exactly where we wanted it to be. Fig. 3.16 illustrates. The cutoff correction (3.12) is a standard technique used in combination with the bilinear transform. It is referred to as *cutoff prewarping*.

Technically, cutoff prewarping means that we use $\tilde{\omega}_c$ instead of ω_c in the gains of the filter's integrators. However, the integrator gains are not exactly ω_c but rather $\omega_c T/2$. From (3.12) and (3.10) we have

$$\frac{\tilde{\omega}_c T}{2} = \frac{2}{T}\tan\frac{\omega_c T}{2}\cdot\frac{T}{2} = \tan\frac{\omega_c T}{2} \tag{3.13}$$

Thus, we can directly apply (3.13) to compute the prewarped gains $\tilde{\omega}_c T/2$. Note that (3.13) is essentially identical to (3.7).

The cutoff prewarping redistributes the frequency error shown in Fig. 3.14. In the absence of prewarping the error was zero at $\omega = 0$ and monotonically growing as ω increases. With the cutoff prewarping the error is zero at $\omega = \omega_c$ instead and grows further away from this point.

Indeed, let $H(j\omega)$ be unit-cutoff analog response of the filter in question. And let's pick up an analog frequency ω_a and find the respective detuning. The correct frequency response at ω_a is

$$H_a(\omega_a) = H(j\omega_a/\omega_c) \tag{3.14a}$$

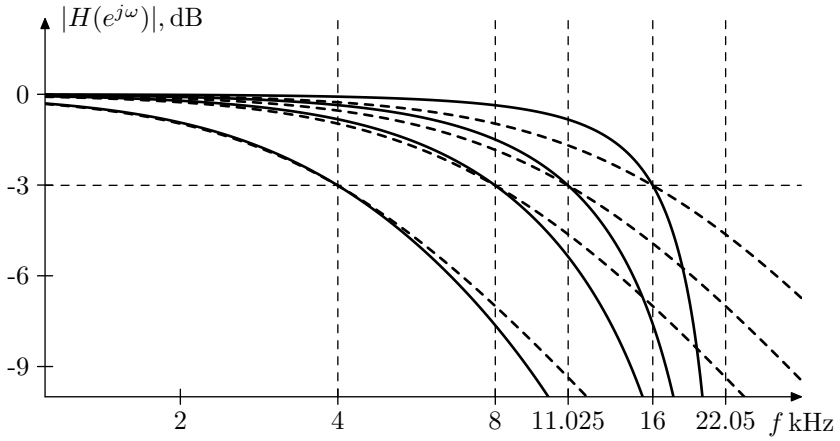


Figure 3.16: Amplitude response of a prewarped bilinear-transformed 1-pole lowpass filter for a number of different cutoffs. Dashed curves represent the respective analog filter responses for the same cutoffs. Sampling rate 44.1kHz.

On the other hand, given the prewarped cutoff $\tilde{\omega}_c = \mu(\omega_c)$, the digital frequency response at some frequency ω_d is

$$H_d(e^{j\omega_d}) = H(j\mu(\omega_d)/\tilde{\omega}_c) = H(j\mu(\omega_d)/\mu(\omega_c)) \tag{3.14b}$$

We want to find such ω_d that the arguments of $H(j\omega)$ in (3.14a) and (3.14b) are identical:

$$\frac{\mu(\omega_d)}{\mu(\omega_c)} = \frac{\omega_a}{\omega_c} \tag{3.15}$$

From where

$$\omega_d = \mu^{-1}\left(\omega_a \frac{\mu(\omega_c)}{\omega_c}\right)$$

The solid curves in Fig. 3.17 illustrate the respective detuning ω_a/ω_d (in semitones) of analog frequencies.

Alternatively from (3.15) we could express ω_a as a function of ω_d:

$$\omega_a = \frac{\omega_c}{\mu(\omega_c)}\mu(\omega_d)$$

thus expressing the analog frequency detuning ω_a/ω_d as a function of ω_d. The dashed curves in Fig. 3.17 illustrate.

Apparently, the maximum error to the left of $\omega = \omega_c$ is attained at $\omega = 0$. Letting $\omega_a \to 0$ and, equivalently, $\omega_d \to 0$ we have $\mu(\omega_d) \sim \omega_d$ and (3.15) turns into

$$\frac{\omega_d}{\mu(\omega_c)} \sim \frac{\omega_a}{\omega_c}$$

or

$$\frac{\omega_d}{\omega_a} \sim \frac{\mu(\omega_c)}{\omega_c}$$

and thus the detuning at $\omega = 0$ is

$$\Delta P\Big|_{\omega=0} = \log_2 \frac{\mu(\omega_c)}{\omega_c} \tag{3.16}$$

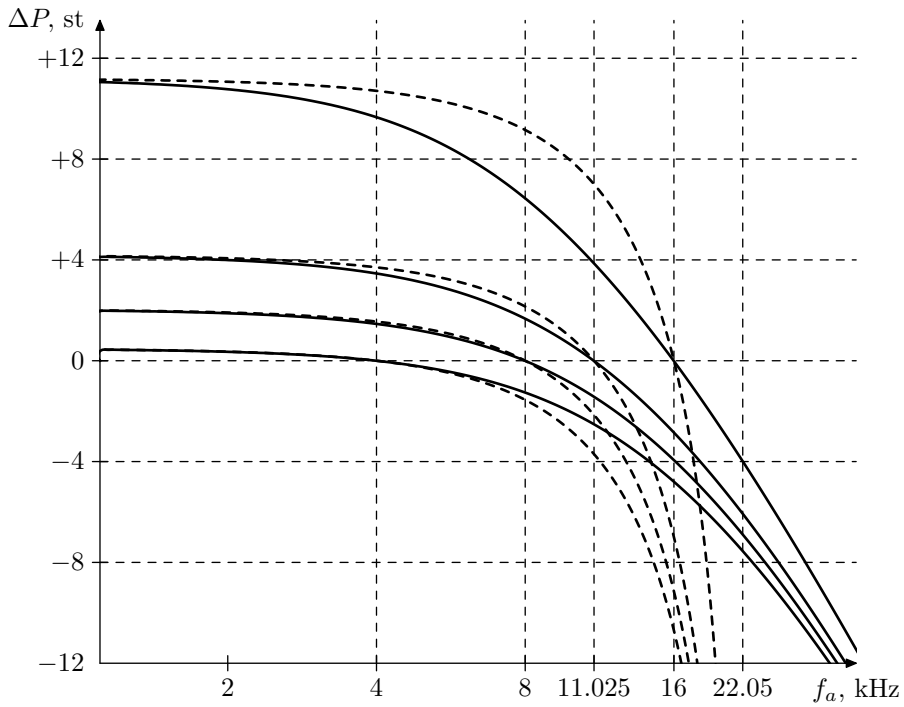


Figure 3.17: Prewarped bilinear transform's detuning of analog frequencies plotted against analog frequency (solid curves) or digital frequency (dashed curves). Different curves correspond to prewarping at different cutoff frequencies. Sampling rate 44.1kHz.

Other prewarping points

We have just developed the prewarping technique from the condition that the cutoff point must be preserved by the mapping (3.6). However instead we could have required any other point ω_p to be preserved.

Given the cutoff ω_c, the analog response at ω_p is

$$H_a(\omega_p) = H(j\omega_p/\omega_c) \tag{3.17a}$$

On the other hand, given the prewarped cutoff $\tilde{\omega}_c$ (we want to prewarp at a different point now, therefore we don't know yet, what is the relationship between ω_c and $\tilde{\omega}_c$) the digital frequency response at ω_p is

$$H_d(e^{j\omega_p}) = H(j\mu(\omega_p)/\tilde{\omega}_c) \tag{3.17b}$$

We want to find such $\tilde{\omega}_c$ that the arguments of $H(j\omega)$ in (3.17a) and (3.17b) are identical:

$$\frac{\omega_p}{\omega_c} = \frac{\mu(\omega_p)}{\tilde{\omega}_c}$$

and

$$\tilde{\omega}_c = \frac{\mu(\omega_p)}{\omega_p}\omega_c \tag{3.18}$$

Equation (3.18) is the generalized prewarping formula, where ω_p is the frequency response point of zero detuning. We will refer to ω_p as the *prewarping point*.

According to (3.18) prewarping at ω_p simply means that the cutoff should be multiplied by $\mu(\omega_p)/\omega_p$. At $\omega_p = \omega_c$ this multiplication reduces to (3.12).

In order to find the detuning at other frequencies, notice that equations (3.14) turn into

$$H_a(\omega_a) = H(j\omega_a/\omega_c)$$
$$H_d(e^{j\omega_d}) = H(j\mu(\omega_d)/\tilde{\omega}_c) = H\left(j\frac{\omega_p\mu(\omega_d)}{\omega_c\mu(\omega_p)}\right)$$

from where, equating the arguments of $H(j\omega)$:

$$\frac{\omega_p\mu(\omega_d)}{\omega_c\mu(\omega_p)} = \frac{\omega_a}{\omega_c}$$

we have

$$\frac{\mu(\omega_d)}{\mu(\omega_p)} = \frac{\omega_a}{\omega_p} \tag{3.19}$$

Equation (3.19) is identical to (3.15) except that it has ω_p in place of ω_c. Thus we could reuse the results of the previous analysis of the analog frequency detuning. In particular Fig. 3.17 fully applies, different curves corresponding to different prewarping points. At the same time, (3.16) simply turns to

$$\Delta P\Big|_{\omega=0} = \log_2 \frac{\mu(\omega_p)}{\omega_p} \tag{3.20}$$

Bounded cutoff prewarping

Even though cutoff prewarping is an absolutely standard technique and is often used without any second thought, the need for a different choice of the prewarping point is actually not as exotic as it might seem. Consider e.g. the amplitude response of a 1-pole highpass filter prewarped by (3.12), shown in Fig. 3.18. One can notice a huge discrepancy between analog and digital amplitude responses occurring well into the audible frequency range $[0, 16\text{kHz}]$. The error is getting particularly bad at cutoffs above 16kHz. In comparison, the responses of unprewarped filters in Fig. 3.19 even look kind of better, especially if only the audible frequency range is considered. This would be even more so, if higher sampling rates are involved, where the audible range error in Fig. 3.19 would become smaller, while the same error in Fig. 3.18 can still get as large, given a sufficiently high cutoff value.

Apparently, the large error within the audible range in Fig. 3.18 is due to the detuning error illustrated in Fig. 3.17. This error wasn't as obvious in the case of the 1-pole lowpass filter, since this filter's amplitude response is almost constant to the left of the cutoff point. On the other hand, highpass filter's amplitude response is changing noticeably in the same area, which makes the detuning error is made much more promiment.

Does this suggest that we shouldn't use cutoff prewarping with highpass filters? In principle this is engineer's decision. However consider the unprewarped resonating 2-pole highpass filter's amplitude response in Fig. 3.20. As with the

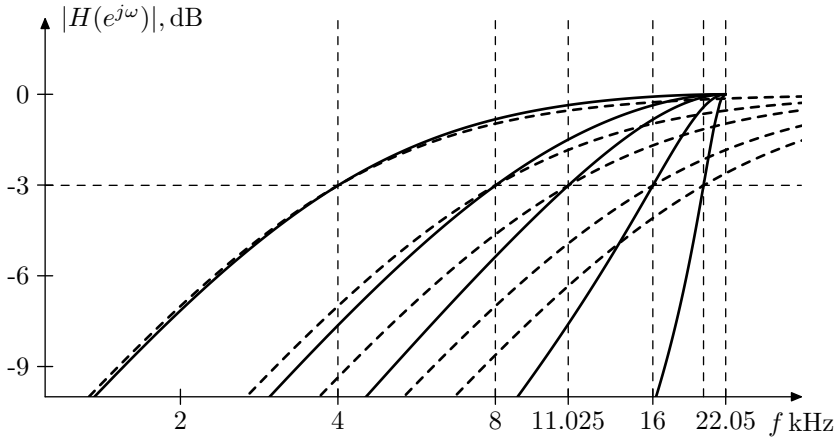


Figure 3.18: Amplitude response of a prewarped bilinear-transformed 1-pole highpass filter for a number of different cutoffs. Dashed curves represent the respective analog filter responses for the same cutoffs. Sampling rate 44.1kHz.

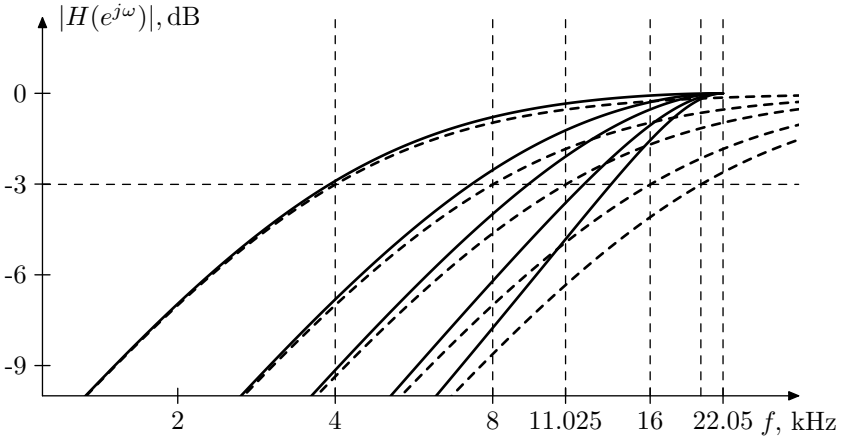


Figure 3.19: Amplitude response of an unprewarped bilinear-transformed 1-pole highpass filter for a number of different cutoffs. Dashed curves represent the respective analog filter responses for the same cutoffs. Sampling rate 44.1kHz.

responating lowpass, the resonance peak detuning is quite prominent here. Also the difference in the response value is magnified around the resonance point. All in all, we'd rather prewarp the cutoff (Fig. 3.21) and tolerate the detuning to the left of ω_c.

However notice, that as the cutoff peak is getting out of the audible range, we stop caring, where exactly it is positioned, since it can't be heard anyway. So, why should we then tolerage the error in the audible range which continues to increase even faster? Instead, at this moment we could fix the prewarping

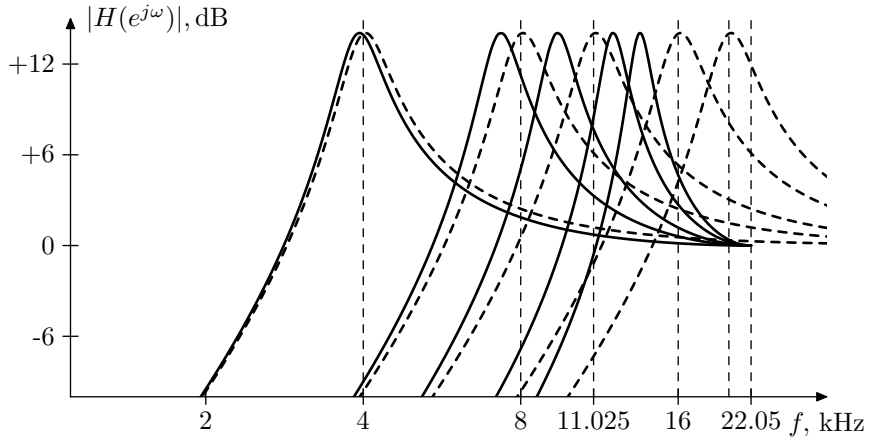


Figure 3.20: Amplitude response of an unprewarped bilinear-transformed resonating 2-pole highpass filter for a number of different cutoffs. Dashed curves represent the respective analog filter responses for the same cutoffs. Sampling rate 44.1kHz.

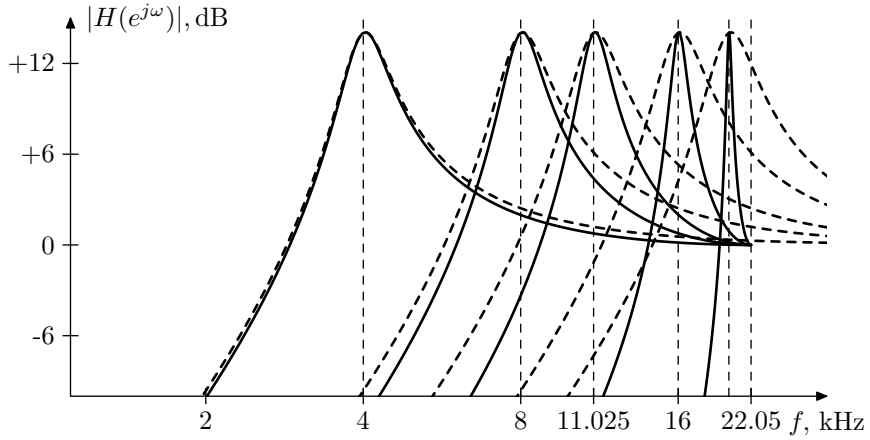


Figure 3.21: Amplitude response of a prewarped bilinear-transformed resonating 2-pole highpass filter for a number of different cutoffs. Dashed curves represent the respective analog filter responses for the same cutoffs. Sampling rate 44.1kHz.

point to the upper boundary of the audible range:

$$\omega_p = \begin{cases} \omega_c & \text{if } \omega_c \leq \omega_{\max} \\ \omega_{\max} & \text{if } \omega_c \geq \omega_{\max} \end{cases}$$

or simply

$$\omega_p = \min\{\omega_c, \omega_{\max}\} \tag{3.21}$$

where $\omega_{\max}$ is some point around 16kHz. At least then the detuning in the audible range won't grow any further than it is at $\omega_c = \omega_{\max}$ (Fig. 3.22). The picture gets even better at higher sampling rates (Fig. 3.23).

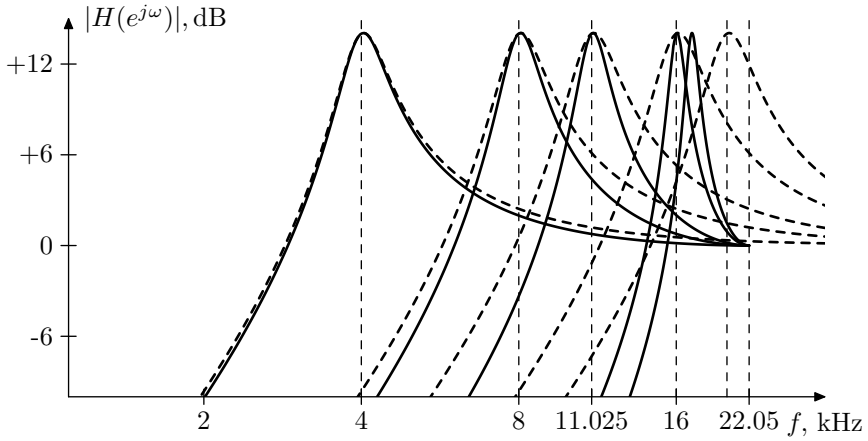


Figure 3.22: Effect of cutoff prewarping bounded at 16kHz. Sampling rate 44.1kHz.

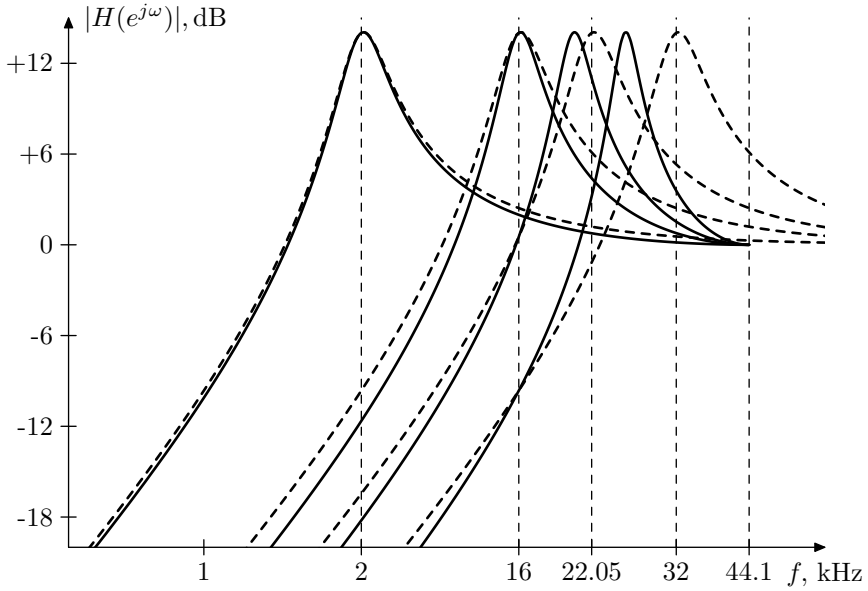


Figure 3.23: Effect of cutoff prewarping bounded at 16kHz. Sampling rate 88.1kHz.

Substituting (3.21) into (3.18) we obtain

$$\tilde{\omega}_c = \frac{\mu(\min\{\omega_c, \omega_{\max}\})}{\min\{\omega_c, \omega_{\max}\}}\,\omega_c = \begin{cases} \mu(\omega_c) & \text{if } \omega_c \le \omega_{\max} \\ \dfrac{\mu(\omega_{\max})}{\omega_{\max}}\,\omega_c & \text{if } \omega_c \ge \omega_{\max} \end{cases} \tag{3.22}$$

The mapping defined by (3.22) is shown in Fig. 3.24. Note that thereby we become able to specify the cutoffs beyond Nyquist, and actually to specify arbitrarily large cutoffs, since the new mapping curve is crossing the former vertical asymptote at $\omega_c = \pi/2$.

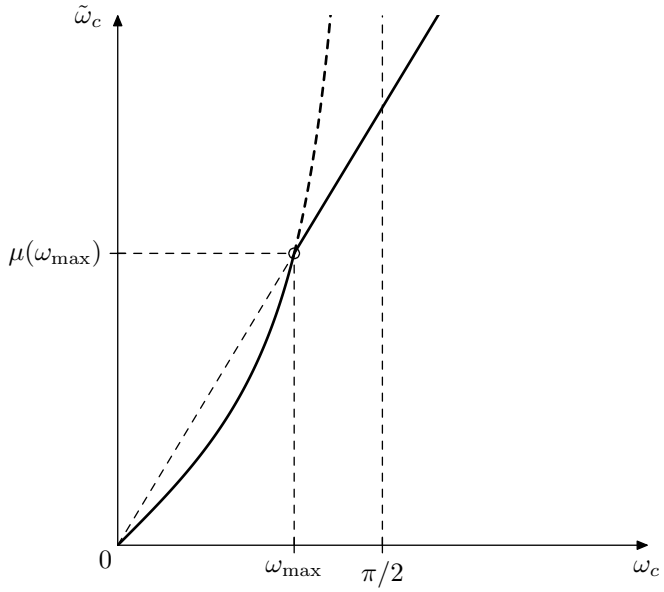


Figure 3.24: Bounded cutoff prewarping. The thick dashed line shows the unbounded prewarping continuation. The thin oblique dashed line is the continuation of the straight part of the prewarping curve. The black dot marks the breakpoint of the prewarping curve.

The breakpoint at $\omega_c = \omega_{\max}$ in Fig. 3.24 can be somewhat unexpected. In order to understand the mechanism behind its appearance suppose ω_c is varying with time. Using (3.18) we compute the time derivative of $\tilde{\omega}_c$:

$$\dot{\tilde{\omega}}_c = \omega_c \frac{\mathrm{d}}{\mathrm{d}t} \frac{\mu(\omega_p)}{\omega_p} + \frac{\mu(\omega_p)}{\omega_p} \dot{\omega}_c$$

As ω_c becomes larger than $\omega_{\max}$ the first term suddenly disappears and there is a jump in $\dot{\tilde{\omega}}_c$. In other words, the variation of the prewarping point makes its own contribution to $\dot{\tilde{\omega}}_c$. As soon as the variation stops, the respective contribution abruptly disappears.

The frequency detunings occurring in case of (3.22) can be found directly from Fig. 3.17, keeping in mind that (3.22) is simply another expression of (3.21).

Continuous-speed prewarping

The breakpoint occurring in Fig. 3.24 might be undesirable if the cutoff is being modulated, since there can be a sudden change of the perceived modulation speed as the cutoff traverses through the prewarping breakpoint. For that reason it might be desirable to smooth the breakpoint in one way or the other. The simplest approach would be to continue the curve as a tangent line after the

breakpoint:

$$\tilde{\omega}_c = \begin{cases} \mu(\omega_c) & \text{if } \omega_c \le \omega_{\max} \\ \mu(\omega_{\max}) + (\omega_c - \omega_{\max})\mu'(\omega_{\max}) & \text{if } \omega_c \ge \omega_{\max} \end{cases} \tag{3.23}$$

where

$$\mu'(\omega) = \frac{\mathrm{d}}{\mathrm{d}\omega}\left(\frac{2}{T}\tan\frac{\omega T}{2}\right) = \frac{1}{\cos^2\dfrac{\omega T}{2}} = 1 + \tan^2\frac{\omega T}{2} = 1 + \left(\frac{T}{2}\mu(\omega)\right)^2$$

is the derivative of $\mu(\omega)$. Note, however, that this will no longer keep $\omega_{\max}$ as the prewarping point and the situation would be something in between (3.12) and (3.22).

At $\omega_c \to \infty$ from (3.23) we have

$$\tilde{\omega}_c = \mu(\omega_{\max}) + (\omega_c - \omega_{\max})\mu'(\omega_{\max}) \sim \mu'(\omega_{\max})\omega_c \tag{3.24}$$

Comparing the right-hand side of (3.24) to (3.18) we obtain the equation for the effective prewarping point at infinity:

$$\frac{\mu(\omega_p)}{\omega_p} = \mu'(\omega_{\max}) \tag{3.25}$$

In principle ω_p can be found from (3.25), however we are not so much interested in how far off will be the prewarping point, as in the estimation of the associated increase in detuning. By (3.20)

$$\Delta P\Big|_{\omega=0} = \log_2\frac{\mu(\omega_p)}{\omega_p} = \log_2\mu'(\omega_{\max}) \qquad (\text{at } \omega_c = \infty)$$

which for $\omega_{\max} = 16\text{kHz}$ at 44.1kHz sampling rate gives ca. 2.5 octaves, while at 88.2kHz sampling rate it gives only about 0.5 octave. In comparison, at $\omega_c = \omega_{\max}$ (and respectively $\omega_p = \omega_{\max}$) we would have a smaller value

$$\Delta P\Big|_{\omega=0} = \log_2\frac{\mu(\omega_p)}{\omega_p} = \log_2\frac{\mu(\omega_{\max})}{\omega_{\max}}$$

which for $\omega_{\max} = 16\text{kHz}$ at 44.1kHz sampling rate gives ca. 1 octave, while at 88.2kHz sampling rate it gives only about 2 semitones.

We have therefore found the effect of (3.23) on the detuning occurring at $\omega = 0$ for $\omega_c \to \infty$. It would also be nice to estimate the same effect at $\omega = \omega_{\max}$. By (3.19)

$$\frac{\mu(\omega_d)}{\omega_a} = \frac{\mu(\omega_p)}{\omega_p}$$

which by (3.25) becomes

$$\frac{\mu(\omega_d)}{\omega_a} = \mu'(\omega_{\max})$$

Letting $\omega_d = \omega_{\max}$ we have

$$\omega_a = \frac{\mu(\omega_{\max})}{\mu'(\omega_{\max})} = \frac{\dfrac{2}{T}\tan\dfrac{\omega_{\max}T}{2}}{\cos^{-2}\left(\dfrac{\omega_{\max}T}{2}\right)} = \frac{1}{T}\sin\left(\omega_{\max}T\right)$$

and the detuning itself is the logarithm of the ratio

$$\frac{\omega_a}{\omega_d} = \frac{\omega_a}{\omega_{\max}} = \frac{\sin(\omega_{\max}T)}{\omega_{\max}T} = \operatorname{sinc}(\omega_{\max}T)$$

For $\omega_{\max} = 16$kHz at 44.1kHz sampling rate this gives ca. -1.5 octaves, while at 88.1kHz it gives ca. -4 semitones.

Since (as illustrated by Fig. 3.17) the detuning is a monotonic function of ω, at $\omega_c = \infty$ we are having the audible range detuning error in the range of ca. $[-1.5, 2.5]$ octaves at 44.1kHz sampling rate and in the range of ca. $[-4, 6]$ semitones at 88.2kHz sampling rate. Therefore it is more or less balanced out, although being a bit larger at higher frequencies.

Apparently, more elaborate ways of smoothing the breakpoint in Fig. 3.24 may be designed, but we won't cover them here as the options are almost infinite.

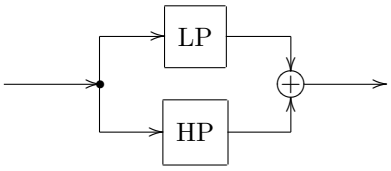
Prewarping of systems of filters

According to (3.18), prewarping is nothing more than a multiplication of the cutoff gains of the integrators by $\mu(\omega_p)/\omega_p$, where ω_p is the prewarping point.

Suppose we are having a system consisting of several filters connected together. When prewarping filters in such system, it would be a good idea to choose a common prewarping point for all filters. In this case we multiply their cutoffs by one and the same coefficient. Thereby their amplitude and phase responses are shifted by one and the same amount (in the logarithmic frequency axis), and they all retain the frequency response relationships which existed between them prior to prewarping. Effectively this is the same as changing the "common cutoff" of the filter system, and the frequency response of the entire system is simply shifted horizontally by the same amount, fully retaining its shape.

On the other hand by prewarping them independently we shift the frequency response of each filter differently from the others and the amplitude and phase relationships between those are thereby destroyed. Therefore the amplitude and phase response shapes of the entire system of filters are not preserved.

As an illustration consider a parallel connection of a 1-pole lowpass and a 1-pole highpass filter, the lowpass cutoff being $\omega_c/2$, the highpass cutoff being $2\omega_c$ where ω_c is the formal cutoff of the system:



The transfer function of such system (written in the unit-cutoff form) is

$$H(s) = \frac{1}{1+2s} + \frac{s/2}{1+s/2}$$

The result of prewarping the lowpass and the highpass separately at their respective cutoffs $\omega_c/2$ and 2ω is shown in Fig. 3.25. Actually for the $\omega_c = 11.025$kHz and $\omega_c = 16$kHz the highpass cutoff $2\omega_c$ needed to be clipped prior to prewaring,

since it is equal or exceeds Nyquist and cannot be directly prewarped by (3.12). Compare to Fig. 3.26 where the prewarping of both filters has been done at the common point ω_c.

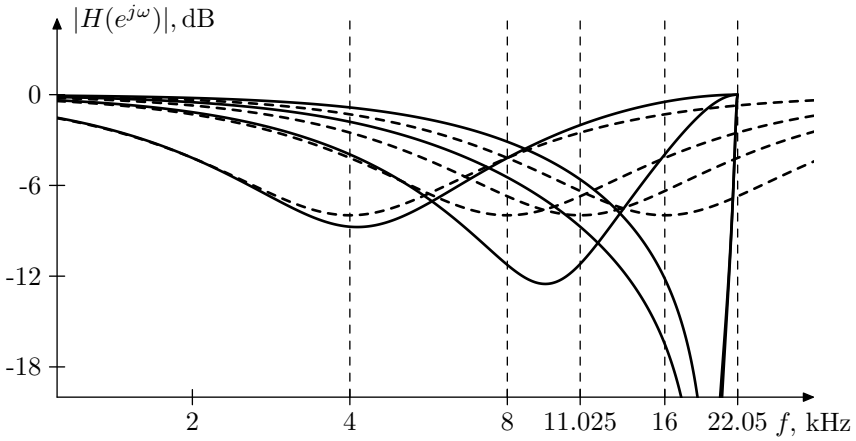


Figure 3.25: Separate prewarping of system components (for a number of different cutoffs). Dashed curves represent the respective analog filter responses for the same cutoffs. Sampling rate 44.1kHz.

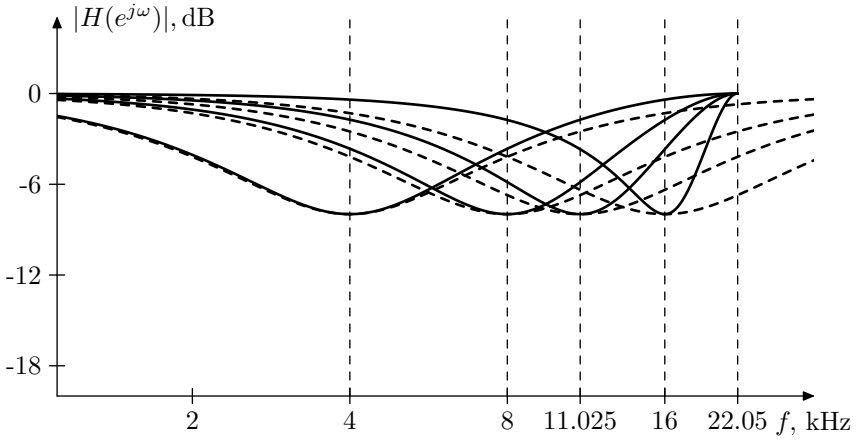


Figure 3.26: Common-point prewarping of system components (for a number of different cutoffs). Dashed curves represent the respective analog filter responses for the same cutoffs. Sampling rate 44.1kHz.

Other prewarping techniques

With 1-pole lowpass and highpass filters the only available control parameter is the filter cutoff. Thus the only option which we had for compensating the bilinear transform's frequency detuning was correcting the cutoff value. Other filters

may have more parameters available. Usually their parameters (e.g. resonance) will have a strong "vertical" effect on the amplitude response and thus are not very suitable for compensating the frequency detuning. However in some cases there will be further options of horizonally altering the filter's amplitude (and phase) responses without causing noticeable changes in the vertical direction. In such cases we will have further options for more detailed compensation of the frequency detuning.

Note, however, that these compensations, being not expressible as cutoff multiplication, may destroy the frequency response of a system of filters, unless there is some other way to make them have identical effect on all of the filters in the system. We will discuss some examples of this later in the book.

3.9 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. 3.27, 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?

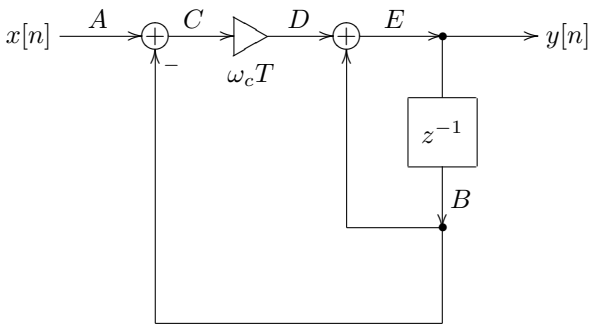


Figure 3.27: Naive 1-pole lowpass filter and the respective signal computation order.

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. 3.28). Now there are no delayless feedback paths and we can arrange the computation order in a way similar to Fig. 3.27. 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.

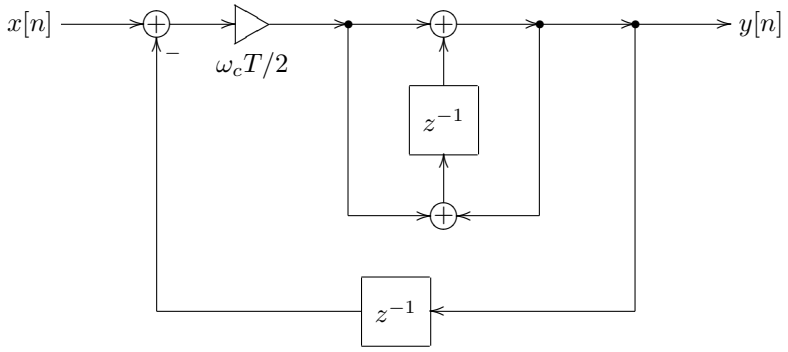


Figure 3.28: Digital 1-pole lowpass filter with a trapezoidal integrator and an extra delay in the feedback.

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. 3.28) 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. Since there are no delayless loops in the integrator, it's not difficult to obtain the following expression for $y[n]$:

$$y[n] = \frac{\omega_c T}{2} x[n] + u[n-1] \tag{3.26}$$

Notice that, at the time $x[n]$ arrives at the integrator's input, all values in the right-hand side of (3.26) are known (no unknown variables). Introducing notation

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

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. 3.29. 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.

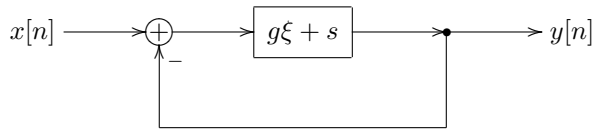


Figure 3.29: 1-pole TPT lowpass filter with the integrator in the instantaneous response form.

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,

$$y = g(x - y) + s \tag{3.27}$$

The equation (3.27) is the zero-delay feedback equation for the filter in Fig. 3.29 (or, for that matter, in Fig. 3.12). Solving this equation, we obtain

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

and respectively

$$y = \frac{gx + s}{1 + g} \tag{3.28}$$

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.[14]

[14]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.

It's worth mentioning that (3.28) can be used to obtain the instantaneous response of the entire filter from Fig. 3.12. Indeed, rewriting (3.28) as

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

and introducing notations

$$G = \frac{g}{1+g}$$
$$S = \frac{s}{1+g}$$

we have

$$y = Gx + S \tag{3.29}$$

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 (3.29) e.g. to solve the zero-delay feedback problem for some larger feedback loop containing a 1-pole lowpass filter.

3.10 Implementations

1-pole lowpass

We are now going to convert the structure in Fig. 3.12 into a piece of code. Let's introduce helper variables into Fig. 3.12 as shown in Fig. 3.30, where we have used the already known to us fact that, given the integrator's instantaneous response $gx + s$, the value of s equals the output of the z^{-1} element.

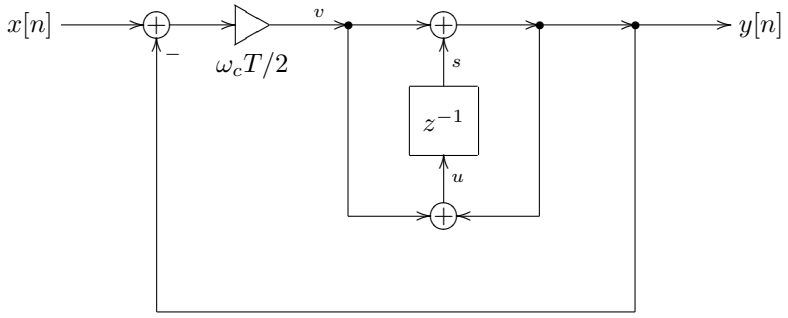


Figure 3.30: 1-pole TPT lowpass filter with helper variables.

We already know y from (3.29). Since $g = \omega_c T/2$, we have

$$v = g(x-y) = g\left(x - Gx - S\right) = g\left(x - \frac{g}{1+g}x - \frac{s}{1+g}\right) =$$
$$= g\left(\frac{1}{1+g}x - \frac{s}{1+g}\right) = g\frac{x-s}{1+g} \tag{3.30}$$

Now (3.30) gives a direct expression for v in terms of known signals, not using y. In order to avoid unnecessary computations, we can simply reobtain y using

the obvious from Fig. 3.30 fact that y is just a sum of v and s:

$$y = v + s \tag{3.31}$$

We also need the z^{-1} input (which we need to store in the z^{-1}'s memory) which is also obtained from Fig. 3.30 in an obvious way:

$$u = y + v \tag{3.32}$$

The equations (3.30), (3.31) and (3.32) can be directly expressed in program code:

```
// perform one sample tick of the lowpass filter
// G = g/(1+g)
// the variable 's' contains the state of z^-1 block
v := (x-s)*G;
y := v + s;
s := y + v;
```

or instead expressed in a block diagram form (Fig. 3.31). Notice that the block diagram doesn't contain any delayless loops anymore.

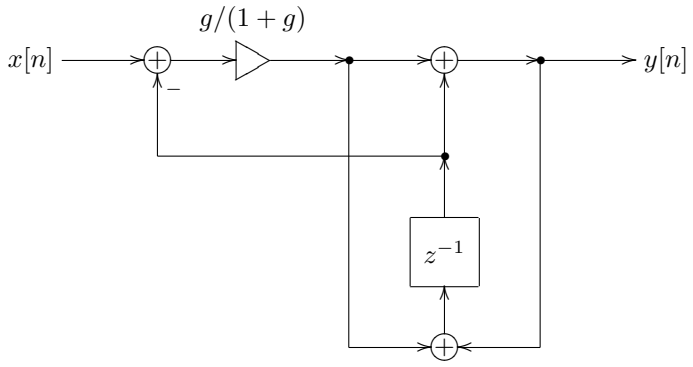


Figure 3.31: 1-pole TPT lowpass filter with resolved zero-delay feedback.

1-pole multimode

The highpass signal can be obtained from the structure in Fig. 3.31 in a trivial manner, since $y_{\mathrm{HP}} = x - y_{\mathrm{LP}}$, thereby turning Fig. 3.31 into a multimode 1-pole.

1-pole highpass

If we need *only* the highpass signal, we could do it in a smaller number of operations than in the multimode 1-pole. By noticing that

$$y_{\mathrm{HP}} = x - y_{\mathrm{LP}} = x - (v + s) = (x - s) - v = (x - s) - \frac{g}{1+g}(x - s) = \frac{1}{1+g}(x - s)$$

and that

$$u = y_{\mathrm{LP}} + v = s + 2v = s + \frac{2g}{1+g}$$

we can implement the highpass filter as follows:

```
// perform one sample tick of the highpass filter
// Ghp = 1/(1+g)
// the variable 's' contains the state of z^-1 block
xs := x - s;
y := xs*Ghp;
s := s + y*2g;
```

however this way we have traded an addition/subtraction pair for one multiplication, plus instead of one cutoff-dependent parameter G we need to store and access Ghp and 2g. Therefore this is not necessarily a performance improvement.

1-pole allpass

The allpass signal can be obtained from the multimode 1-pole in a trivial manner, recalling that $y_{\mathrm{AP}} = y_{\mathrm{LP}} - y_{\mathrm{HP}}$. However, if we need *only* the allpass signal, we could save a couple of operations. Noticing that

$$y_{\mathrm{AP}} = y_{\mathrm{LP}} - y_{\mathrm{HP}} = v + s - (x - (v + s)) = (s + 2v) - (x - s)$$

and that $s + 2v = u$ is the new state of the z^{-1} block, we obtain

```
// perform one sample tick of the allpass filter
// 2Glp=2g/(1+g)
// the variable 's' contains the state of z^-1 block
xs := x - s;
s := s + xs*2Glp;
y := s - xs;
```

3.11 Direct forms

Consider again the equation (3.5), 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.

Having obtained a digital transfer function from (3.5), 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 $1+\omega_c T/2$:

$$H_d(z) = \frac{\frac{\frac{\omega_c T}{2}}{1+\frac{\omega_c T}{2}}(1+z^{-1})}{1-\frac{1-\frac{\omega_c T}{2}}{1+\frac{\omega_c T}{2}}z^{-1}} \tag{3.33}$$

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:

$$H(z) = \frac{\sum_{n=0}^{N} b_n z^{-n}}{1-\sum_{n=1}^{N} a_n z^{-n}} \tag{3.34}$$

This transfer function can be implemented by the structure in Fig. 3.32 or by the structure in Fig. 3.33. One can verify (by computing the transfer functions of the respective structures) that they indeed implement the transfer function (3.34). There are also transposed versions of these structures, which the readers should be able to construct on their own.

Let's use the direct form II to implement (3.33). Apparently, we have

$$N = 1$$
$$b_0 = b_1 = \frac{\frac{\omega_c T}{2}}{1+\frac{\omega_c T}{2}}$$
$$a_1 = \frac{1-\frac{\omega_c T}{2}}{1+\frac{\omega_c T}{2}}$$

and the direct form implementation itself is the one in Fig. 3.34 (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. 3.34 should be identical to the TPT filter in Fig. 3.12, since both

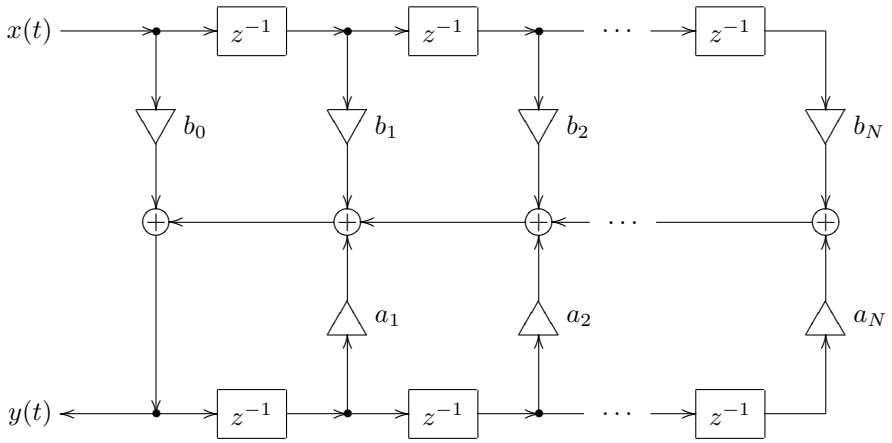


Figure 3.32: Direct form I (DF1).

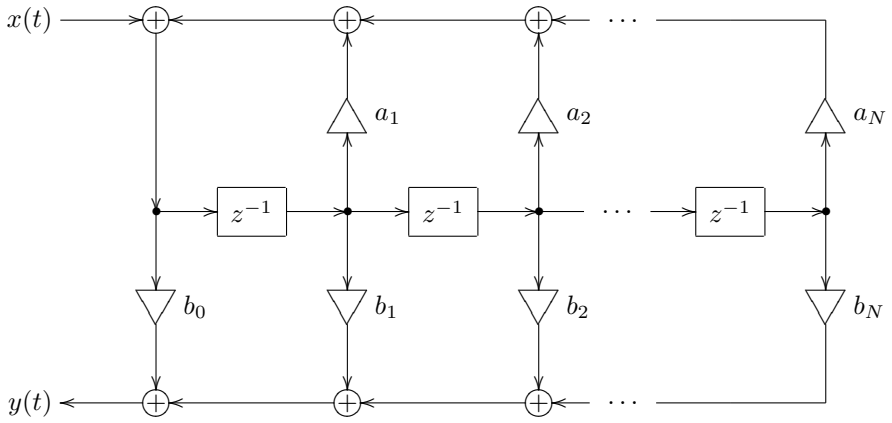


Figure 3.33: Direct form II (DF2), a.k.a. canonical form.

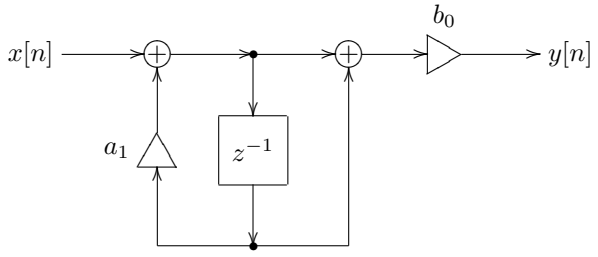


Figure 3.34: Direct form II 1-pole lowpass filter.

implement the same bilinear-transformed analog transfer function (2.5). When

the cutoff is modulated, however, the performance will be different.

We have already discussed in Sections 2.7 and 2.16 that different topologies may have different time-varying behavior even if they share the same transfer function. Apparently, the difference in behavior between Fig. 3.34 and Fig. 3.12 is another example of that. Comparing the implementations in Figs. 3.34 and 3.12, we notice that the structure in Fig. 3.34 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. 3.34 is inferior to the one in Fig. 3.12 (or Fig. 3.31). 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. 3.34 is concerned solely with implementing a correct transfer function. Since Fig. 3.34 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, TPBLT). 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.[15]

In principle, sometimes there are possibilities to "manually fix" the structures such as in Fig. 3.34. 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.[16] 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,[17] but even 2nd-order direct forms could already noticeably suffer from precision losses.

[15] 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.

[16] 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.

[17] 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.

3.12 Transient response

Looking at the 1-pole lowpass filter's discrete-time transfer function (3.33) and noticing that $\omega_c = -p$ where p is the analog pole, we could rewrite (3.33) as

$$H(z) = \frac{\frac{\frac{-pT}{2}}{1-\frac{pT}{2}}(1+z^{-1})}{1-\frac{1+\frac{pT}{2}}{1-\frac{pT}{2}}z^{-1}}$$

Comparing this to (3.9) we notice that

$$\frac{1+\frac{pT}{2}}{1-\frac{pT}{2}} = \tilde{p}$$

where $\tilde{p}$ is the result of the application of the inverse bilinear transform formula (3.9) to p. Further noticing that

$$\frac{\frac{-pT}{2}}{1-\frac{pT}{2}} = \frac{1}{2}\cdot\left(\frac{1-\frac{pT}{2}}{1-\frac{pT}{2}} - \frac{1+\frac{pT}{2}}{1-\frac{pT}{2}}\right) = \frac{1-\tilde{p}}{2} \tag{3.35}$$

we rewrite $H(z)$ as

$$H(z) = \frac{1-\tilde{p}}{2}\cdot\frac{1+z^{-1}}{1-\tilde{p}z^{-1}} = \frac{1-\tilde{p}}{2}\cdot\frac{z+1}{z-\tilde{p}}$$

Thus $\tilde{p}$ is the pole of $H(z)$, as we should have expected.

On the other hand, applying trapezoidal integration to the 1-pole lowpass differential equation in the pole form (2.14), we have

$$y[n] - y[n-1] = pT\cdot\left(\frac{y[n]+y[n-1]}{2} - \frac{x[n]+x[n-1]}{2}\right)$$

from where

$$\left(1-\frac{pT}{2}\right)y[n] = \left(1+\frac{pT}{2}\right)y[n-1] - \frac{pT}{2}\cdot(x[n]+x[n-1])$$

$$\begin{aligned} y[n] &= \frac{1+\frac{pT}{2}}{1-\frac{pT}{2}}y[n-1] + \frac{\frac{-pT}{2}}{1-\frac{pT}{2}}\cdot(x[n]+x[n-1]) = \\ &= \tilde{p}y[n-1] + (1-\tilde{p})\frac{x[n]+x[n-1]}{2} \end{aligned}$$

where we have used (3.35).

Now consider a complex exponential $x[n] = X(z)z^n$. For such $x[n]$ we have

$$y[n] = \tilde{p}y[n-1] + (1-\tilde{p})\frac{1+z^{-1}}{2}X(z)z^n = \tilde{p}y[n-1] + \tilde{q}z^n \tag{3.36}$$

where we introduced notation

$$\tilde{q} = (1-\tilde{p})\frac{1+z^{-1}}{2}X(z)$$

for conciseness. Recursively substituting (3.36) into itself at progessively decreasing values of n we obtain

$$\begin{aligned}
y[n] &= \tilde{p}y[n-1] + \tilde{q}z^n = \\
&= \tilde{p}\left(\tilde{p}y[n-2] + \tilde{q}z^{n-1}\right) + \tilde{q}z^n = \\
&= \tilde{p}^2 y[n-2] + \left(\tilde{p}z^{-1} + 1\right)\tilde{q}z^n = \\
&= \tilde{p}^2\left(\tilde{p}y[n-3] + \tilde{q}z^{n-2}\right) + \left(\tilde{p}z^{-1} + 1\right)\tilde{q}z^n = \\
&= \tilde{p}^3 y[n-3] + \left(\left(\tilde{p}z^{-1}\right)^2 + \tilde{p}z^{-1} + 1\right)\tilde{q}z^n = \\
&\dots \\
&= \tilde{p}^n y[0] + \left(\left(\tilde{p}z^{-1}\right)^{n-1} + \left(\tilde{p}z^{-1}\right)^{n-2} + \dots + \tilde{p}z^{-1} + 1\right)\tilde{q}z^n = \\
&= \tilde{p}^n y[0] + \frac{\left(\tilde{p}z^{-1}\right)^n - 1}{\tilde{p}z^{-1} - 1}\tilde{q}z^n = \tilde{p}^n y[0] + \frac{\tilde{p}^n - z^n}{\tilde{p} - z}\tilde{q}z = \\
&= \tilde{p}^n y[0] + \frac{z^n - \tilde{p}^n}{z - \tilde{p}}(1 - \tilde{p})\frac{z+1}{2}X(z) = \\
&= \tilde{p}^n y[0] + (z^n - \tilde{p}^n)H(z)X(z) = \\
&= H(z)X(z)z^n + (y[0] - H(z)X(z)) \cdot \tilde{p}^n = \\
&= y_s[n] + (y[0] - y_s[0]) \cdot \tilde{p}^n = y_s[n] + y_t[n]
\end{aligned}$$

where

$$\begin{aligned}
y_s[n] &= H(z)X(z)z^n \\
y_t[n] &= (y[0] - y_s[0]) \cdot p^n
\end{aligned}$$

are the steady-state and transient responses respectively.

Thus, the discrete-time 1-pole transient response is a decaying exponent $\tilde{p}^n$, provided the discrete-time system is stable and $|\tilde{p}| < 1$. If $|\tilde{p}| > 1$ the transient response grows infinitely.

3.13 Instantaneously unstable feedback

Writing the solution (3.28) for the zero-delay feedback equation (3.27) we in fact have slightly jumped the gun. Why? Let's consider once again the structure in Fig. 3.29 and suppose g gets negative and starts growing in magnitude further in the negative direction.[18] When g becomes equal to -1, the denominator of (3.28) turns into zero. Something bad must be happening at this moment.

Instantaneous smoother

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 instantaneously. It can change very quickly, but not instantaneously, it's always a continuous function of time. We could imagine there is a smoother

[18] 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.

unit somewhere in the feedback path (Fig. 3.35). This smoother unit has a very very fast response time. We introduce the notation $\bar{y}$ for the output of the smoother.

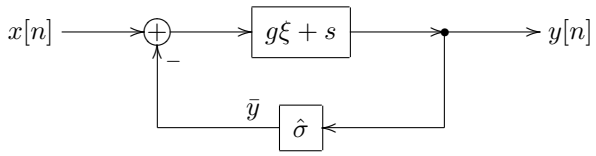


Figure 3.35: Digital 1-pole lowpass filter with a trapezoidal integrator in the instantaneous response form and a smoother unit $\hat{\sigma}$ in the delayless feedback path.

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.

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

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

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.[19] 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:

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

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

Now consider that

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

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.

[19] 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).

So, in case $1 + g > 0$ the instantaneous feedback smoothing process always converges. Now assume $1 + g \leq 0$. 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 idea of the smoother introduced in Fig. 3.35 can be used as a general means for analysing zero-delay feedback structures for instantaneous instability. We will refer to this technique as *instantaneous smoother*.

1-pole lowpass as an instantaneous smoother

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 $\frac{1}{s+1}$)[20] and notice that in that case the structure in Fig. 3.35 can be redrawn as in Fig. 3.36. This filter has two formal inputs $x[n]$ and s and one output $y[n]$.

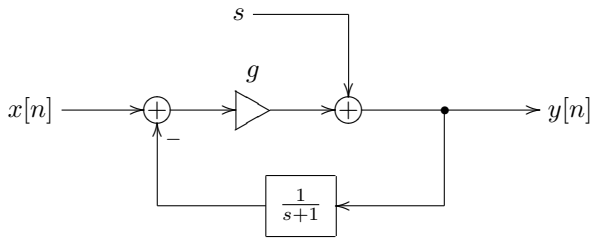


Figure 3.36: An instantaneous representation of a digital 1-pole lowpass filter with a trapezoidal integrator and an analog lowpass smoother.

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 $y[n](t)$:

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

from where

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

that is

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

[20] 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.

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.[21]

Generalized zero-delay feedback loop

The zero-delay feedback instantaneous response structure in Fig. 3.29 can be considered as a particular case of a general one, drawn in Fig. 3.37, where the input signal $x[n]$ has been incorporated into the s term of the instantaneous response $g\xi + s$ and the negative feedback has been incorporated into the factor g. Indeed, Fig. 3.37 can be obtained from Fig. 3.29 via

$$\begin{aligned} G &= -g \\ S &= s + gx \end{aligned}$$

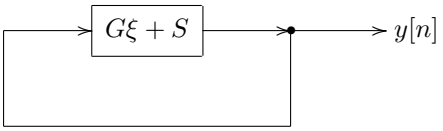


Figure 3.37: General zero-delay feedback structure in the instantaneous response form.

The zero-delay feedback equation solution written for Fig. 3.37 is obviously

$$y = \frac{S}{1 - G} \tag{3.37}$$

From the previous discussion it should be clear that the structure becomes instantaneously unstable for $G \geq 1$, that is when the total instantaneous gain of the feedback loop is 1 or more.

The solution form (3.37) therefore provides a generic means to check an arbitrary zero-delay feedback loop for instantaneous instability. E.g. rewriting (3.28) (which we had written for Fig. 3.29) in the form (3.37) we obtain

$$y = \frac{gx + s}{1 - (-g)}$$

where $-g$ is the total instantaneous gain of the feedback loop (including the feedback inversion), and thus the structure is instantaneously unstable at $-g \geq 1$ (or, equivalently, $g \leq -1$).

[21] 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 singular cases, arising out of pole/zero cancellation are possible, though).

It might be tempting to simply say that the instantaneously unstable zero-delay feedback occurs whenever the denominator of the zero-delay feedback equation's solution becomes zero or negative. However, this actually depends on how did we arrive at the solution expression. E.g. if we multiply both the numerator and the denominator of (3.28) by -1, the instantaneously unstable case will occur for zero or positive denominator values. Therefore, we need to make sure that our solution is written in the form (3.37) (where we need to verify that G is the total instantaneous gain of the feedback loop) and only then can we say that zero or negative denominator values correspond to instantaneously unstable feedback.

Limits of bilinear transform

We have seen that for 1-poles the continuous- and discrete-time transient responses are

$$y_t(t) = (y(0) - y_s(0)) \cdot e^{pt}$$
$$y_t[n] = (y[0] - y_s[0]) \cdot \tilde{p}^n$$

where the the discrete-time pole $\tilde{p}$ is obtained from continuous-time pole p via inverse bilinear transform (3.9).

In order to compare the transient responses we could compare the growth of y over one sampling period T:

$$y_t(t) = y_t(t - T) \cdot e^{pT}$$
$$y_t[n] = y_t[n - 1] \cdot \tilde{p}^n$$

The comparison of e^{pT} vs. $\tilde{p}$ is done in Fig. 3.38. One can notice that as $p \to 2/T - 0$ the value of $\tilde{p}$ grows too quickly (compared to e^{pT}), approaching infinity. This means that discrete-time transient response is growing infinitely fast at $p = 2/T$, or, respectively as $pT/2 = 1$. At $p > 2/T$ the value of $\tilde{p}$ is getting completely different from e^{pT}, particulary the sign of $y[n]$ begins to alternate between successive samples.

Now recall that in the 1-pole zero-delay feedback equation (3.28) we had $g = \omega_c T/2 = -pT/2$. Thus, as $g = -pT/2 \to -1 + 0$ the discrete-time transient response is becoming infinitely fast. Close to this point and further beyond it, trapezoidal integration doesn't deliver a reasonable approximation to the continuous-time case anymore.

If we attempt to interpret the same in terms of bilinear transform, then we already know (Fig. 3.38) that the inverse bilinear transform (3.9) is becoming infinitely large at $s = 2/T$, that is the inverse bilinear transform formula has a pole at $s = 2/T$. This means that, if we are having a continuous-time system with a pole at $s \approx -2/T$ (which in case of the 1-pole lowpass corresponds to $g = \omega_c T/2 \approx -1$), then after the bilinear transform the system will have a pole at $z \approx \infty$, and the transformation result doesn't work really well.

Avoiding instantaneously unstable feedback

Alright, so we have found out that zero-delay feedback structures are instantaneously unstable when the total instantaneous gain of the feedback loop is greater than or equal to 1, but what can we do about it? Firstly, the problem

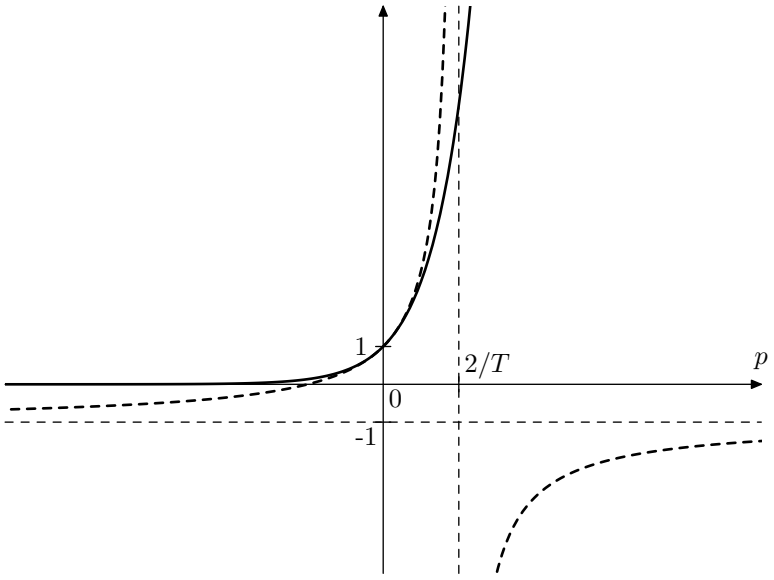


Figure 3.38: e^{-pT} (solid) vs. $\tilde{p} = (1 + pT/2)/(1 - pT/2)$ (thick dashed) as functions of p. The two thin dashed lines are asymptotes of $(1 + pT/2)/(1 - pT/2)$.

typically doesn't occur. Mostly, in (3.37) we have $G < 0$, e.g. in the 1-pole lowpass case we have $G = -g < 0$ for positive cutoff values. Even if G is or can become positive, the situation $G \geq 1$ occurs at really excessive parameter settings. Therefore 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.

Secondly, let's notice that $g = \omega_c T/2$. Therefore another solution could be to increase the sampling rate, which reduces the sampling period T and respectively the value of g (from an alternative point of view, it shifts the inverse bilinear transform's pole $2/T$ further away from the origin).

Unstable bilinear transform

There is yet another idea, which is not widely used, but we are going to discuss it anyway.[22] So, the instantaneous instability is occurring at the moment when one of the analog filter's poles hits the pole of the inverse bilinear transform (3.9), 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 this is that the inverse Laplace

[22]This idea has occurred to the author during the writing of the first revision of this book. The author didn't try it in practice yet, neither is he aware of other attempts.

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 non-linear filters, and there are not many 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.

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 (3.4), 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:

$$s = \sigma + \frac{2}{T} \cdot \frac{ze^{-\sigma T} - 1}{ze^{-\sigma T} + 1} \tag{3.38}$$

We can refer to (3.38) 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.[23] Notice that at $\sigma = 0$ the unstable bilinear transform turns into an ordinary bilinear transform. The inverse transform is obtained by

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

from where

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

and

$$z = e^{\sigma T}\frac{1 + \frac{(s-\sigma)T}{2}}{1 - \frac{(s-\sigma)T}{2}} \tag{3.39}$$

Apparently the inverse unstable bilinear transform (3.39) 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

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

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 (3.38) we first need to obtain the expression for $1/s$ from (3.38):

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

[23] Apparently, the unstable bilinear transform defines the same relationship between $\operatorname{Im} s$ and $\arg z$ as the ordinary bilinear transform. Therefore prewarping can be done in the same way as for the ordinary bilinear transform.

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

That is

$$\frac{1}{s} = \frac{T}{2+\sigma T} \cdot \frac{1+e^{\sigma T}z^{-1}}{1-\frac{2-\sigma T}{2+\sigma T}e^{\sigma T}z^{-1}} \tag{3.40}$$

A discrete-time structure implementing (3.40) could be e.g. the one in Fig. 3.39. Yet another approach could be to convert the right-hand side of (3.40) 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. 3.39.

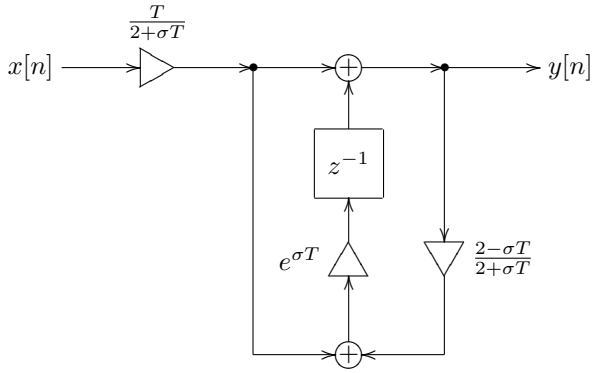


Figure 3.39: Transposed direct form II-style "unstable" trapezoidal integrator.

3.14 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.[24] In this case we could say

[24]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 $\mu_2(\omega)$.

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 (3.6)). One cool thing about 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 (3.6):

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

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.[25] The transfer function of a differentiator is apparently simply $H(s) = s$, so we could use (3.4) to build a discrete-time

[25] The real-world analog electronic circuits are "built around" integrators rather than differentiators. However, formally one still can "invert" the causality direction in the equations and pretend that $\dot{x}(t)$ is defined by $x(t)$, and not vice versa.

"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 better 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. 2.9 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.[26] 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 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 (3.9). From (3.9) we obtain

$$z^{-1} = \frac{1 - \frac{sT}{2}}{1 + \frac{sT}{2}} \tag{3.41}$$

The right-hand side of (3.41) 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 (3.41). 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

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

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

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

$$z^{-1} = \frac{1 - sT/2\alpha}{1 + sT/2\alpha} \tag{3.42}$$

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 (3.42), 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.[27] 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 (3.42), 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.

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.

[27] 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.

Chapter 4

State variable filter

After having discussed 1-pole filters, we are going to instroduce a 2-pole filter. With 2-pole filters there is more freedom in choosing the filter topology than with 1-poles, where any implementation of the latter would essentially be based on a feedback loop around an integrator. A 2-pole topology of fundamental importance and high usability is a classical analog model, commonly referred to as *state-variable filter* (SVF). It can also serve as a basis for building arbitrary 2-pole filters by means of modal mixture.

4.1 Analog model

The block diagram of the state-variable filter is shown in Fig. 4.1. The three outputs are the highpass, bandpass and lowpass signals. As usual, one can apply transposition to obtain a filter with highpass, bandpass and lowpass inputs (Fig. 4.2).

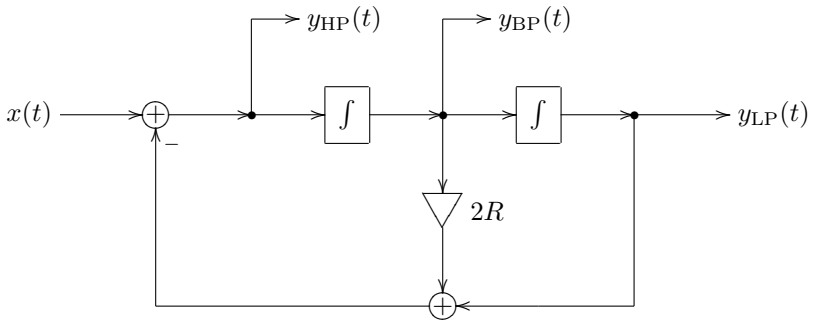


Figure 4.1: 2-pole multimode state-variable filter.

The differential equations implied by Fig. 4.1 are

$$\begin{aligned} y_{\mathrm{HP}} &= x - 2Ry_{\mathrm{BP}} - y_{\mathrm{LP}} \\ \dot{y}_{\mathrm{BP}} &= \omega_c \cdot y_{\mathrm{HP}} \\ \dot{y}_{\mathrm{LP}} &= \omega_c \cdot y_{\mathrm{BP}} \end{aligned} \tag{4.1}$$

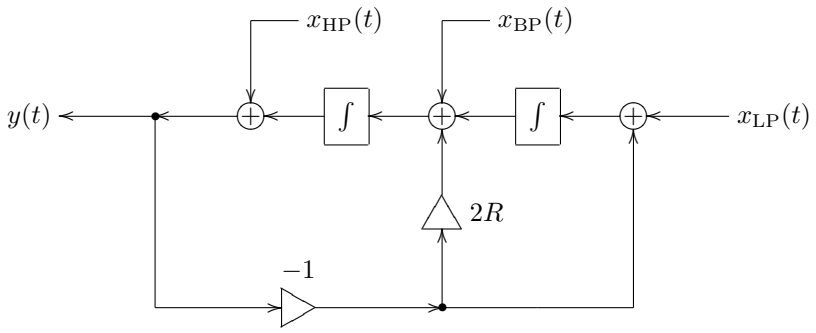


Figure 4.2: Transposed 2-pole multimode state-variable filter.

Rewriting them in terms of the lowpass signal $y = y_{\mathrm{LP}}$ and combining them together we obtain

$$\frac{\ddot{y}}{\omega_c}^2 + 2R\frac{\dot{y}}{\omega_c} + y = x \tag{4.2}$$

or

$$\ddot{y} + 2R\omega_c\dot{y} + \omega_c^2 y = \omega_c^2 x \tag{4.3}$$

In a similar fashion one can easily obtain the transfer functions for the output signals in Fig. 4.1. Assuming unit cutoff and complex exponential signals, we have

$$\begin{aligned} y_{\mathrm{HP}} &= x - 2Ry_{\mathrm{BP}} - y_{\mathrm{LP}} \\ y_{\mathrm{BP}} &= \frac{1}{s}y_{\mathrm{HP}} \\ y_{\mathrm{LP}} &= \frac{1}{s}y_{\mathrm{BP}} \end{aligned}$$

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

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

Thus

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

Notice that $y_{\rm LP}(t)+2Ry_{\rm BP}(t)+y_{\rm HP}(t) = x(t)$, that is, the input signal is split into lowpass, bandpass and highpass components. The same can be expressed in the transfer function form:

$$H_{\rm LP}(s) + 2RH_{\rm BP}(s) + H_{\rm HP}(s) = 1 \tag{4.4}$$

Amplitude responses

The amplitude responses of the state-variable filter are plotted in Figs. 4.3, 4.4 and 4.5. The pass-, stop- and transition bands of the low- and high-pass filters are defined in the same manner as for the 1-poles, where the transition band now can contain a peak in the amplitude response. For the bandpass the passband is located in the middle (around the cutoff), and there is a stop- and a transition band on each side of the cutoff. The slope rolloff speed is obviously −12dB/oct for the low- and high-pass, and −6dB/oct for the bandpass.

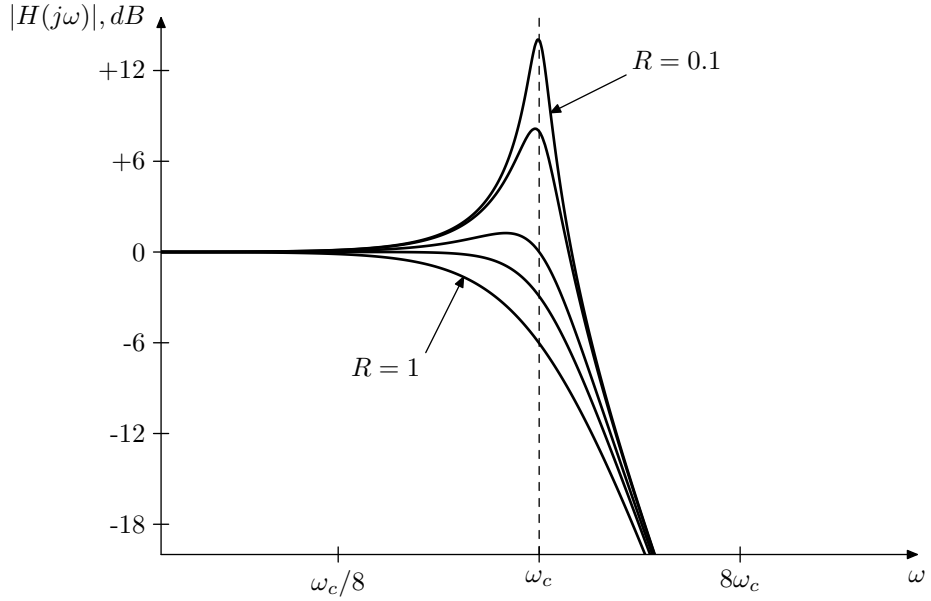


Figure 4.3: Amplitude response of a 2-pole lowpass filter.

One could observe that the highpass response is a mirrored version of the lowpass response, while the bandpass response is symmetric by itself. The symmetry between the lowpass and the highpass amplitude responses has a clear algebraic explanation: applying the LP to HP substitution to a 2-pole lowpass produces a 2-pole highpass and vice versa. The symmetry of the bandpass amplitude response has the same explanation: applying the LP to HP substitution to the 2-pole bandpass converts it into itself.

Since

$$|s^2 + 2Rs + 1|\Big|_{s=j} = |-1 + 2Rj + 1| = 2R$$

the amplitude response at the cutoff is $1/2R$ for all three filter types. Except for the bandpass, the cutoff point $\omega = 1$ is not exactly the peak location but it's

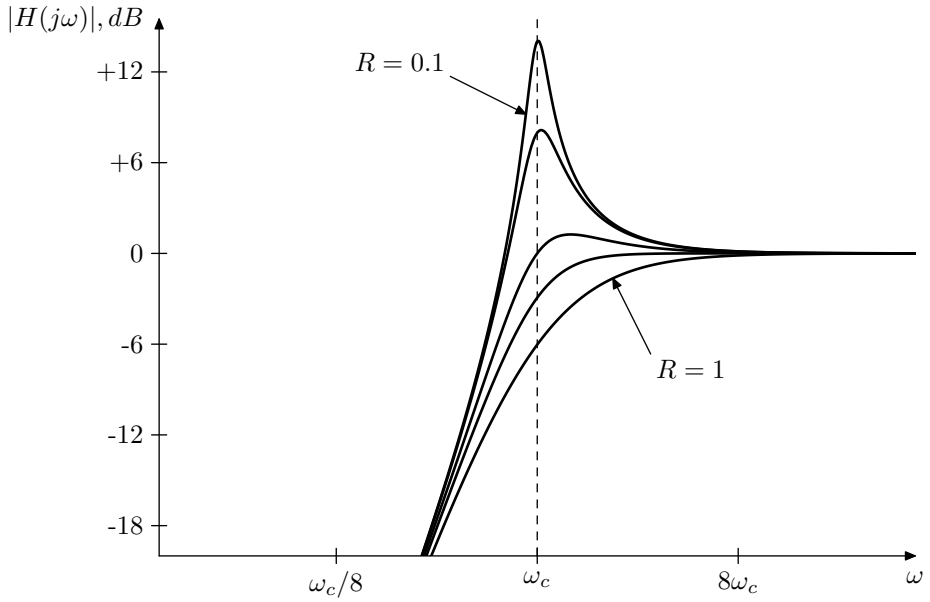


Figure 4.4: Amplitude response of a 2-pole highpass filter.

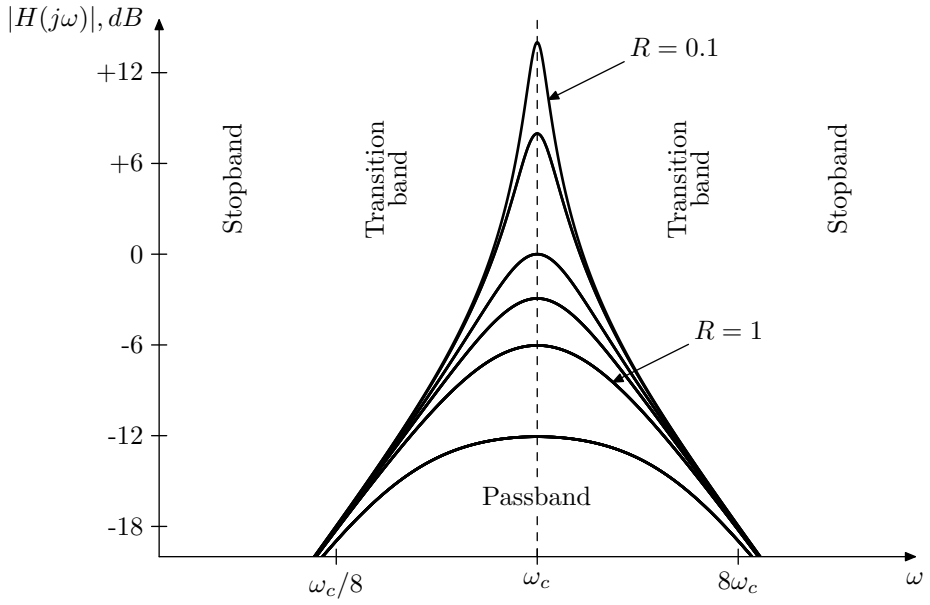


Figure 4.5: Amplitude response of a 2-pole bandpass filter.

pretty close (the smaller the value of R, the closer is the true peak to $\omega = 1$).

Phase responses

The phase response of the lowpass is

$$\arg H_{\mathrm{LP}}(j\omega) = \arg \frac{1}{1 + 2Rj\omega - \omega^2} = -\arg(1 + 2Rj\omega - \omega^2) =$$
$$= -\arctan \frac{2R\omega}{1-\omega^2} = -\operatorname{arccot} \frac{1-\omega^2}{2R\omega} = -\operatorname{arccot} \frac{\omega^{-1} - \omega}{2R} \quad (4.5)$$

where we had to switch from arctan to arccot, since the principal value of arctan gives wrong results for $\omega > 1$. Fig. 4.6 illustrates.

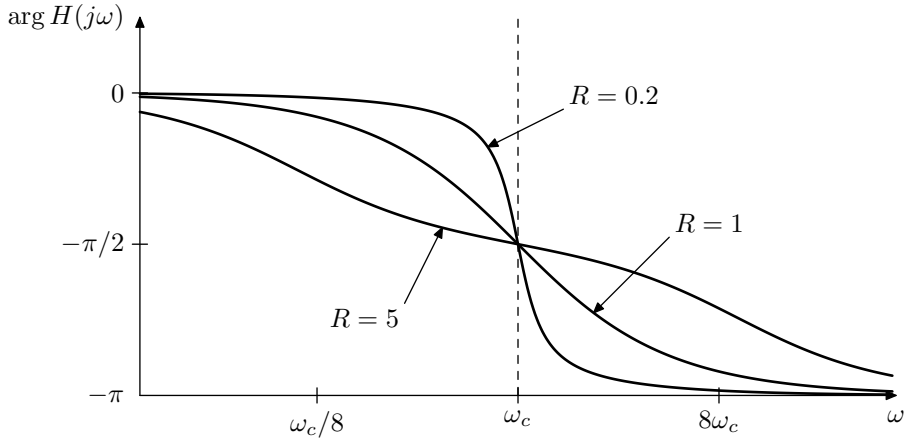


Figure 4.6: Phase response of a 2-pole lowpass filter. Bandpass and highpass responses are the same, except that they are shifted by +90° and 180° respectively.

We could notice the 2-pole phase response has the same kind of symmetry around the cutoff point in the logarithmic frequency scale as the 1-pole filters. This property can be explained from (4.5) by noticing that the substitution $\omega \leftarrow 1/\omega$ changes the sign of the argument of arccot and by using the property of arccot

$$\operatorname{arccot} x + \operatorname{arccot}(-x) = \pi$$

We also could notice that the steepness of the phase response is affected by the parameter R. Explicitly writing the phase response in a logarithmic frequency scale we have

$$\arg H_{\mathrm{LP}}(je^x) = -\operatorname{arccot} \frac{e^{-jx} - e^{jx}}{2R} = -\operatorname{arccot} \frac{-\sinh x}{R} \quad (4.6)$$

thus R simply scales the argument of arccot which results in stretching or shrinking of the phase response.

The bandpass phase response is a +90°-shifted lowpass response:

$$\arg H_{\mathrm{BP}}(j\omega) = \arg \frac{j\omega}{1 + 2Rj\omega - \omega^2} = \frac{\pi}{2} + \arg H_{\mathrm{LP}}(s)$$

The bandpass phase response is a 180°-shifted lowpass response:

$$\arg H_{\mathrm{HP}}(j\omega) = \arg \frac{(j\omega)^2}{1 + 2Rj\omega - \omega^2} = \pi + \arg H_{\mathrm{LP}}(s)$$

The phase response at the cutoff is $-90°$ for the lowpass:

$$\arg H_{\mathrm{LP}}(j) = \arg \frac{1}{1 + 2Rj - 1} = \arg \frac{1}{2Rj} = -\frac{\pi}{2}$$

respectively giving 0° for the bandpass and +90° for the highpass.

It can be also observed in Fig. 4.6 that the lowpass phase response is close to zero in the passband, the same as for the 1-pole lowpass. As we shuld have expected, the same also holds for the highpass's passband. Somewhat remarkably, as we just established by evaluating the bandpass phase response at the cutoff, the same property also holds for the bandpass's passpand, although at small values of R the phase will be close to zero only in a small neighborhood of the cutoff.

4.2 Resonance

With a 1-pole lowpass or highpass filter, the only parameter to control was the filter cutoff, shifting the amplitude response to the left or to the right in the logarithmic frequency scale. With 2-pole filters there is an additional parameter R, which, as the reader could have noticed from Figs. 4.3, 4.4 and 4.5 controls the height of the amplitude response peak occuring closely to $\omega = \omega_c$. A narrow peak in the amplitude response is usually referred to as *resonance*. Thus, we can say that the R parameter controls the amount of resonance in the filter.

On the other hand, from the same figures we can notice that the resonance increases (the peak becomes higher and more narrow) as R decreases. It is easy to verify that at $R = 0$ the resonance peak becomes infinitely high. A little bit later we will also establish the fact that the state variable filter is stable if and only if $R > 0$. Thus, the parameter R actually has the function of decreasing or *damping* the resonance. For that reason we refer to the R parameter as the *damping*.[1] By controlling the damping parameter we effectively control the filter's resonance.[2]

Damping and selfoscillation

At $R = 0$ and $x(t) \equiv 0$ the equation (4.3) turns into

$$\ddot{y} = -\omega_c^2 y$$

[1] A more correct term, used in theory of harmonic oscillations, is *damping ratio*, where the commonly used notation for the same parameter is ζ.

[2] The "resonance" control for the SVF filter can be introduced in a number of different ways. One common approach is to use the parameter $Q = 1/2R$, however this doesn't allow to go easily into the selfoscillation range in the nonlinear versions of this filter, also the math is generally more elegant in terms of R than in terms of Q. Another option is using $r = 1 - R$, which differs from the resonance control parameter k of SKF/TSK filters (discussed in Section 5.8) just by a factor of 2, the selfoscillation occuring at $r = 1$. Other, more sophisticated mappings, can be used for a "more natural feel" of the resonance control.

which is effectively a spring-mass equation

$$m\ddot{y} = -ky$$

or

$$\ddot{y} = -\frac{k}{m}y$$

where respectively $\omega_c = \sqrt{k/m}$. Starting from a non-zero initial state such system will oscillate around the origin infinitely long. Thus, in the absence of the damping signal path (Fig. 4.7), the filter will be constantly *selfoscillating*.[3] Notably, the selfoscillation is appearing at the setting $R = 0$ where the resonance peak is getting infinitely high. This is a general property of resonating filters and has to do with the relationship between the filter poles and the filter's transient response, both covered later in this chapter and additionally and in a more general form in Chapter 7.

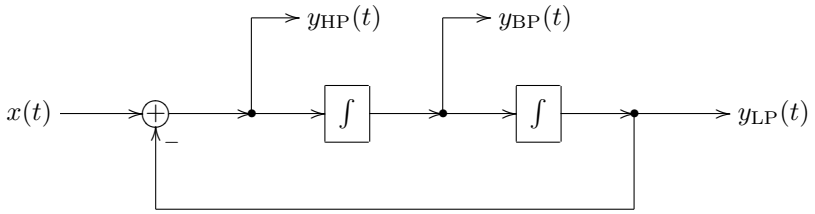


Figure 4.7: 2-pole multimode state-variable filter without the damping path (selfoscillating).

The introduction of the damping signal

$$\ddot{y} = -\omega_c^2 y - 2R\dot{y}$$

reduces the amount of resonance in the filter, which in terms of a spring-mass system works as a 1st-order energy dissipation term:

$$m\ddot{y} = -ky - 2c\dot{y}$$

This should give a better idea of why the R parameter is referred to as damping.

By further adding an external force to the spring-mass system one effectively adds the input signal.[4]

[3] The selfoscillating state at $R = 0$ is a marginally stable state. As mentioned earlier, due to the noise present in the system (such as numerical errors in a digital implementation), we shouldn't expect to be able to exactly hold a system in a marginally stable state. In order to have reliable selfoscillation one usually needs to introduce nonlinear elements into the system. E.g. by introducing the saturating behavior one would be able to lower R below 0, thereby increasing the resonance even further, without making the filter explode. So, while selfoscillation formally appears at $R = 0$, it is becoming reliable at $R < 0$, given that nonlinearities prevent the filter from exploding.

[4] Thereby the differential equation becomes formally equivalent to an SVF, but there still is an essential difference. The state of a spring-mass system consists of a position $y(t)$ and a velocity $\dot{y}(t)$. Changes to the system parameters will therefore directly change the kinetic and potential energies, which can result in a sudden increase or reduction of the amplitude of the

Resonance peak

We can find the exact position and the height of the resonance peak by looking for the local maximum of the (squared) amplitude response. E.g. for the lowpass amplitude response:

$$|H_{\mathrm{LP}}(j\omega)|^2 = \frac{1}{|(j\omega)^2 + 2Rj\omega + 1|^2} = \frac{1}{(1-\omega^2)^2 + 4R^2\omega^2}$$

Instead of looking for the maximum of $|H_{\mathrm{LP}}(j\omega)|^2$ we can look for the minumum of the reciprocal function:

$$|H_{\mathrm{LP}}(j\omega)|^{-2} = \omega^4 + 2(2R^2 - 1)\omega^2 + 1$$

Clearly, $|H_{\mathrm{LP}}(j\omega)|^{-2}$ is a quadratic polynomial in ω^2 with the minimum at $\omega^2 = 1 - 2R^2$. The resonance peak position is thus

$$\omega_{\mathrm{peak}} = \sqrt{1 - 2R^2}$$

where for $R \geq 1/\sqrt{2}$ (we are considering only positive values of R) there is no minimum at $\omega^2 > 0$ and respectively no resonance peak. Note that the peak thereby starts at $\omega = 0$ at $R = 1/\sqrt{2}$ and, as R decreases to zero, moves towards $\omega = 1$.

The resonance peak height is simply the value of the amplitude response evaluated at ω_{peak}:

$$\begin{aligned} |H_{\mathrm{LP}}(j\omega_{\mathrm{peak}})|^2 &= \frac{1}{(1-(1-2R^2))^2 + 4R^2(1-2R^2)} = \frac{1}{4R^4 + 4R^2 - 8R^4} = \\ &= \frac{1}{4R^2 - 4R^4} = \frac{1}{4R^2(1-R^2)} \qquad (R < 1/\sqrt{2}) \end{aligned}$$

and

$$|H_{\mathrm{LP}}(j\omega_{\mathrm{peak}})| = \frac{1}{2R\sqrt{1-R^2}} \qquad (R < 1/\sqrt{2})$$

Thus at $R = 1/\sqrt{2}$ the peak height is formally $|H_{\mathrm{LP}}(j\omega_{\mathrm{peak}})| = 1$, corresponding to the amplitude response not having the peak yet. At $R \to 0$ we have $|H_{\mathrm{LP}}(j\omega_{\mathrm{peak}})| \sim 1/2R$. The above expression also allows us to find the value of R given a desired peak height A. Starting from

$$A = \frac{1}{2R\sqrt{1-R^2}} \tag{4.7}$$

we have

$$\begin{aligned} 2R\sqrt{1-R^2} &= A^{-1} \\ R^2(1-R^2) &= \frac{A^{-2}}{4} \\ R^4 - R^2 + \frac{A^{-2}}{4} &= 0 \end{aligned}$$

swinging. In comparison, in the SVF the system state consists of the "lowpass" integrator's state $y(t)$ and "bandpass" integrator's state, which according to (4.1) is $\dot{y}(t)/\omega_c$. In this case changes to the filter parameters will affect the filter's output in a more gradual way. Particulary, according to (2.28), changes to the cutoff will not affect the output amplitude at all.

$$R^2 = \frac{1 \pm \sqrt{1 - A^{-2}}}{2}$$

Taking into account the allowed range of R (which is $0 < R < 1/\sqrt{2}$), we obtain

$$R = \sqrt{\frac{1 - \sqrt{1 - A^{-2}}}{2}} \qquad (A \geq 1) \tag{4.8}$$

Recalling that the amplitude response of the 2-pole highpass is simply a symmetrically flipped amplitude response of the 2-pole lowpass, we realize that the same considerations apply to the 2-pole highpass, except that the expression for ω_{peak} needs to be reciprocated. For the bandpass filter the amplitude response peak is always exactly at the cutoff.

Butterworth filter

The threshold value $R = 1/\sqrt{2}$ at which the resonance peak starts to appear has another interesting property. At this setting the (logarithmic frequency scale) amplitude responses of the 2-pole lowpass and highpass are shrunk horizontally two times around the cutoff point, as compared to those of 1-poles (the phase response is transformed in a more complicated way, which is of little interest to us here). This is a particular case of a *Butterworth filter*. Butterworth filters will be discussed in a generalized form in Chapter 8, but we can also show this shrinking property explicitly here. Indeed, for $R = 1/\sqrt{2}$ we have

$$\left|s^2 + \sqrt{2} \cdot s + 1\right|^2 \Big|_{s=j\omega} = \left|1 - \omega^2 + j\sqrt{2} \cdot \omega\right|^2 = (1 - \omega^2)^2 + 2\omega^2 =$$

$$= 1 + \omega^4 = \left|1 + j\omega^2\right|^2 = |1 + s|^2 \Big|_{s=j\omega^2}$$

Now, the substitution $\omega \leftarrow \omega^2$ corresponds to the two times shrinking in the logarithmic frequency scale: $\log \omega \leftarrow 2 \log \omega$. Thus, for the lowpass 2-pole we have

$$\left|\frac{1}{s^2 + \sqrt{2} \cdot s + 1}\right| \Big|_{s=j\omega} = \left|\frac{1}{1 + s}\right| \Big|_{s=j\omega^2}$$

and for the highpass filter we have

$$\left|\frac{s^2}{s^2 + \sqrt{2} \cdot s + 1}\right| \Big|_{s=j\omega} = \left|\frac{s}{1 + s}\right| \Big|_{s=j\omega^2}$$

The readers can refer to Fig. 8.13 for the illustration of the shrinking effect. Since for $R < 1/\sqrt{2}$ the amplitude response obtains a resonance peak, the Butterworth 2-pole filter is the one with the "sharpest" possible cutoff among all non-resonating 2-poles.

4.3 Poles

Solving $s^2 + 2Rs + 1 = 0$ we obtain the poles of the filter at

$$p_{1,2} = -R \pm \sqrt{R^2 - 1} = \begin{cases} -R \pm \sqrt{R^2 - 1} & \text{if } |R| \geq 1 \\ -R \pm j\sqrt{1 - R^2} & \text{if } -1 \leq R \leq 1 \end{cases}$$

Thus, the poles are located in the left semiplane if and only if $R > 0$. As with 1-poles, the location of the poles in the left semiplane is sufficient and necessary for the filter to be stable.[5]

For $|R| \leq 1$ the poles are located on the unit circle

$$(\mathrm{Re}\, p)^2 + (\mathrm{Im}\, p)^2 = (-R)^2 + (\sqrt{1 - R^2})^2 = 1$$

This also implies that R is equal to the cosine of the angle between the negative real axis and the direction to the pole (Fig. 4.8).

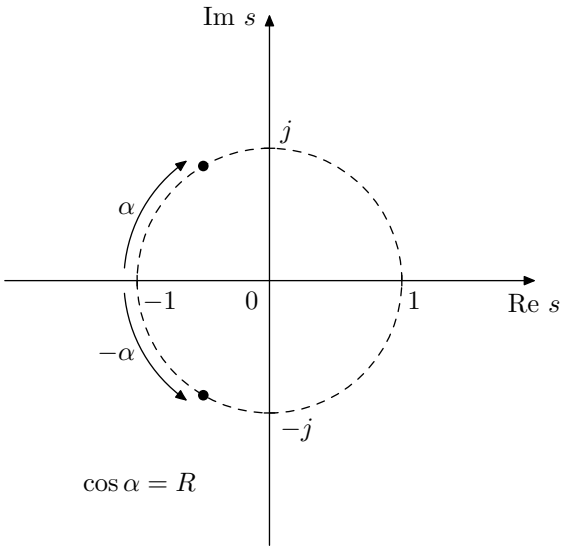


Figure 4.8: Poles of a resonating 2-pole filter ($\omega_c = 1$).

As R is getting close to zero, the poles are getting close to the imaginary axis. By definition of a pole, the transfer function is infinitely large at the poles, which means it is also having large values on the imaginary axis close to the poles. This corresponds to the resonance peak appearing in the amplitude response. At $R = 0$ the poles are located right on the imaginary axis and the filter selfoscillates.

At $|R| \geq 1$ the poles are real and mutually reciprocal:[6]

$$(-R - \sqrt{R^2 - 1}) \cdot (-R + \sqrt{R^2 - 1}) = 1$$

(Fig. 4.9). The filter thus "falls apart" into a serial combination of two 1-pole filters:

$$H_{\mathrm{LP}}(s) = \frac{1}{s^2 + 2Rs + 1} = \frac{1}{s - p_1} \cdot \frac{1}{s - p_2}$$

[5] Later we will discuss the transient response of the SVF and the respective effects of the poles position on the stability.

[6] Actually, the poles are mutually reciprocal at any R (since their product should be equal to the constant term of the denominator). For complex poles the reciprocal property manifests itself as conjugate symmetry of the poles, since the poles are lying on the unit circle and the reciprocation does not change their absolute magnitude.

$$H_{\mathrm{BP}}(s) = \frac{s}{s^2 + 2Rs + 1} = \frac{s}{s - p_1} \cdot \frac{1}{s - p_2}$$
$$H_{\mathrm{HP}}(s) = \frac{s^2}{s^2 + 2Rs + 1} = \frac{s}{s - p_1} \cdot \frac{s}{s - p_2}$$

where $p_1 p_2 = 1$.[7] These 1-pole filters become visible in the amplitude responses at sufficiently large R as two different "cutoff points" (Fig. 4.10).

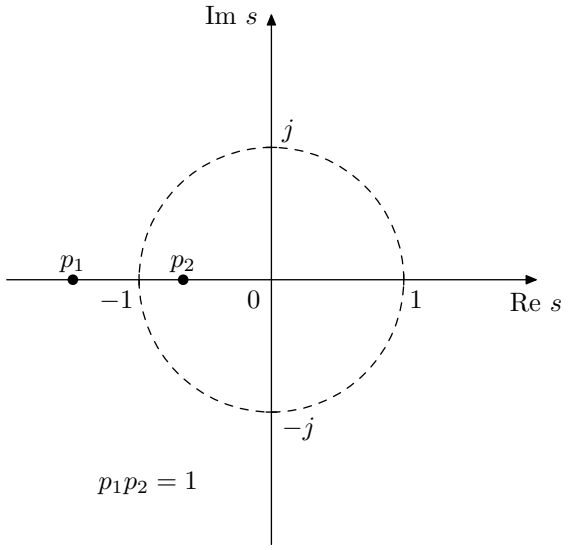


Figure 4.9: Poles of a non-resonating 2-pole filter ($\omega_c = 1$).

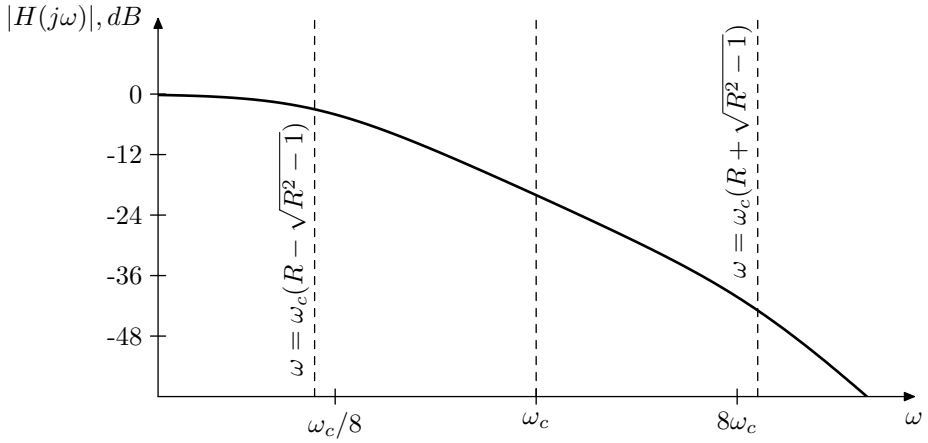


Figure 4.10: Amplitude response of a non-resonating 2-pole low-pass filter.

[7] Of course the same decomposition is formally possible for complex poles, but a 1-pole filter with a complex pole cannot be implemented as a real system.

Resonance redefined

The pole positions can give us another way of defining the point where we consider the resonance to appear. Previously we have found that the resonance peak appears at $R < 1/\sqrt{2}$. However, the amplitude response peak is only one manifestation of the resonance effect. Another aspect of resonance is that, as we shall see later, the transient response of the filter contains sinusoidal oscillation, which occurs whenever the poles are complex. Therefore, using the presence of transient oscillations as the alternative definition of the resonance, we can say that the resonance occurs when $R < 1$.

Similarly to $R = 1/\sqrt{2}$, the threshold setting $R = 1$ has a special property. At this setting both poles are located at $s = -1$ and the transfer function of the 2-pole lowpass becomes equal to the transfer function of two serially connected 1-pole lowpasses:

$$\frac{1}{s^2 + 2s + 1} = \left(\frac{1}{s+1}\right)^2$$

while the transfer function of the 2-pole highpass becomes equal to the transfer function of two serially connected 1-pole highpasses:

$$\frac{s^2}{s^2 + 2s + 1} = \left(\frac{s}{s+1}\right)^2$$

This means that at this value of R the (decibel-scale) amplitude responses of the 2-pole lowpass and highpass are stretched vertically two times compared to those of the 1-pole lowpass and highpass (Fig. 4.11), and the same holds for the phase responses (Fig. 4.12).

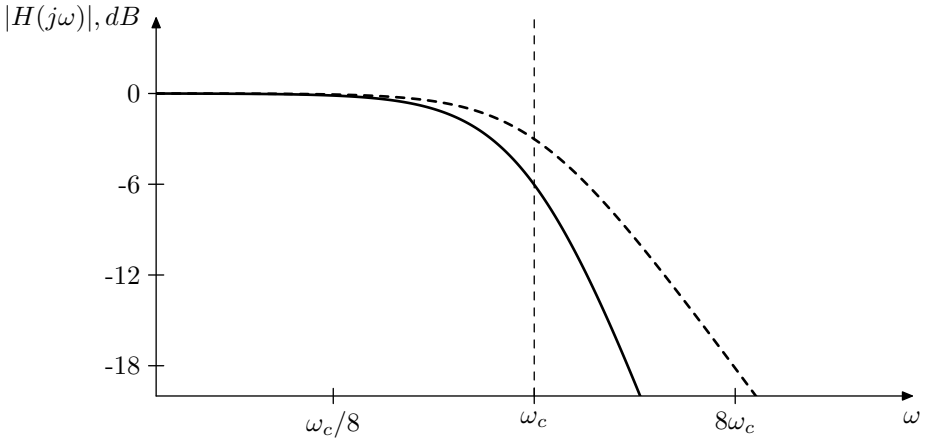


Figure 4.11: Amplitude response of the 2-pole lowpass filter at $R = 1$ (solid line) compared to the amplitude response of the 1-pole lowpass filter (dashed line).

Non-unit cutoff

If $\omega_c \neq 1$ then the transfer function denominator becomes $s^2 + 2R\omega_c s + \omega_c^2$ (or $(s/\omega_c)^2 + 2Rs/\omega_c + 1$, if no simplifications are performed on the entire transfer

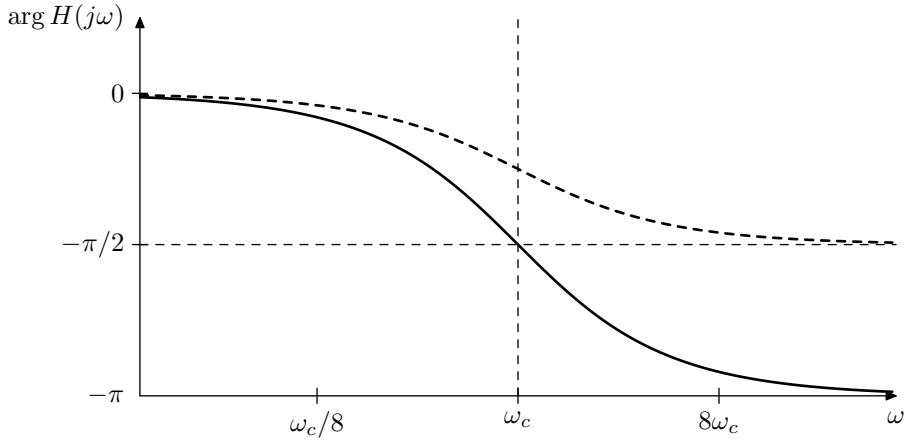


Figure 4.12: Phase response of the 2-pole lowpass filter at $R = 1$ (solid line) compared to the amplitude response of the 1-pole lowpass filter (dashed line).

function) and the formula for the poles becomes

$$p_{1,2} = \omega_c \cdot (-R \pm \sqrt{R^2 - 1}) = \begin{cases} \omega_c \cdot (-R \pm \sqrt{R^2 - 1}) & \text{if } |R| \geq 1 \\ \omega_c \cdot (-R \pm j\sqrt{1 - R^2}) & \text{if } -1 \leq R \leq 1 \end{cases} \tag{4.9}$$

The formula (4.9) can be obtained either by directly solving the quadratic equation or by noticing that the cutoff substitution $s \leftarrow s/\omega_c$ scales the poles according to $p \leftarrow p\omega_c$. Complex poles are therefore located on the circle of radius ω_c (Fig. 4.13), while real poles have a geometric mean equal to ω_c (Fig. 4.14).

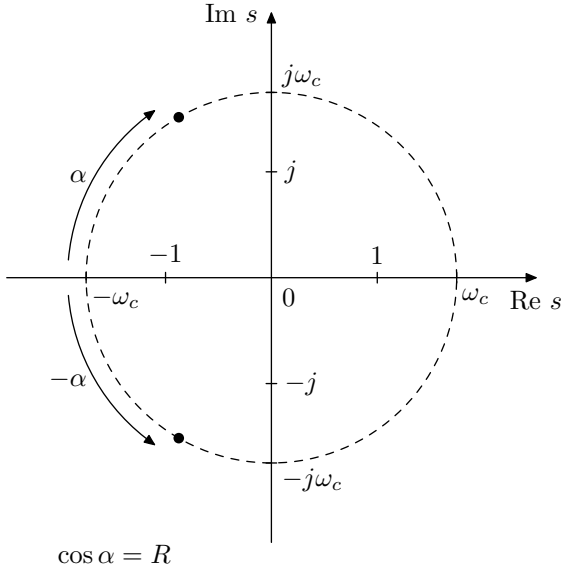


Figure 4.13: Poles of a resonating 2-pole filter ($\omega_c \neq 1$).

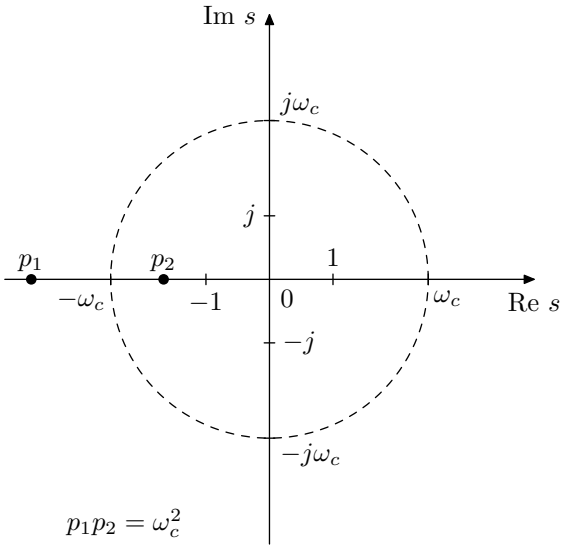


Figure 4.14: Poles of a non-resonating 2-pole filter ($\omega_c \neq 1$).

Transfer function in terms of poles

Writing the lowpass transfer funtion in terms of poles we have for $\omega_c = 1$

$$H_{\mathrm{LP}}(s) = \frac{1}{s^2 + 2Rs + 1} = \frac{1}{s - p_1} \cdot \frac{1}{s - p_2} = \frac{1}{s^2 - (p_1 + p_2)s + 1}$$

and for an arbitrary ω_c respectively

$$H_{\mathrm{LP}}(s) = \frac{\omega_c^2}{s^2 + 2R\omega_c s + \omega_c^2} = \frac{p_1 p_2}{s^2 - (p_1 + p_2)s + p_1 p_2}$$

Respectively

$$-(p_1 + p_2) = 2R\omega_c \tag{4.10a}$$

$$p_1 p_2 = \omega_c^2 \tag{4.10b}$$

from where

$$\omega_c = \sqrt{p_1 p_2} \tag{4.11a}$$

$$R = -\frac{p_1 + p_2}{2\omega_c} = -\frac{(p_1 + p_2)/2}{\sqrt{p_1 p_2}} \tag{4.11b}$$

In terms of $\omega_1 = -p_1$ and $\omega_2 = -p_2$ the same turns into

$$\omega_1 + \omega_2 = 2R\omega_c$$

$$\omega_1 \omega_2 = \omega_c^2$$

and

$$\omega_c = \sqrt{\omega_1 \omega_2} \tag{4.12a}$$

$$R = \frac{\omega_1 + \omega_2}{2\omega_c} = \frac{(\omega_1 + \omega_2)/2}{\sqrt{\omega_1\omega_2}} \tag{4.12b}$$

Notice that thereby ω_c is a geometric mean of the 1-pole cutoffs, and R is a ratio of their arithmetic and geometric means. Equations (4.12) can be used to represent a series of two 1-poles with given cutoffs by an SVF.[8]

Pole cutoff and damping

A pair of complex poles of an SVF must be a conjugate pair, therefore we have $|p_1| = |p_2|$, $\operatorname{Re} p_1 = \operatorname{Re} p_2$ and $\operatorname{Im} p_1 = -\operatorname{Im} p_2$. The equations (4.10) in this case turn into

$$\begin{aligned} -2\operatorname{Re} p_n &= 2R\omega_c \\ |p_n|^2 &= \omega_c^2 \end{aligned} \qquad (n = 1, 2)$$

These relationships motivate the introduction of the notion of the "associated cutoff and damping" of an arbitrary pair of conjugate poles p and p^*, where we would have

$$\begin{aligned} -2\operatorname{Re} p &= 2R\omega_c \\ |p|^2 &= \omega_c^2 \end{aligned} \qquad (n = 1, 2)$$

and

$$\begin{aligned} \omega_c &= |p| \\ R &= \frac{-\operatorname{Re} p}{|p|} \end{aligned} \qquad (n = 1, 2) \tag{4.13}$$

(Fig. 4.13 can serve as an illustration).

This idea is particularly convenient, if we imply that a particular high-order transfer function is to be implemented as a cascade of 2-poles (further discussed in Section 8.2), in which case (4.13) gives us ready formulas for the computation of the cutoff and damping of the respective 2-pole. Also, unless the high-order transfer function is having coinciding complex poles, the separation of complex poles into pairs of conjugate poles is unambiguous.

The same can be done for real poles, if desired, where we can use (4.11) instead of (4.13) but this would work only under the restriction that both poles are having the same sign (Fig. 4.14 can serve as an illustration).[9] Also the grouping of such poles into pairs can be done in different ways.

Sometimes the same terminology is also convenient for zeros. Even though formally it is not correct, since zeros are not directly associated with a cutoff or damping, it is sometimes handy to treat a pair of zeros as roots of a polynomial $s^2 + 2R\omega_c s + \omega_c^2$.

4.4 Digital model

Skipping the naive implementation, which the readers should be perfectly capable of creating and analyzing themselves by now, we proceed with the discussion of the TPT model.

[8] Apparently, (4.12) defines only the denominator of the SVF's transfer function. The numerator would need to be computed separately.

[9] Apparently (4.11) can be used all the time, regarless of whether the poles are complex or real. It's just that in case of complex poles we have simpler and more intuitive formulas (4.13).

Assuming $g\xi+s_n$ instantaneous responses for the two trapezoidal integrators one can redraw Fig. 4.1 to obtain the discrete-time model in Fig. 4.15.

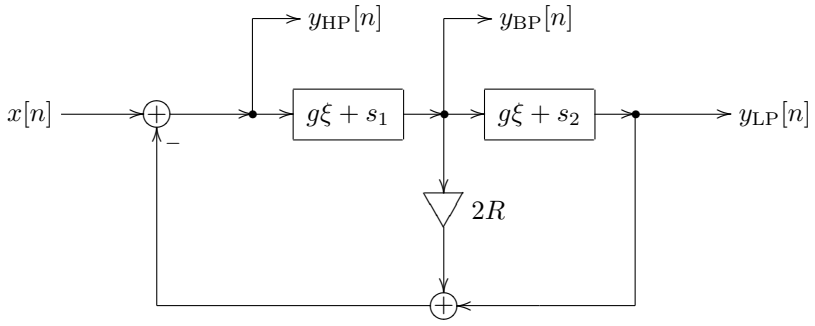


Figure 4.15: TPT 2-pole multimode state-variable filter in the instantaneous response form.

Picking y_{HP} as the zero-delay feedback equation's unknown[10] we obtain from Fig. 4.15:

$$y_{\mathrm{HP}} = x - 2R(gy_{\mathrm{HP}} + s_1) - g(gy_{\mathrm{HP}} + s_1) - s_2$$

from where

$$\left(1 + 2Rg + g^2\right) y_{\mathrm{HP}} = x - 2Rs_1 - gs_1 - s_2$$

from where

$$y_{\mathrm{HP}} = \frac{x - (2R + g)s_1 - s_2}{1 + 2Rg + g^2} \tag{4.14}$$

Apparently (4.14) has the form (3.37), where the total instantaneous gain of the zero-delay feedback loop in Fig. 4.15 is $G = -(2Rg + g^2)$ and thus the instantaneously unstable case occurs when the denominator of (4.14) is negative. However, as long as $g > 0$ and $R > -1$, the denominator of (4.14) is always positive:

$$1 + 2Rg + g^2 > 1 + 2 \cdot (-1) \cdot g + g^2 = (1 - g)^2 \geq 0$$

thus under these conditions the filter in not becoming instantaneously unstable.

Using y_{HP} we can proceed defining the remaining signals in the structure, in the same way as we did for the 1-pole in Section 3.9. Assuming that we are using trasposed direct form II integrators (Fig. 3.11), s_n are the states of the z^{-1} elements in the respective integrators and $g = \omega_c T/2$ (prewarped). Therefore by precomputing the values $1/(1+2Rg+g^2)$ and $2R+g$ in advance, the formula (4.14) can be computed in 2 subtractions and 2 multiplications. What remains is the processing of both integrators. A transposed direct form II integrator can be computed in 1 multiplication and 2 additons. Thus, the entire SVF processing routine needs 4 multiplications and 6 additions/subtractions:

```
// perform one sample tick of the SVF
HP := (x-g1*s1-s2)*d; // g1=2R+g, d=1/(1+2Rg+g^2)
```

[10]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.

```
v1 := g*HP; BP := v1+s1; s1 := BP+v1; // first integrator
v2 := g*BP; LP := v2+s2; s2 := LP+v2; // second integrator
```

If we are not interested in the highpass signal, we could obtain a more optimal implementation by solving for y_{BP} instead:

$$y_{\mathrm{BP}} = g(x - 2Ry_{\mathrm{BP}} - gy_{\mathrm{BP}} - s_2) + s_1$$

$$(1 + 2Rg + g^2)y_{\mathrm{BP}} = g(x - s2) + s_1$$

$$y_{\mathrm{BP}} = \frac{g(x - s2) + s_1}{1 + 2Rg + g^2}$$

This gives us:

```
// perform one sample tick of the SVF BP/LP
BP := (g*(x-s2)+s1)*d; // d=1/(1+2Rg+g^2)
v1 := BP-s1; s1 := BP+v1; // first integrator
v2 := g*BP; LP := v2+s2; s2 := LP+v2; // second integrator
```

This implementation has 3 multiplications and 6 additions/subtractions.

If we need only the BP signal, then we could further transform the expressions used to update the integrators:

```
// perform one sample tick of the SVF BP
BP := (g*(x-s2)+s1)*d; // d=1/(1+2Rg+g^2)
BP2 := BP+BP; s1 := BP2-s1; // first integrator
v22 := g*BP2; s2 := s2+v22; // second integrator
```

That's 3 multiplications and 5 additions/subtractions.

4.5 Normalized bandpass filter

By multiplying the bandpass filter's output by $2R$:

$$H_{\mathrm{BP1}}(s) = 2RH_{\mathrm{BP}}(s) = \frac{2Rs}{s^2 + 2Rs + 1} \tag{4.15}$$

we obtain a bandpass filter which has a unit gain (and zero phase response) at the cutoff:

$$H_{\mathrm{BP1}}(j) = \frac{2Rj}{j^2 + 2Rj + 1} = 1$$

For that reason this version of the 2-pole bandpass filter is referred to as a *unit-gain* or *normalized* bandpass. Fig. 4.16 illustrates the amplitude response.

The normalized bandpass has a better defined passband than the ordinary bandpass, since here we can define the frequency range where $|H_{\mathrm{BP1}}(j\omega)| \approx 1$ as the passband. Notably, in Fig. 4.16 one observes that the width of the passband grows with R. At the same time from Fig. 4.6 one can notice that the width of the band where the bandpass phase response is close to zero also grows with R. Thus, the phase response of the normalized bandpass filter is close to zero in the entire passband of the filter, regardless of R.[11]

[11] This can be confirmed in a more rigorous manner by the fact (which we establish in Section 4.6) that the frequency response of the 2-pole normalized bandpass filter can be obtained from the frequency response of the 1-pole lowpass filter by a frequency axis mapping.

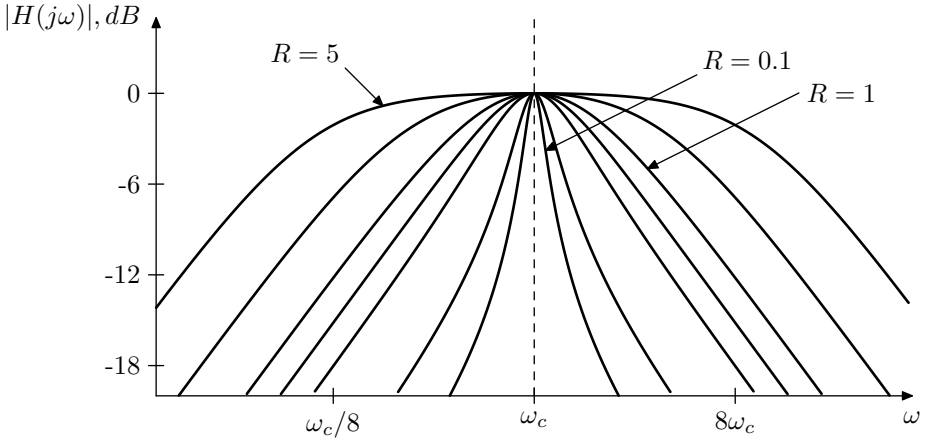


Figure 4.16: Amplitude response of a 2-pole unit gain bandpass filter.

Rewriting (4.4) in terms of the normalized bandpass we get

$$H_{\mathrm{LP}}(s) + H_{\mathrm{BP1}}(s) + H_{\mathrm{HP}}(s) = 1$$

that is

$$x(t) = y_{\mathrm{LP}}(t) + y_{\mathrm{BP1}}(t) + y_{\mathrm{HP}}(t)$$

Topology

Notice that the unit gain bandpass signal can be directly picked up at the output of the $2R$ gain element as shown in Fig. 4.17.

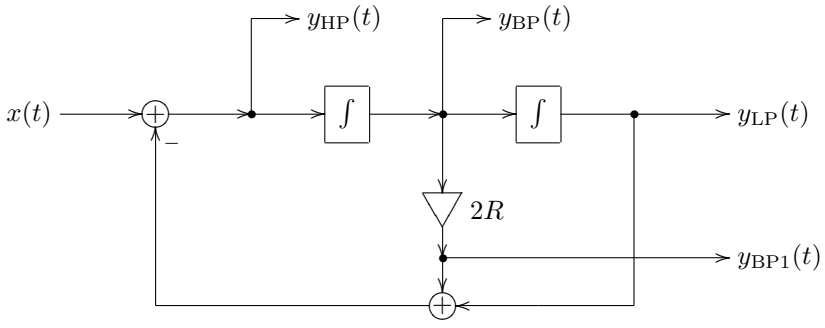
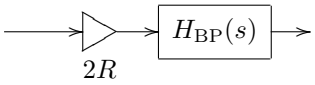


Figure 4.17: State-variable filter with a normalized bandpass output.

If the damping parameter is to be modulated at high rate, rather than multiplying the bandpass output by $2R$, it might be better to multiply the

filter's input by $2R$:



The reasoning is pretty much the same as for positioning the cutoff gains before the integrators or for preferring the transposed (multi-input) filters for modal mixing: we let the integrator smooth the jumps or quick changes in the signal. This will be given for granted if we use the transposed version of Fig. 4.17.

Instead of using the transposed version, we could inject the input signal into the Fig. 4.17 filter structure as shown in Fig. 4.18. However, by multplying the input rather than the output by $2R$ we have not only changed the "BP" output signal to normalized bandpass, we have also changed the amplitudes of the LP and HP outputs. Notably, Fig. 4.18 is essentially the transposed version of Fig. 4.17, except for the relative placement of the second integrator and an invertor.

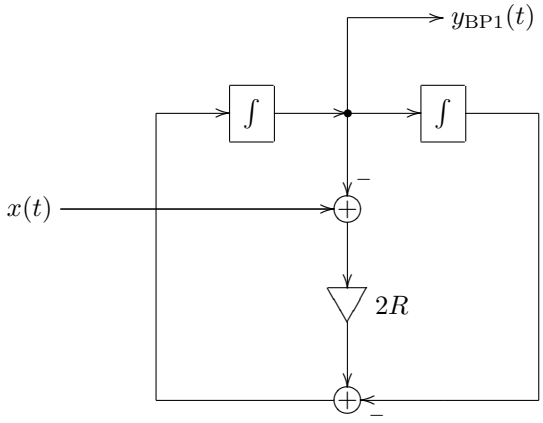


Figure 4.18: Normalized bandpass state-variable filter with pre-filter $2R$ gain.

Prewarping

The standard application of the bilinear transform prewarping technique implies that we want the cutoff point to be positioned exactly at ω_c on the digital frequency axis. However with the normalized bandpass filter the positioning of the left and right transition band slopes is more important than the exact positioning of the cutoff. At the same time, the damping parameter doesn't seem to have much (or any) vertical effect on the amplitude response, mainly controlling the distance between the slopes. Thus we have two degrees of control freedom (the cutoff and the damping) which we could attempt to use to position the two slopes as exactly as possible. Instead of developing the corresponding math just for the normalized bandpass filter, though, we are going to do this in a more general manner in Section 4.6.

4.6 LP to BP/BS substitutions

The 2-pole unit gain bandpass response can be obtained from the lowpass response $1/(1+s)$ by the so-called *LP to BP* (lowpass to bandpass) *substitution*:

$$s \leftarrow \frac{1}{R} \cdot \frac{s + s^{-1}}{2} \tag{4.16}$$

We will also occasionally refer to the LP to BP substitution as the *LP to BP transformation*, making no particular disctinction between both terms.

Since s and $1/s$ are used symmetrically within the right-hand side of (4.16), it immediately follows that the result of the substitution is invariant relative to the LP to HP substitution $s \leftarrow 1/s$. Therefore the result of the LP to BP substitution has an amplitude response which is symmetric in the logarithmic frequency scale.

Using $s = j\omega$, we obtain

$$j\omega \leftarrow \frac{1}{R} \cdot \frac{j\omega + 1/j\omega}{2}$$

or

$$\omega \leftarrow \frac{1}{R} \cdot \frac{\omega - \omega^{-1}}{2}$$

Denoting the new ω as ω' we write

$$\omega = \frac{1}{R} \cdot \frac{\omega' - \omega'^{-1}}{2} \tag{4.17}$$

Instead of trying to understand the mapping of ω to ω' it is easier to understand the inverse mapping from ω' to ω, as explicitly specified by (4.17). Furthermore, it is more illustrative to express ω' in the logarithmic scale:

$$\omega = \frac{1}{R} \cdot \frac{e^{\ln \omega'} - e^{-\ln \omega'}}{2} = \frac{1}{R} \sinh \ln \omega' \qquad \text{if } \omega > 0$$
$$\omega = -\frac{1}{R} \cdot \frac{e^{\ln |\omega'|} - e^{-\ln |\omega'|}}{2} = -\frac{1}{R} \sinh \ln |\omega'| \qquad \text{if } \omega < 0$$

Thus

$$\omega = \frac{1}{R} \sinh \left(\operatorname{sgn} \omega' \cdot \ln |\omega'| \right) \tag{4.18}$$

Since $\ln |\omega'|$ takes up the entire real range of values in each of the cases $\omega > 0$ and $\omega < 0$ and respectively, so does $\sinh(\operatorname{sgn} \omega' \cdot \ln |\omega'|)$,

$$\omega' \in (0, +\infty) \quad \Longleftrightarrow \quad \omega \in (-\infty, +\infty)$$
$$\omega' \in (-\infty, 0) \quad \Longleftrightarrow \quad \omega \in (-\infty, +\infty)$$

This means that the entire range $\omega \in (-\infty, +\infty)$ is mapped once onto the positive frequencies ω' and once onto the negative frequencies ω'. Furthermore, the mapping and its inverse are strictly increasing on each of the two segments $\omega > 0$ and $\omega < 0$, since $\mathrm{d}\omega/\mathrm{d}\omega' > 0$. The unit frequencies $\omega' = \pm 1$ are mapped from $\omega = 0$.

Since we are often dealing with unit-cutoff transfer functions ($\omega_c = 1$), it's interesting to see to which frequencies ω_c' the unit cutoff is mapped. Recalling

that the entire bipolar range of ω is mapped to the positive range of ω', we need to include the negative cutoff point ($\omega_c = -1$) into our transformation. On the other hand, we are interested only in positive ω'_c, since the negative-frequency range of the amplitude response is symmetric to the positive-frequency range anyway. Under these reservations, from (4.18) we have:

$$\frac{1}{R}\sinh\ln\omega'_c = \pm 1$$

from where $\ln\omega'_c = \pm\sinh^{-1} R$, or, changing the logarithm base:

$$\log_2\omega'_c = \pm\frac{\sinh^{-1} R}{\ln 2}$$

Note that the above immediately implies that the two points ω'_c are located at mutually reciprocal positions.

The distance in octaves between the two ω'_c points can be defined as the bandwidth of the transformation:

$$\Delta = \frac{2}{\ln 2}\sinh^{-1} R \tag{4.19}$$

Since the points ω'_c are mutually reciprocal, they are located at $\pm\Delta/2$ octaves from $\omega = 1$.

Inverting (4.19) we can obtain the damping, given the bandwidth Δ:

$$R = \sinh\frac{\Delta\cdot\ln 2}{2} = \frac{2^{\Delta/2} - 2^{-\Delta/2}}{2} \tag{4.20}$$

Frequency axis warping and parameter prewarping

An important consequence of the fact that the LP to BP substitution can be seen as a mapping of the ω axis is that the only effect of the variation of the R parameter is the warping of the frequency axis. This means that (like in the bilinear transform) the amplitude and phase responses are warped identically and the relationship between amplitude and phase responses is therefore preserved across the entire range of ω.

If LP to BP substitution is involved, the resulting frequency response has two points of interest which are the images ω'_1 of the original point at $\omega_1 = 1$, which often is the cutoff point of the original frequency response.[12] Given a digital implementation of such LP to BP substitution's result, we can prewarp the R parameter of the substitution in such a way that the distance between the ω'_1 points in the digital frequency response is identical to the distance between those in analog frequency response.

Indeed, given the original value of R, we can use (4.19) to compute the distance Δ between the ω'_1 points. We know that the points are positioned at $\pm\Delta/2$ octaves from $\omega = 1$, or, if the substitution result has its own cutoff parameter, from ω_c. That is

$$\omega'_1 = \omega_c\cdot 2^{\pm\Delta/2}$$

[12] We are using ω_1 and ω'_1 instead of previously used ω'_c and ω_c notation for the respective point, since we are going to need ω_c to denote the substitution result's cutoff.

So, these are the frequencies at which the unit frequency's image points would be normally located on an analog filter's response and where we want them to be located on the digital filter's response. If ω_1' are the points on the digital frequency response, then by (3.10) the corresponding analog points should be located at

$$\tilde{\omega}_1' = \mu(\omega_1') = \mu\left(\omega_c \cdot 2^{\pm\Delta/2}\right)$$

At unit cutoff $\tilde{\omega}_c$ the points $\tilde{\omega}_1'$ would have been mutually reciprocal. If the cutoff is not unity, then it must be equal to the geometric mean of $\tilde{\omega}_1'$:

$$\tilde{\omega}_c = \sqrt{\mu\left(\omega_c \cdot 2^{\Delta/2}\right) \cdot \mu\left(\omega_c \cdot 2^{-\Delta/2}\right)}$$

while the bandwidth is simply the logarithm of the ratio of $\tilde{\omega}_1'$:

$$\tilde{\Delta} = \log_2 \frac{\mu\left(\omega_c \cdot 2^{\Delta/2}\right)}{\mu\left(\omega_c \cdot 2^{-\Delta/2}\right)}$$

Given $\tilde{\Delta}$, we obtain $\tilde{R}$ from (4.20).

So, we have obtained the *prewarped* parameters $\tilde{\omega}_c$ and $\tilde{R}$, which can be used to control a bilinear transform-based digital implementation of an LP to BP substitution's result, thereby ensuring the correct positioning of the ω_1' points. Particularly, treating the normalized bandpass filter as the result of LP to BP substitution's application to a 1-pole lowpass $1/(1+s)$, we could prewarp the bandpass filter's parameters to have exact positioning of the -3dB points on the left and right slopes (since these are the images of the 1-pole lowpass's unit cutoff point).

In principle, any other two points could have been chosen as prewarping points, where the math is much easier if these two points are located symmetrically relatively to the cutoff in the logarithm frequency scale. We will not go into further detail of this, as the basic ideas of deriving the respective equations are exactly the same.

Poles and stability

The transformation of the poles and zeros by the LP to BP transformation can be obtained from

$$s = \frac{1}{R} \cdot \frac{s' + s'^{-1}}{2} \tag{4.21}$$

resulting in

$$s' = Rs \pm \sqrt{R^2 s^2 - 1}$$

Regarding the stability preservation consider that the sum $(s' + 1/s')$ in (4.21) is located in the same complex semiplane (left or right) as s'. Therefore, as long as $R > 0$, the original value s is located in the same semiplane as its images s'. which implies that the stability is preserved. On the other hand, negative values of R "flip" the stability.

Topological LP to BP substitution

As for performing the LP to BP substitution in a block diagram, differently from the LP to HP substitution, here we don't need differentiators. The substitution

can be performed by replacing all (unit-cutoff) integrators in the system with the structure in Fig. 4.19, thereby substituting $2Rs/(s^2+1)$ for $1/s$, which is algebraically equivalent to (4.16).[13]

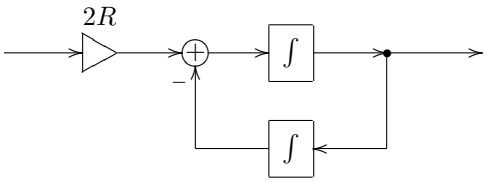


Figure 4.19: "LP to BP" integrator.

LP to BS substitution

The *LP to BS* (lowpass to bandstop) *substitution*[14] is obtained as a series of LP to HP substitution followed by an LP to BP substitution. Indeed, applying the LP to BP substitution to a 1-pole highpass, we obtain the 2-pole notch ("bandstop") filter. Therefore, applying a series of LP to HP and LP to BP substitutions to a 1-pole lowpass we also obtain the 2-pole notch filter.

Combining the LP to HP and LP to BP substitutions expressions in the mentioned order gives an algebraic expression for the LP to BS substitution:

$$\frac{1}{s} \leftarrow \frac{1}{R} \cdot \frac{s + s^{-1}}{2} \tag{4.22}$$

The bandwidth considerations of the LP to BS substitution are pretty much equivalent to those of LP to BP substitution and can be obtained by considering the LP to BS substitution as an LP to BP substitution applied to a result of the LP to HP substitution.

The block-diagram form of the LP to BS substitution can be obtained by directly implementing the right-hand expression in (4.22) as a replacement for the integrators. This however requires a differentiator for the implementation of the s term of the sum.

4.7 Further filter types

By mixing the lowpass, bandpass and highpass outputs one can obtain further filter types. We are now going to discuss some of them.

Often it will be convenient to also include the input signal and the normalized bandpass signal into the set of the mixing sources. Apparently this doesn't bring any new possibilities in terms of the obtained transfer functions, since the input signal can be obtained as a linear combination of LP, BP and HP signals. However the mixing coefficients might look simpler in certain cases. One can

[13] For a differentiator, a similar substitution structure (containing an integrator and a differentiator) is trivially obtained from the right-hand side of (4.16).

[14] Notice that BS here stands for "bandstop" and not for "band-shelving". The alternative name for the substitution could have been "LP to notch", but "LP to bandstop" seems to be commonly used, so we'll stick to that one.

also go further and consider using different topologies implementing a given 2-pole transfer function. Such topologies could differ not only in which specific signals are mixed, but also whether certain mixing coefficients are used at the input or at the output, whether transposed or non-transposed SVF is being used, etc. We won't go here into addressing this kind of detail, however the discussion of the topological aspects of the normalized bandpass in Section 4.5 could serve as an example.

Band-shelving filter

By adding/subtracting the unit gain bandpass signal to/from the input signal one obtains the band-shelving filter (Fig. 4.20):

$$H_{\mathrm{BS}}(s) = 1 + K \cdot H_{\mathrm{BP1}}(s) = 1 + 2RKH_{\mathrm{BP}}(s) = 1 + \frac{2RKs}{s^2 + 2Rs + 1}$$

As with 1-pole shelving we can also specify the shelving boost in decibel:

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

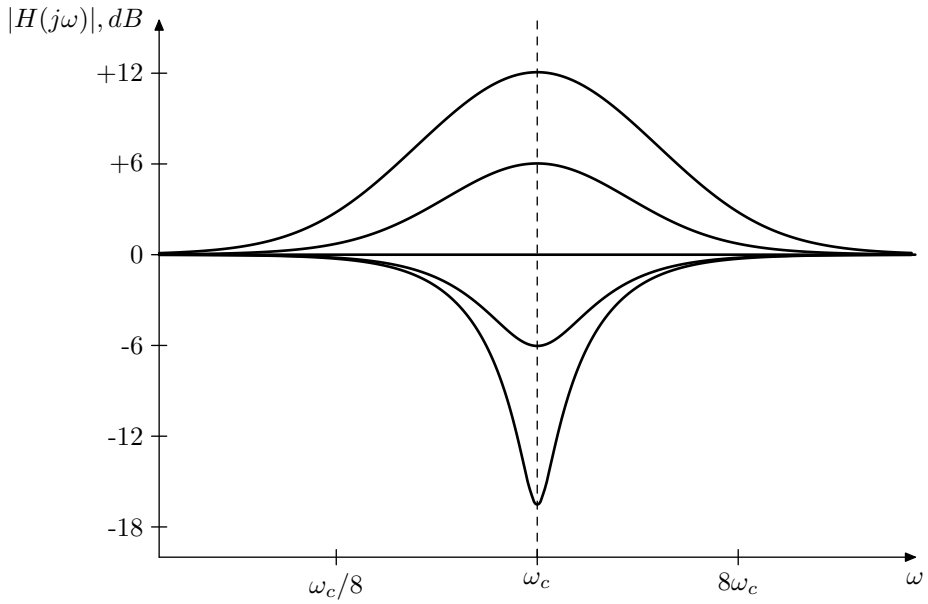


Figure 4.20: Amplitude response of a 2-pole band-shelving filter for $R = 1$ and varying K.

The immediately noticeable problem in Fig. 4.20 is that the bandwidth of the filter varies with the shelving boost K. A way to address this issue will be described in Chapter 10.

Low- and high-shelving filters

Attempting to obtain 2-pole low- and high-shelving filters in a straightforward fashion:

$$H_{\mathrm{LS}}(s) = 1 + K \cdot H_{\mathrm{LP}}(s) \qquad H_{\mathrm{HS}}(s) = 1 + K \cdot H_{\mathrm{HP}}(s)$$

we notice that the amplitude responses of such filters have a strange dip (for $K > 0$) or peak (for $K < 0$) even at a non-resonating setting of $R = 1$ (Fig. 4.21). This peak/dip is due to a steeper phase response curve of the 2-pole lowpass and highpass filters compared to 1-poles. A way to build 2-pole low- and high-shelving filters, which do not have this problem, is described in Chapter 10.

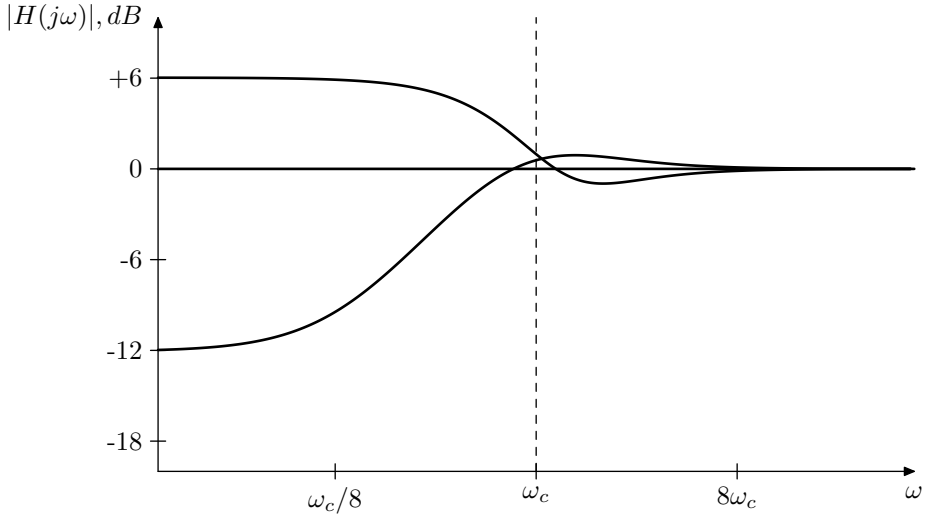


Figure 4.21: Amplitude response of a naive 2-pole low-shelving filter for $R = 1$ and varying K.

Notch filter

At $K = -1$ the band-shelving filter turns into a notch (or bandstop) filter (Fig. 4.22):

$$H_{\mathrm{N}}(s) = 1 - H_{\mathrm{BP1}}(s) = 1 - 2RH_{\mathrm{BP}}(s) = \frac{s^2 + 1}{s^2 + 2Rs + 1}$$

Allpass filter

At $K = -2$ the band-shelving filter turns into an allpass filter (Fig. 4.23):

$$H_{\mathrm{AP}}(s) = 1 - 2H_{\mathrm{BP1}}(s) = 1 - 4RH_{\mathrm{BP}}(s) = \frac{s^2 - 2Rs + 1}{s^2 + 2Rs + 1} \tag{4.23}$$

It is not difficult to show that for purely imaginary s the absolute magintudes of the transfer function's numerator and denominator are equal and thus $|H_{\mathrm{AP}}(j\omega)| = 1$.

We could also notice that the phase respose of the 2-pole allpass is simply the doubled 2-pole lowpass phase response:

$$\begin{aligned}\arg H_{\mathrm{AP}}(j\omega) &= \arg \frac{1 - 2Rj\omega - \omega^2}{1 + 2Rj\omega - \omega^2} = \\ &= \arg(1 - 2Rj\omega - \omega^2) - \arg(1 + 2Rj\omega - \omega^2) =\end{aligned}$$

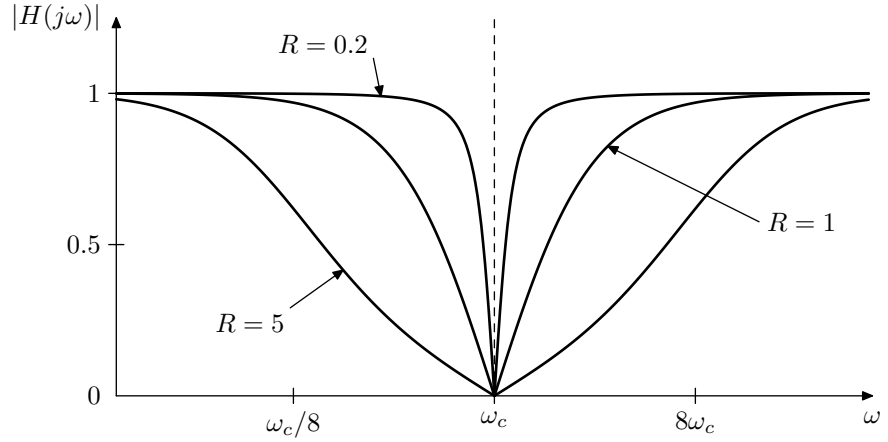


Figure 4.22: Amplitude response of a 2-pole notch filter. The amplitude scale is linear.

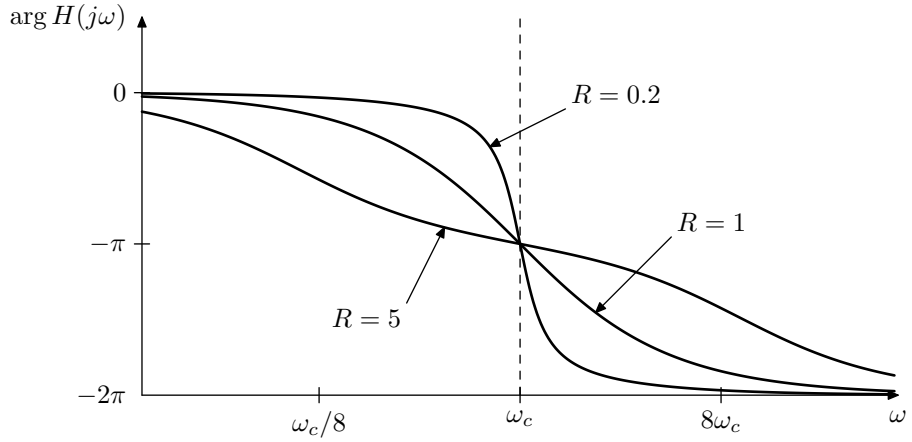


Figure 4.23: Phase response of a 2-pole allpass filter.

$$= -2\arg(1 + 2Rj\omega - \omega^2) = 2\arg H_{\mathrm{LP}}(j\omega) \tag{4.24}$$

Thus the allpass phase response has the same symmetry around the cutoff point and the damping parameter has a similar effect on the phase response slope.

At $R \geq 1$ the 2-pole allpass can be decomposed into the product of 1-pole allpasses:

$$H_{\mathrm{AP}}(s) = \frac{s-\omega_1}{s+\omega_1} \cdot \frac{s-\omega_2}{s+\omega_2} = \frac{\omega_1 - s}{\omega_1 + s} \cdot \frac{\omega_2 - s}{\omega_2 + s}$$

where $\omega_n = -p_n$. At $R = 1$ we have $\omega_1 = \omega_2 = 1$ and the filter turns into the squared 1-pole allpass:

$$H_{\mathrm{AP}}(s) = \left(\frac{s-1}{s+1}\right)^2 = \left(\frac{1-s}{1+s}\right)^2$$

Peaking filter

By subtracting the highpass signal from the lowpass signal (or also vice versa) we obtain the peaking filter (Fig. 4.24):

$$H_{\mathrm{PK}}(s) = H_{\mathrm{LP}}(s) - H_{\mathrm{HP}}(s) = \frac{1 - s^2}{s^2 + 2Rs + 1}$$

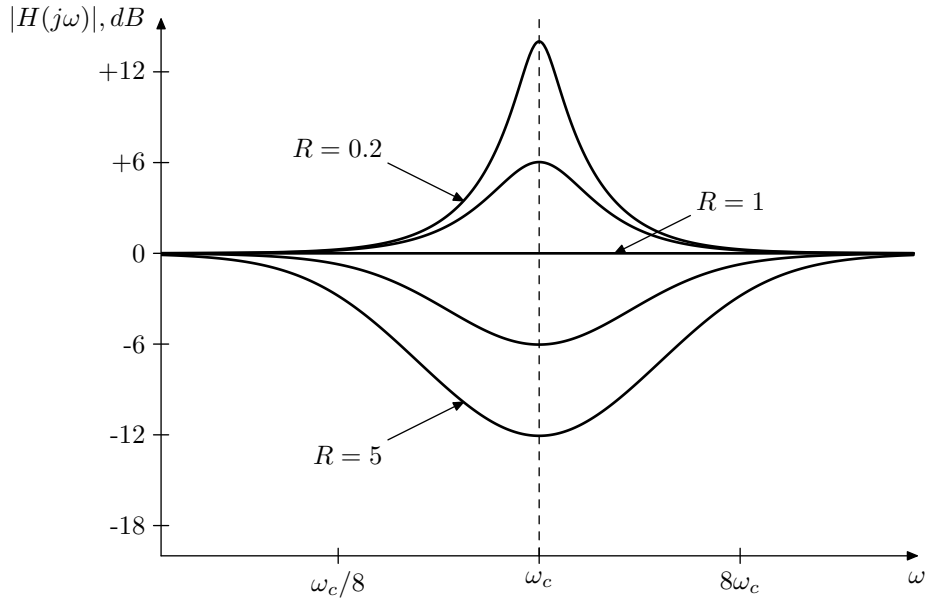


Figure 4.24: Amplitude response of a 2-pole peaking filter.

The peaking filter is a special kind of bandshelving filter. However, as one can see from Fig. 4.24, the bandwidth of the filter varies drastically with R, which often may be undesired. A "properly built" bandshelving filter allows to avoid this problem. This topic is further discussed in Chapter 10.

Arbitrary 2-pole transfer functions

It's easy to see that the state-variable filter can be used to implement any 2nd-order stable differential filter. Indeed, consider the generic 2nd-order transfer function

$$H(s) = \frac{b_2 s^2 + b_1 s + b_0}{s^2 + a_1 s + a_0}$$

where we assume $a_0 > 0$.[15] Then

$$H(s) = \frac{b_2 s^2 + b_1 s + b_0}{s^2 + 2\dfrac{a_1}{2\sqrt{a_0}}\sqrt{a_0}s + \sqrt{a_0}^2} = \frac{b_2 s^2 + b_1 s + b_0}{s^2 + 2R\omega_c s + \omega_c^2} =$$

[15] If $a_0 = 0$, this means that either one or both of the poles of $H(s)$ are at $s = 0$. If $a_0 < 0$ this means that we are having two real poles of opposite signs. Both situations correspond to pretty exotic unstable cases.

$$= b_2 \frac{s^2}{s^2 + 2R\omega_c s + \omega_c^2} + \frac{b_1}{\omega_c} \cdot \frac{\omega_c s}{s^2 + 2R\omega_c s + \omega_c^2} + \frac{b_0}{\omega_c^2} \cdot \frac{\omega_c^2}{s^2 + 2R\omega_c s + \omega_c^2} =$$
$$= b_2 H_{\mathrm{HP}}(s) + \frac{b_1}{\omega_c} H_{\mathrm{BP}}(s) + \frac{b_0}{\omega_c^2} H_{\mathrm{LP}}(s)$$

where we introduced $\omega_c = \sqrt{a_0}$ and $R = a_1/\omega_c$.

4.8 Transient response

In the transient response analysis of the state-variable filter we will concentrate on the lowpass output. The bandpass and highpass can be obtained from the lowpass using (4.1):

$$y_{\mathrm{BP}} = \dot{y}_{\mathrm{LP}}/\omega_c \tag{4.25a}$$
$$y_{\mathrm{HP}} = \dot{y}_{\mathrm{BP}}/\omega_c = \ddot{y}_{\mathrm{LP}}/\omega_c^2 \tag{4.25b}$$

Using (4.10) we rewrite (4.3) in terms of poles, obtaining

$$\ddot{y} - (p_1 + p_2)\dot{y} + p_1 p_2 = p_1 p_2 x \tag{4.26}$$

where $y = y_{\mathrm{LP}}$. Let[16]

$$u_1 = \dot{y} - p_2 y \tag{4.27a}$$
$$u_2 = \dot{y} - p_1 y \tag{4.27b}$$

Therefore

$$\dot{u}_1 = \ddot{y} - p_2 \dot{y}$$
$$\dot{u}_2 = \ddot{y} - p_1 \dot{y}$$

and

$$(\dot{u}_1 + \dot{u}_2) - (p_1 u_1 + p_2 u_2) = (2\ddot{y} - (p_1 + p_2)\dot{y}) - ((p_1 + p_2)\dot{y} - 2p_1 p_2 y) =$$
$$= 2\ddot{y} - 2(p_1 + p)2)\dot{y} + 2p_1 p_2 y \tag{4.28}$$

Noticing that the last expression is simply the doubled left-hand side of (4.26) we obtain an equivalent form of (4.26):

$$(\dot{u}_1 + \dot{u}_2) - (p_1 u_1 + p_2 u_2) = 2p_1 p_2 x \tag{4.29}$$

Splitting the latter in two halves we have:

$$\dot{u}_1 - p_1 u_1 = p_1 p_2 x \tag{4.30a}$$
$$\dot{u}_2 - p_2 u_2 = p_1 p_2 x \tag{4.30b}$$

Adding both equations (4.30) back together, we obtain (4.29), which is equivalent to (4.26). This means that if u_1 and u_2 are solutions of (4.30) then using (4.27) we can find y from u_1 and u_2, which will be the solution of (4.26).

[16]The substitution (4.27) can be obtained, knowing in advance the transient response form $y = C_1 e^{p_1 t} + C_2 e^{p_2 t}$ and expressing $e^{p_n t}$ via y and $\dot{y}$. Alternatively, it can be found by diagonalizing the state-space form.

Now, each of the equations (4.30) is a Jordan 1-pole with input signal p_1p_2x. Applying (2.22) we obtain

$$u_n(t) = u_n(0)e^{p_n t} + p_1p_2 \int_0^t e^{p_n(t-\tau)}x(\tau)\,\mathrm{d}\tau \qquad (n = 1, 2)$$

or, for $x(t) = X(s)e^{st}$, we have from (2.23):

$$u_n(t) = H_n(s)x(t) + \big(u_n(0) - H_n(s)x(0)\big)e^{p_n t} = u_{sn}(t) + u_{tn}(t) \tag{4.31}$$

where

$$H_n(s) = \frac{p_1p_2}{s - p_n}$$

and where $u_{sn}(t)$ and $u_{tn}(t)$ denote the steady-state and transient response parts of $u_n(t)$ respectively. Expressing y via u_n from (4.27) we have

$$y = \frac{u_1 - u_2}{p_1 - p_2} \tag{4.32}$$

For the steady-state response we therefore obtain from (4.31):

$$y_s(t) = \frac{u_{s1} - u_{s2}}{p_1 - p_2} = \frac{H_1(s) - H_2(s)}{p_1 - p_2}x(t) = H(s)x(t)$$

where

$$\begin{aligned} H(s) = \frac{H_1(s) - H_2(s)}{p_1 - p_2} &= \frac{\dfrac{p_1p_2}{s - p_1} - \dfrac{p_1p_2}{s - p_2}}{p_1 - p_2} = \\ &= \frac{p_1p_2}{p_1 - p_2} \cdot \frac{(s - p_2) - (s - p_1)}{s^2 - (p_1 + p_2)s + p_1p_2s} = \\ &= \frac{p_1p_2}{s^2 - (p_1 + p_2)s + p_1p_2s} = \frac{\omega_c^2}{s^2 + 2R\omega_c + \omega_c^2} \end{aligned} \tag{4.33}$$

is the familiar 2-pole lowpass transfer function. The steady-state response $y_s(t)$ is therefore having the same form $H(s)x(t)$ for a complex exponential $x(t) = X(s)e^{st}$ as in case of the 1-pole filter. For signals of general form we respectively obtain the same formula (2.20a) as for 1-poles.

For the transient response we have

$$\begin{aligned} y_t(t) &= \frac{u_{t1} - u_{t2}}{p_1 - p_2} = \\ &= \frac{\dot y(0) - p_2y(0) - H_1(s)x(0)}{p_1 - p_2} \cdot e^{p_1 t} - \frac{\dot y(0) - p_1y(0) - H_2(s)x(0)}{p_1 - p_2} \cdot e^{p_2 t} = \\ &= \frac{\dot y(0) - p_2(y(0) - G_1(s)x(0))}{p_1 - p_2} \cdot e^{p_1 t} - \frac{\dot y(0) - p_1(y(0) - G_2(s)x(0))}{p_1 - p_2} \cdot e^{p_2 t} \end{aligned} \tag{4.34}$$

where we introduce the ordinary (except that p_n may be complex) 1-pole lowpass transfer functions

$$G_n(s) = \frac{-p_n}{s - p_n}$$

Provided $\operatorname{Re} p_{1,2} < 0$ we are having a sum of two exponentially decaying terms. Since $y(0) = y_s(0) + y_t(0)$, the initial value of this sum is $y_t(0) = y(0) - y_s(0)$, the same as in the 1-pole case, so we're having an exponentially decaying discrepancy between the output signal and the steady-state response. However the decaying is now being "distributed" between two exponents $e^{p_1 t}$ and $e^{p_2 t}$. Also notice that while in the 1-pole case the decaying was only affected by the initial state $y(0)$, in the 2-pole case $\dot{y}(0)$ is also a part of the initial state and therefore also affects the decaying shape. Apparently, $y(0)$ is the state of the second ("lowpass") integrator of the SVF, while, according to (4.25a), $\dot{y}(0)$ is essentially the state of the first ("bandpass") integrator.

At $\operatorname{Re} p_{1,2} > 0$ the transient response grows infinitely and the filter explodes.

Steady-state response

In regards to the choice of the steady-state response, there is a similar ambiguity arising out of evaluating the inverse Laplace transform of $H(s)X(s)$ to the left or to the right of the poles of $H(s)$. We won't specifically go into the analysis of this situation for the real poles occurring in the case $|R| > 1$. Complex poles occurring in the case $|R| < 1$ deserve some specical attention.

Apparently $\operatorname{Re} p_1 = \operatorname{Re} p_2$ in this case, and we wish to know how much does the inverse Laplace transform change when we switch the integration path from $\operatorname{Re} s < \operatorname{Re} p_n$ to $\operatorname{Re} s > \operatorname{Re} p_n$. By the residue theorem this change will be equal to the sum of the residues of $H(s)X(s)e^{st}$ at $s = p_1$ and $s = p_2$ respectively, which is

$$\operatorname*{Res}_{s=p_1} H(s)X(s)e^{st} + \operatorname*{Res}_{s=p_2} H(s)X(s)e^{st} = \frac{p_1 p_2}{p_1 - p_2}\left(X(p_1)e^{p_1 t} - X(p_2)e^{p_2 t}\right) \tag{4.35}$$

(where we have used (4.33)). That is we are again obtaining the terms which already exist in the transient response and the integration path choice only affects the amplitudes of the transient response partials, as long as we are staying within the region of convergence of $X(s)$.

The case of coinciding poles requires a separate analysis which can be done as a limiting case $R \to \pm 1$. The respective discussion is occurring later in this section. Even though we don't specifically address the question of evaluation of the inverse Laplace transform in the steady-state response there, it should be clear what the principles would be.

Continuity

Since the input signal of an SVF passes through two integrators on the way to the lowpass output, the lowpass signal should not only always be continuous but should also always have a continuous 1st derivative. Therefore the appearance of $\dot{y}(0)$ besides $y(0)$ in the transient response expression must have somehow taken care of that. Let's verify that this is indeed the case.

Evaluating (4.34) at $t = 0$ using we obtain

$$y_t(0) = \frac{\dot{y}(0) - p_2 y(0) - H_1(s)x(0)}{p_1 - p_2} - \frac{\dot{y}(0) - p_1 y(0) - H_2(s)x(0)}{p_1 - p_2} =$$
$$= y(0) - \frac{H_1(s) - H_2(s)}{p_1 - p_2}x(0) = y(0) - H(s)x(0) = y(0) - y_s(0)$$

where we have used (4.33). Evaluating the derivative of (4.34) at $t = 0$ we obtain

$$\begin{aligned}
\dot{y}_t(0) &= p_1 \frac{\dot{y}(0) - p_2 y(0) - H_1(s)x(0)}{p_1 - p_2} - p_2 \frac{\dot{y}(0) - p_1 y(0) - H_2(s)x(0)}{p_1 - p_2} = \\
&= \dot{y}(0) + \frac{-p_1 H_1(s) + p_2 H_2(s)}{p_1 - p_2} \cdot p_1 p_2 x(0) = \\
&= \dot{y}(0) + \frac{\frac{-p_1}{s - p_1} - \frac{-p_2}{s - p_2}}{p_1 - p_2} \cdot p_1 p_2 x(0) = \\
&= \dot{y}(0) + \frac{(-p_1)(s - p_2) - (-p_2)(s - p_1)}{p_1 - p_2} \cdot \frac{p_1 p_2}{(s - p_1)(s - p_2)} x(0) = \\
&= \dot{y}(0) - \frac{p_1 p_2 \cdot s}{s^2 - (p_1 + p_2)s + p_1 p_2} x(0) = \dot{y}(0) - \frac{\omega^2 s}{s^2 + 2R\omega_c + \omega_c^2} x(0) = \\
&= \dot{y}(0) - \frac{\omega^2}{s^2 + 2R\omega_c + \omega_c^2} \cdot sX(s)e^{st}\Big|_{t=0} = \dot{y}(0) - \dot{y}_s(0)
\end{aligned}$$

which confirms our expectations.

Complex vs. real poles

If $p_{1,2}$ are complex we have

$$e^{p_n t} = e^{t \operatorname{Re} p_n} \cdot (\cos(t \operatorname{Im} p_n) + j \sin(t \operatorname{Im} p_n))$$

The mutual conjugate property of poles will ensure that the two terms of (4.34) are mutually conjugate as well, therefore the addition result is purely real and has the form

$$\begin{aligned}
y_t(t) &= a \cdot e^{t \operatorname{Re} p_1} \cdot \cos\left(|\operatorname{Im} p_1| \cdot t + \varphi\right) = \\
&= a \cdot e^{t \operatorname{Re} p_2} \cdot \cos\left(|\operatorname{Im} p_2| \cdot t + \varphi\right) = \\
&= a \cdot e^{-R\omega_c t} \cdot \cos\left(\omega_c \sqrt{1 - R^2} \cdot t + \varphi\right)
\end{aligned} \tag{4.36}$$

The transient response therefore is a sinusoidal oscillation of frequency $|\operatorname{Im} p_n|$ decaying (or exploding) as $e^{t \operatorname{Re} p_n}$. Fig. 4.25 illustrates.

For purely real poles the transient response contains just two real exponents of the form $e^{p_n t}$, thereby having no oscillations. However, it can still contain one "swing" at certain combinations of the amplitudes of the transient partials $e^{p_1 t}$ and $e^{p_2 t}$ (Fig. 4.26).

Strong resonance case

The decay speed of the transient response oscillation (4.36) gets slower as R decreases, which leads to an increased perceived duration of the transient in the output signal. Therefore at high resonance settings a transient in the input signal will produce audible ringing at resonance frequency, even if the steady-state signal doesn't contain it.

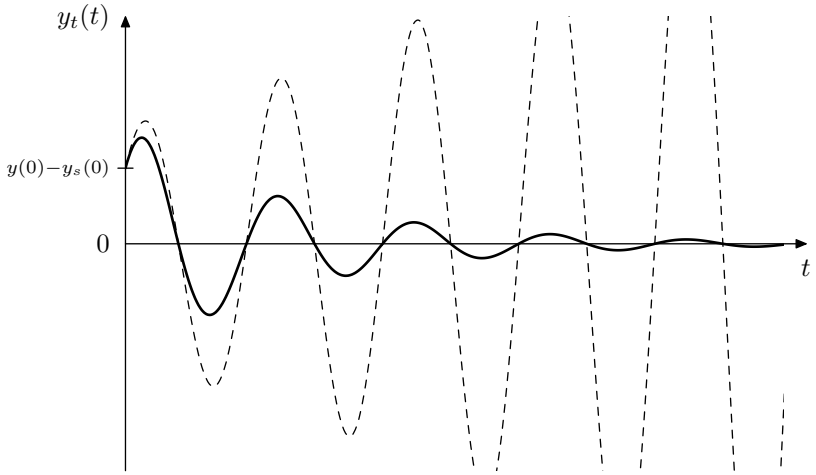


Figure 4.25: Transient response of a resonating 2-pole lowpass filter (dashed line depicts the unstable case).

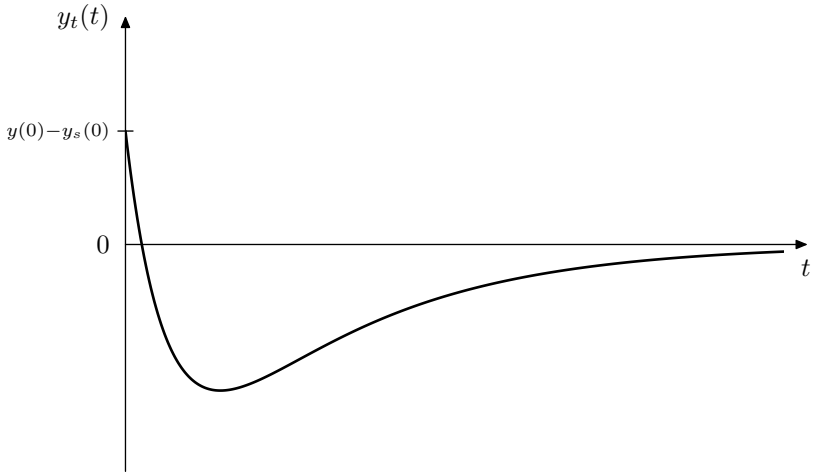


Figure 4.26: Transient response of a non-resonating 1-pole lowpass filter (for the case of a single zero-crossing).

A pretty characteristic and easy to analyse case occurs if we suddenly switch off the filter's input signal. At this moment the steady-state response instantaneously turns to zero and (4.34) turns into

$$\begin{aligned} y_t(t) &= \frac{\dot{y}(0) - p_2 y(0)}{p_1 - p_2} \cdot e^{p_1 t} - \frac{\dot{y}(0) - p_1 y(0)}{p_1 - p_2} \cdot e^{p_2 t} = \\ &= \frac{\dot{y}(0) - p_1^* y(0)}{2j \operatorname{Im} p_1} \cdot e^{p_1 t} - \frac{\dot{y}(0) - p_1 y(0)}{2j \operatorname{Im} p_1} \cdot e^{p_1^* t} = \\ &= \frac{\dot{y}(0) - p_1^* y(0)}{2j \operatorname{Im} p_1} \cdot e^{p_1 t} + \frac{\dot{y}(0) - p_1 y(0)}{2j^* \operatorname{Im} p_1} \cdot e^{p_1^* t} = \end{aligned}$$

$$= 2\,\mathrm{Re}\left(\frac{\dot{y}(0) - p_1^* y(0)}{2j\,\mathrm{Im}\,p_1} e^{p_1 t}\right) = \mathrm{Re}\left(\frac{\dot{y}(0) - p_1^* y(0)}{j\,\mathrm{Im}\,p_1} e^{p_1 t}\right)$$

Therefore

$$y(t) = y_s(t) + y_t(t) = 0 + y_t(t) = \mathrm{Re}\left(\frac{\dot{y}(0) - p_1^* y(0)}{j\,\mathrm{Im}\,p_1} e^{p_1 t}\right)$$

Unless both $y(0) = 0$ and $\dot{y}(0) = 0$, the signal $y(t)$ will have a non-zero amplitude and according to (4.36) we are having a sinusoid of frequency $\omega_c\sqrt{1-R^2}$ decaying as $e^{-R\omega_c t}$.

The opposite situation of a signal being turned on is a kind of a dual case of turning a signal off. Indeed, let $x_0(t)$ be some infinitlely long (that is $t \in (-\infty, \infty)$) steady input signal and let $y_0(t)$ be the respective output signal. Assuming that the filter is stable and that the initial time moment was at $t = -\infty$, by any finite time moment t the transient response component of $y_0(t)$ has decayed to zero, and $y_0(t)$ consists solely of the steady-state response. Let

$$x_1(t) = \begin{cases} x_0(t) & \text{if } t < 0 \\ 0 & \text{if } t \geq 0 \end{cases}$$

be another infinitely long signal decribing the case of the signal $x_0(t)$ being turned off and let $y_1(t)$ be the respective output signal. The signal $x_1(t)$ contains a transient at $t = 0$, thus $y_1(t)$ contains a non-zero transient response component for $t \geq 0$. The case of $x_0(t)$ being turned on is respectively described by

$$x_2(t) = x_0(t) - x_1(t) = \begin{cases} 0 & \text{if } t < 0 \\ x_0(t) & \text{if } t \geq 0 \end{cases}$$

and we let $y_2(t)$ denote the corresponding output signal. Since the system is linear, the output signals are related in the same way as the input signals:

$$y_2(t) = y_0(t) - y_1(t)$$

However $y_0(t)$ doesn't contain any transient response, therefore the only transient response present in $y_2(t)$ is coming from $y_1(t)$, simply having the opposite sign.

The effect of the transient response is particularly remarkable if the input signal is a sinusoid of the same frequency $\omega_c\sqrt{1-R^2}$ as the transient response. First considering the case of turning such sinusoid off we take

$$x_0(t) = a_{\mathrm{in}} \cos(\omega_c\sqrt{1-R^2} \cdot t + \varphi_{\mathrm{in}})$$
$$x_1(t) = \begin{cases} x_0(t) & \text{if } t < 0 \\ 0 & \text{if } t \geq 0 \end{cases}$$

We must have the same sinusoid at the output:

$$y_0(t) = a_{\mathrm{out}} \cos(\omega_c\sqrt{1-R^2} \cdot t + \varphi_{\mathrm{out}})$$
$$y_1(t) = \begin{cases} a_{\mathrm{out}} \cos(\omega_c\sqrt{1-R^2} \cdot t + \varphi_{\mathrm{out}}) & \text{if } t < 0 \\ a_t e^{-R\omega_c t} \cdot \cos(\omega_c\sqrt{1-R^2} \cdot t + \varphi_t) & \text{if } t \geq 0 \end{cases}$$

where the transient response's amplitude and phase a_t and φ_t may differ from the steady-state response's a_{out} and φ_{out} due to the additional factor $e^{-R\omega_c t}$ appearing in the signal. However from the requirement of continuity of $y_1(t)$ and $\dot{y}_1(t)$ at $t = 0$ we may conclude that $a_t \to a_{\text{out}}$ and $\varphi_t \to \varphi_{\text{out}}$ for $R \to 0$.

Now let's consider the case of turning the signal on. We let $x_2(t) = x_0(t) - x_1(t)$. Since we already know that $y_2(t) = 0$ for $t < 0$, we are interested only in $y_2(t)$ for $t \geq 0$ where we have

$$\begin{aligned} y_2(t) &= y_0(t) - y_1(t) = \\ &= a_{\text{out}} \cos(\omega_c \sqrt{1 - R^2} \cdot t + \varphi_{\text{out}}) - a_t e^{-R\omega_c t} \cdot \cos(\omega_c \sqrt{1 - R^2} \cdot t + \varphi_t) \end{aligned}$$

Since at $R \approx 0$ we have $a_t \approx a_{\text{out}}$ and $\varphi_t \approx \varphi_{\text{out}}$, we may in this case rewrite the above as

$$y_2(t) \approx (1 - e^{-R\omega_c t}) \cdot a_{\text{out}} \cos(\omega_c \sqrt{1 - R^2} \cdot t + \varphi_{\text{out}}) \qquad (R \approx 0)$$

That is the sinusoid in the output signal is exponentially fading in as $1 - e^{-R\omega_c t}$. Effectively the transient response is suppressing the steady state signal in the beginning and then slowly lets it fade in (Fig. 4.27).

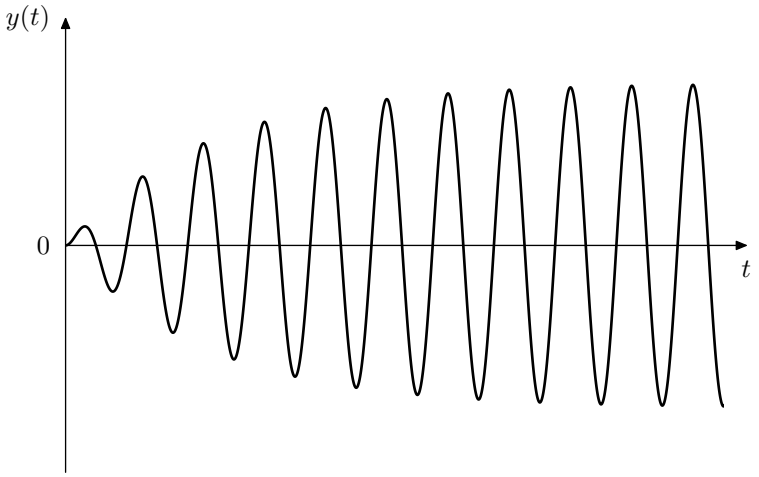


Figure 4.27: Initial suppression of the steady-state signal at $\omega = \omega_c \sqrt{1 - R^2}$ by the transient response.

Selfoscillation

At $R = 0$ the transient response oscillates at a constant amplitude, the frequency of the oscillation being ω_c and coinciding with the infinitely high peak of the amplitude response. Thus, if in the absence of the input signal the system is somehow in a non-zero state, it will stay in this state forever, producing a sinusoid of frequency ω_c. Such state of oscillating without an input signal is referred to as *selfoscillation*.

At $R < 0$ the transient response turns into an infinitely growing signal, while the oscillation frequency becomes lower than ω_c according to (4.36). In nonlinear filters at $-1 < R < 0$ the growing amplitude of the oscillating transient

response will be limited by the saturation, which thereby prevents the filter from exploding. In either case, apparently it is the transient response which is responsible for the selfoscillation of the filter.

We can therefore refer to $-1 < R \leq 0$ as the selfoscillation range of the filter. The boundary $R = 0$ at which the selfoscillation appears may be referred to as *selfoscillation point*.[17]

At the selfoscillation point the poles of the system are located right on the imaginary axis and we can "hit" them with an input sinusoidal signal of frequency ω_c. Since $H(\pm j\omega_c) = \infty$, the steady-state response $H(s)X(s)e^{st}$ becomes infinite too and we need a different choice of the steady-state response signal.

A real sinusoidal signal of frequency ω_c consists of two complex sinusoidal signals of frequencies $\pm\omega_c$. Each of these two signals hits the respective complex pole of the system at $p_{1,2} = \pm j\omega_c$. As we should recall from the discussion in Section 2.15, when a system pole p is hit by an input e^{pt}, the output of the system consists of a linear combination of partials e^{pt} and te^{pt}, where we cannot unambigously select the steady-state response part. From two conjugate poles p_1 and p_2 we'll get a linear combination of $e^{p_1 t}$ and $te^{p_1 t}$ and another one of $e^{p_2 t}$ and $te^{p_2 t}$. After these signals are further combined by (4.32) we'll get a real signal of the form

$$y(t) = a_1 \cdot \cos(\omega_c t + \varphi_1) + a_2 \cdot t\cos(\omega_c t + \varphi_2)$$

Thus, the output signal is a sinusoid of frequency ω_c with the amplitude asymptotically growing as a linear function of time.[18] Clearly, this is a marginal case between the sinusoidal output stabilizing with time if $R > 0$, as e.g. shown in Fig. 4.27, and exponentially exploding if $R < 0$.

Coinciding poles

A special situation occurs if $R = \pm 1$ and thus $p_1 = p_2$. The denominator $p_1 - p_2$ therefore turns to zero, but we can treat this as a limiting case of $R \to \pm 1$. Let $p_{1,2} = p \pm \Delta$ (where $p_{1,2} \to p$ and $\Delta \to 0$). Noticing that

$$G_1(s) \to \frac{-p}{s-p} \qquad G_2(s) \to \frac{-p}{s-p}$$

we can replace $G_n(s)$ in (4.34) with $-p/(s-p)$ before taking the limit:

$$\begin{aligned} y_t(t) &= \frac{\dot{y}(0) - p_2(y(0) - \frac{-p}{s-p}x(0))}{p_1 - p_2} \cdot e^{p_1 t} - \frac{\dot{y}(0) - p_1(y(0) - \frac{-p}{s-p}x(0))}{p_1 - p_2} \cdot e^{p_2 t} = \\ &= \dot{y}(0) \cdot \frac{e^{p_1 t} - e^{p_2 t}}{p_1 - p_2} + \left(y(0) - \frac{-p}{s-p}x(0)\right) \cdot \frac{-p_2 e^{p_1 t} + p_1 e^{p_2 t}}{p_1 - p_2} \end{aligned} \tag{4.37}$$

In the first term of (4.37) we have

$$\frac{e^{p_1 t} - e^{p_2 t}}{p_1 - p_2} = \frac{e^{\Delta t} - e^{-\Delta t}}{2\Delta} \cdot e^{pt} =$$

[17] The other boundary $R = -1$ is hardly ever being reached, therefore we won't introduce a special name for it.

[18] Notice that as the ratio of the amplitudes of the two sinusoids changes, the phase of their sum (which in principle is a sinusoid of the same frequency but of a different amplitude and phase) will slightly drift.

$$= \frac{e^{\Delta t} - e^{-\Delta t}}{2\Delta t} \cdot te^{pt} = \frac{\sinh \Delta t}{\Delta t} \cdot te^{pt} \to te^{pt} \qquad (\Delta \to 0)$$

and in the second term respectively

$$\begin{aligned}
\frac{-p_2 e^{p_1 t} + p_1 e^{p_2 t}}{p_1 - p_2} &= \frac{-(p-\Delta)e^{\Delta t} + (p+\Delta)e^{-\Delta t}}{2\Delta} \cdot e^{pt} = \\
&= -p\frac{e^{\Delta t} - e^{-\Delta t}}{2\Delta} \cdot e^{pt} + \Delta \frac{e^{\Delta t} + e^{-\Delta t}}{2\Delta} \cdot e^{pt} = \\
&= -p\frac{\sinh \Delta t}{\Delta t} \cdot te^{pt} + \cosh \Delta t \cdot e^{pt} \to -pte^{pt} + e^{pt} \qquad (\Delta \to 0)
\end{aligned}$$

and (4.37) at $\Delta = 0$ can be rewritten as

$$\begin{aligned}
y_t(t) &= \dot{y}(0) \cdot te^{pt} + \left(y(0) - \frac{-p}{s-p} x(0) \right) \cdot (-pte^{pt} + e^{pt}) = \\
&= \left(y(0) - \frac{-p}{s-p} x(0) \right) \cdot e^{pt} + \left(\dot{y}(0) - p \cdot \left(y(0) - \frac{-p}{s-p} x(0) \right) \right) \cdot te^{pt}
\end{aligned}$$

Thus, in the case of $p_1 = p_2$ the terms contained in the transient response are having the form e^{pt} or te^{pt}.

The change (4.35) in the inverse Laplace transform in the steady-state response as we take the integral to the left or to the right of the poles of $H(s)$ respectively becomes

$$\begin{aligned}
&\operatorname*{Res}_{s=p+\Delta} H(s)X(s)e^{st} + \operatorname*{Res}_{s=p-\Delta} H(s)X(s)e^{st} \sim \\
&\quad \sim \frac{p^2}{2\Delta} \left(X(p+\Delta)e^{(p+\Delta)t} - X(p-\Delta)e^{(p-\Delta)t} \right) = \\
&\quad = p^2 \frac{X(p+\Delta)e^{(p+\Delta)t} - X(p-\Delta)e^{(p-\Delta)t}}{2\Delta} \to \\
&\quad \to p^2 \frac{X'(p)e^{pt} + X(p)te^{pt} + X'(p)e^{pt} + X(p)te^{pt}}{2} = \\
&\quad = p^2 \left(X'(p)e^{pt} + X(p)te^{pt} \right) \qquad (\Delta \to 0)
\end{aligned}$$

where we have used l'Hôpital's rule.[19] Thus, the change is again solely in the amplitudes of the transient response partials.

It is important to realize that the different form of the transient response components at $R = \pm 1$ doesn't imply that the filter behavior is abruptly switched at this point. The switching of the mathematical expression is solely due to the limitations of the mathematical notation, but doesn't correspond to a jump in any of the signals.

The same result could have been obtained formally by introducing the helper variables u_1 and u_2 differently:[20]

$$u_1 = \dot{y} - py$$

[19] More rigorously speaking, we have used l'Hôpital's rule as a short way to express the following: we expand $X(p \pm \Delta)$ and $e^{(p\pm\Delta)t}$ into Taylor series with respect to Δ, followed by expanding the respective products and cancelling the terms containing Δ with the denominator. One also could expand just $X(p \pm \Delta)$ into Taylor series with respect to Δ and then convert $e^{(p\pm\Delta)t}$ into sinh and cosh in the same way as in the transient response derivation.

[20] This corresponds to using Jordan normal form in the state space representation.

$$u_2 = y$$

(where $p = p_1 = p_2$) thereby obtaining the equations

$$\dot{u}_1 - pu_1 = p^2 x$$
$$\dot{u}_2 - pu_2 = u_1$$

which can be solved using 1-pole techniques. Since u_2 is the input signal for u_1 we have a serial connection of 1-poles, building up a Jordan chain. As we should remember from the discussion of Jordan chains in Section 2.15, the transient response will consist of the partials of the form e^{pt} and te^{pt}. However, due to a completely different substitution of variables, we wouldn't have known, whether the output is changing in a continuous way as R crosses the point $R = 1$. On the other hand, obtaining the result as a limiting case, as we did earlier, gives an answer to that question.

Bandpass and highpass

Notice that (4.25) can be applied separately to steady-state and transient responses (in the sense that the results will still give correct separation of the signal into the steady-state and transient parts). Indeed, e.g. applying (4.25a) to a complex exponential $y_{\mathrm{LP}} = Y(s)e^{st}$ we obtain

$$\dot{y}_{\mathrm{LP}}/\omega_c = sY(s)e^{st}/\omega_c = y_{\mathrm{LP}} \cdot s/\omega_c$$

which matches $H_{\mathrm{BP}}(s) = s/\omega_c \cdot H_{\mathrm{LP}}(s)$. Therefore $\dot{y}_{\mathrm{LP}}/\omega_c$, when applied to a lowpass steady-state response $y_{\mathrm{LPs}}(t)$, will give bandpass steady-state response, etc.

This means that the transient response for the bandpass and highpass signals can be obtained by differentiating the lowpass transient response according to (4.25), resulting in a sum of the same kind of exponential terms $e^{p_1 t}$ and $e^{p_2 t}$ (or e^{pt} and te^{pt} in case $p_1 = p_2$). We won't write the resulting expressions explicitly here.

SUMMARY

The state-variable filter has the structure shown in Fig. 4.1. 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

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

and can be combined to build further filter types.

Chapter 5

Ladder filter

In this chapter we are going to discuss the most classical analog filter model: the transistor ladder filter. The main idea of this structure, which is to create resonance by means of a feedback loop, is encountered in many other filter designs, some of which we are also going to discuss. We will be referring to the class of such filters as simply *ladder filters.*[1]

5.1 Analog model

The most classical example of a ladder filter is transistor ladder filter, which implements a 4-pole lowpass structure shown in Fig. 5.1.[2] The structure in Fig. 5.1 is not limited to transistor-based analog implementations. Particularly, there are many implementations of the same structure based on OTAs (operational transconductance amplifiers). The difference between transistor- and OTA-based ladders is, however, lying in the nonlinear behavior, which we are not touching at this point yet. The linear aspects of both are identical.

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 creates resonance in the filter. Typically $k \geq 0$, although $k < 0$ is also sometimes used.

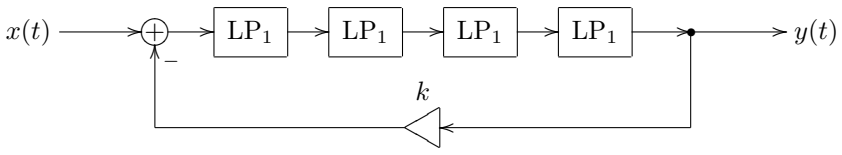


Figure 5.1: Transistor (4-pole lowpass) ladder filter.

[1] Quite unfortunately, there is already another class of filter structures commonly referred to as "ladder filters". Fortunately, this class is not so widely encountered in the synth filter context, on the other hand "transistor ladder" is also a commonly used term. Therefore we'll stick with using the term "ladder filters" for the flters based on a resonating feedback loop.

[2] A widely known piece of work describing this linear model is *Analyzing the Moog VCF with considerations for digital implementation* by T.Stilson and J.Smith.

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

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

and the transfer function of the ladder filter is

$$H(s) = \frac{y}{x} = \frac{H_1^4(s)}{1 + kH_1^4(s)} = \frac{\frac{1}{(1+s)^4}}{1 + k\frac{1}{(1+s)^4}} = \frac{1}{k + (1+s)^4} \tag{5.1}$$

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

The poles of the filter are respectively found from

$$k + (1+s)^4 = 0$$

giving

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

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

$$s = -1 + \frac{\pm 1 \pm j}{\sqrt{2}} k^{1/4} \qquad (k \geq 0) \tag{5.2}$$

(this time $k^{1/4}$ is understood in the real sense). Thus there are 4-poles and we can also refer to this filter as a *4-pole lowpass ladder* filter.

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" (Fig. 5.2). 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. 5.3). At $k = 4$ they hit the imaginary axis:

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

and the filter becomes unstable.[3]

In Fig. 5.3 one could notice that, as the resonance increases, the filter gain at low frequencies begins to drop. Indeed, substituting $s = 0$ into (5.1) we obtain

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

This is a general issue with ladder filter designs.

[3]This time we will not develop an explicit expression for the transient response, since it's getting too involved. Still, the general rule, which we will develop in Section 7.7, is that the transient response is always a linear combination of partials of the form $e^{p_n t}$ (and $t^\nu e^{p_n t}$ in case of repeated poles), where p_n are the filter poles. Respectively, as soon as some of the poles leave the left complex semiplane, the filter becomes unstable.

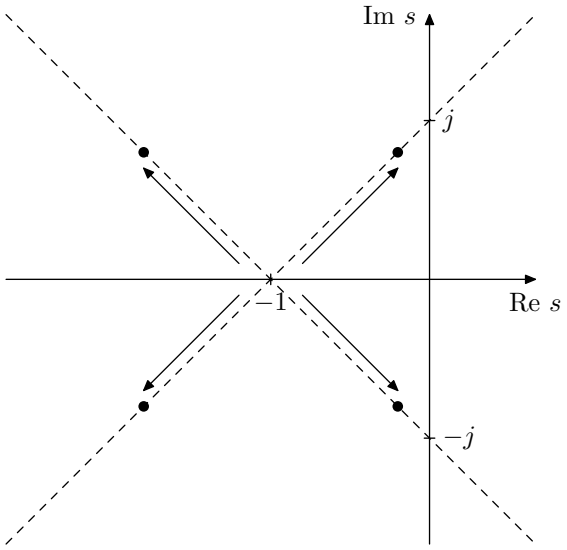


Figure 5.2: Poles of the 4-pole lowpass ladder filter.

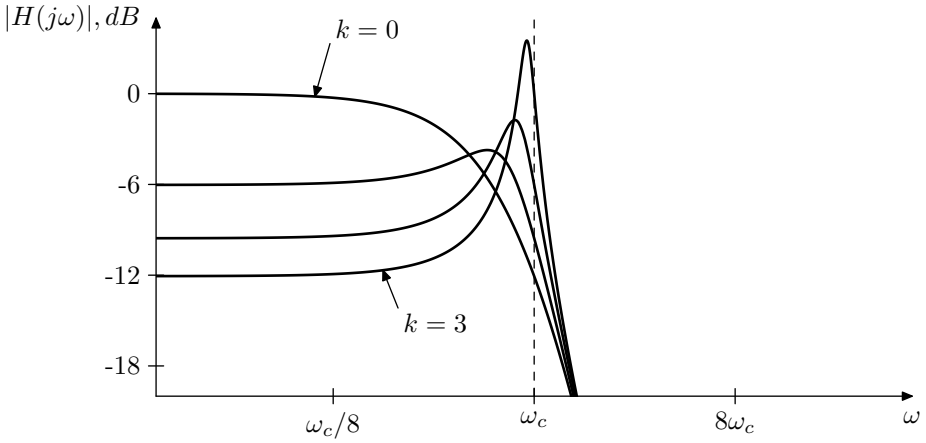


Figure 5.3: Amplitude response of the 4-pole lowpass ladder filter for various k.

5.2 Feedback and resonance

Before we continue with discussing more practical aspects of the ladder filter, we'd like to make one important observation considering the resonance peaks created by the ladder filter feedback.

In Fig. 5.3 we can see that, similarly to the 2-pole case, the resonance frequency is approaching the filter cutoff frequency as the filter approaches selfoscillation at $k = 4$. This is a manifestation of a more general principle concerning ladder filters as such. Consider a general ladder filter in Fig. 5.4, where $G(s)$ denotes a more or less arbitrary structure, whose transfer function is $G(s)$. Notice that the feedback in Fig. 5.4 is not inverted.

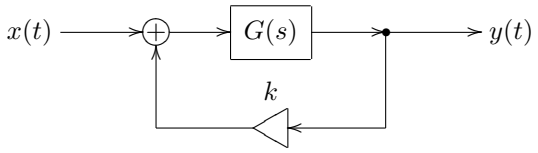


Figure 5.4: Structure of a generic ladder filter.

The transfer function of the entire structure is therefore

$$H(s) = \frac{G(s)}{1 - kG(s)} = \frac{1}{G^{-1}(s) - k} \tag{5.3}$$

and the poles are defined by the equation

$$G^{-1}(s) = k \tag{5.4}$$

That is at $k = 0$ the poles of $H(s)$ are the zeros of $G^{-1}(s)$ (the latter obviously simply being the poles of $G(s)$). As k begins to deviate from zero, the solutions of (5.4) will move in the s-plane, usually in a continuous fashion. E.g. for the 4-pole lowpass ladder (Fig. 5.1) we had $G^{-1}(s) = (s + 1)^4$ and (5.4) takes the form $(s + 1)^4 = -k$, where we take $-k$ instead of k because of the inverted feedback in Fig. 5.1.

The value of k at which the filter starts to selfoscillate should correspond to some of the poles being located on the imaginary axis. At this moment the infinitely high resonance peak in the amplitude response is occurring exactly at these pole positions. Denoting a purely imaginary pole position as $j\omega$, we rewrite (5.4) for such poles as

$$G^{-1}(j\omega) = k$$

or

$$kG(j\omega) = 1 \tag{5.5}$$

We can refer to (5.5) as the *selfoscillation equation for a feedback loop.* This equation implies that selfoscillation appears at the moment when the total frequency response across the feedback loop $kG(j\omega)$ exactly equals 1 at some frequency ω. That is the total amplitude gain must be 1, and the total phase shift must be 0°.

This is actually a pretty remarkable result. Of course it is quite intuitive that selfoscillation tends to occur at frequencies where the feedback signal doesn't cancel the input signal, but rather boosts it. And such boosting tends to be strongest at frequencies where we have a 0° total phase shift across the feedback loop. However, what is quite counterintuitive, is that selfoscillation can appear (as k is reaching the respective threshold value) *only* at such frequencies.[4]

Therefore for $k > 0$ the selfoscillation appears at frequencies where the phase response of $G(s)$ is 0°. For $k < 0$ the selfoscillation appears at frequencies where

[4] As k continues to grow into the unstable range, the frequencies of the exploding (or still selfoscillating, if the filter is nonlinear) sinusoidal transient response partials can change, since the imaginary part of the resonating poles can change as the poles move beyond the imaginary axis.

the phase response of $G(s)$ is 180°. The respective value of k can be found from (5.5) giving

$$k = \frac{1}{G(j\omega)} \tag{5.6}$$

or, rewriting (5.6) in terms of the amplitude response of $G(s)$:

$$k = \pm\frac{1}{|G(j\omega)|} \tag{5.7}$$

where we take the plus sign if the phase response of $G(s)$ at ω is 0° and the minus sign if the phase response of $G(s)$ at ω is 180°.

The just discussed effects are the reason that we used negative feedback in the 4-pole lowpass ladder filter. We want the resonance to occur at the filter's cutoff. The phase response of a single 1-pole lowpass at the cutoff frequency is −45°, respectively the phase response of a chain of four 1-poles is −180°, exactly what we need for the resonance peak, if we use negative feedback.

At the same time, the amplitude response of a 1-pole lowpass at the cutoff is $|1/(1+j)| = 1/\sqrt{2}$, respectively the amplitude response of a chain of four 1-poles is $(1/\sqrt{2})^4 = 1/4$. According to (5.7), the infinite resonance is attained at $k = 1/(1/4) = 4$.

At $\omega = 0$ a chain of four 1-pole lowpasses will have a phase shift of 0°, while the amplitude response at $\omega = 0$ is 1. Therefore, in Fig. 5.1 the "selfoscillation" at $\omega = 0$ will occur at $k = -1$. However the amplitude response peak at $\omega = 0$ hardly can count as resonance.

5.3 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 (3.29), 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:

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

and the fourth one:

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

where

$$G = g^4$$

$$S = g^3 s_1 + g^2 s_2 + g s_3 + s_4$$

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. 5.5.

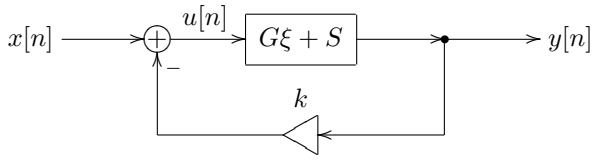


Figure 5.5: TPT 4-pole ladder filter in the instantaneous response form.

Rather than solving for y, let's solve for the signal u at the feedback point. From Fig. 5.5 we obtain

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

from where

$$u = \frac{x - kS}{1 + kG} \tag{5.8}$$

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.

Apparently the total instantaneous gain of the zero-delay feedback loop in Fig. 5.5 and in (5.8) is $-kG$. As we should recall from the discussion of 1-pole lowpass filters, $0 < g < 1$ for positive cutoff settings. Respectively $0 < G < 1$ and the filter doesn't become instantaneously unstable provided $k \geq -1$.

5.4 Feedback shaping

We have observed that at high resonance settings the amplitude gain of the filter at low frequencies drops (Fig. 5.3). An obvious way to fix this problem would be e.g. to boost the input signal by the $(1+k)$ factor.[5] However there's another way to address the same issue. We could "kill" the feedback for the low frequencies only by introducing a highpass filter into the feedback path (Fig. 5.6). In the simplest case this could be a 1-pole highpass.

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. 5.7).

In principle one can also use other filter types in the feedback shaping. One has to be careful though, since this changes the total phase and amplitude responses of the feedback path, thus the frequency of the resonance peak and the

[5] 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.

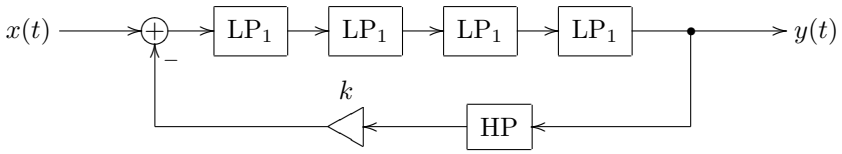


Figure 5.6: Transistor ladder filter with a highpass in the feedback.

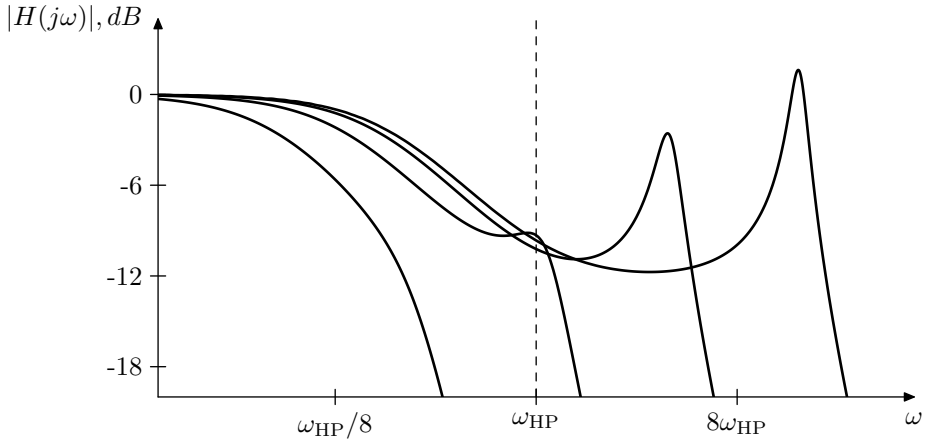


Figure 5.7: Amplitude response of the ladder filter with a static-cutoff highpass in the feedback for various lowpass cutoffs.

value of k at which selfoscillation is reached may be changed. E.g., quite counterintuitively, inserting a 1-pole lowpass into the feedback path can destabilize an otherwise stable filter.

In order to establish and analyse the latter fact mathematically, we'd need to find the total amplitude response across the feedback loop at the point where the total phase shift is 180°. Let $H_1(s) = 1/(1+s)$ be the underlying 1-pole lowpass of the ladder filter and let $H_f(s) = 1/(1+s/\omega_{cf})$ be the lowpass in the feedback, with a generally speaking different cutoff ω_{cf}. The 180° point is found from the equation

$$\begin{aligned} 4\arg H_1(j\omega) + \arg H_f(j\omega) &= 4\arg\frac{1}{1+j\omega} + \arg\frac{1}{1+j\omega/\omega_{cf}} = \\ &= -4\arctan\omega - \arctan\frac{\omega}{\omega_{cf}} = -\pi \end{aligned} \tag{5.9}$$

where we have used (2.8). The equation (5.9) looks a bit daunting, if having an analytic solution at all. Fortunately, we don't actually need to know the frequency of the 180° point, it would suffice to know the respective amplitude responses.

Let $\varphi_1(\omega)$ be the negated phase response of $H_1(s)$:

$$\varphi_1(\omega) = -\arg H_1(j\omega) = \arctan\omega > 0 \qquad \forall\omega$$

Expressing ω as a function of φ_1 we have $\omega = \tan\varphi_1$. Respectively, expressing

the amplitude response as a function of the (negated) phase response we have

$$A_1 = |H_1(j\omega)| = \frac{1}{\sqrt{1+\omega^2}} = \frac{1}{\sqrt{1+\tan^2\varphi_1^2}} = \cos\varphi_1 \tag{5.10}$$

Thus, the total amplitude response of the four 1-poles in the feedforward path of the ladder filter is

$$A_1^4(\omega) = \frac{1}{(1+\varphi_1^2)^2}$$

and the total phase response of the feedforward path is $4\varphi_1$.

Since (5.10) is cutoff-independent, it also holds for $H_f(s)$:

$$A_f = \cos\varphi_f$$

where $A_f = |H_f(j\omega)|$, $\varphi_f = -\arg H_f(j\omega)$. Now let ω_0 be the (unknown to us) solution of (5.9), that is the total phase shift at ω_0 is 180°. In terms of the just introduced functions $\varphi_1(\omega)$ and $\varphi_f(\omega)$ equation (5.9) can be rewritten as

$$4\varphi_1(\omega_0) + \varphi_f(\omega_0) = \pi \tag{5.11}$$

Since $\varphi_f(\omega) > 0\ \forall\omega$, the 180° phase shift is achieved earlier than without the feedback filter, that is $\omega_0 < 1$ (whatever the value of ω_{cf} is).

Computing the total amplitude response of all five 1-pole lowpasses at ω_0 we have

$$A_1^4(\omega_0)\cdot A_f(\omega_0) = \cos^4\varphi_1(\omega_0)\cdot\cos\varphi_f(\omega_0) = \cos^4\left(\frac{\pi}{4} - \frac{\varphi_f(\omega_0)}{4}\right)\cdot\cos\varphi_f(\omega_0)$$

Considering only the first factor we have

$$\cos^4\left(\frac{\pi}{4} - \frac{\varphi_f}{4}\right) = \left(\frac{1+\cos\left(\frac{\pi}{2} - \frac{\varphi_f}{2}\right)}{2}\right)^2 = \left(\frac{1+\sin\frac{\varphi_f}{2}}{2}\right)^2$$

(where we dropped the argument ω_0, understanding it implicity). Respectively

$$A_1^4\cdot A_f = \frac{1}{4}\cdot\left(1+\sin\frac{\varphi_f}{2}\right)^2\cdot\cos\varphi_f \qquad (\omega=\omega_0) \tag{5.12}$$

Fig. 5.8 contains the graph of (5.12). The interpretation of this graph is like follows. Suppose the feedback lowpass's cutoff ω_{cf} is very large ($\omega_{cf}\to+\infty$). In the limit the feedback lowpass has no effct and

$$\omega_0 = 1 \quad \varphi_f(\omega_0) = 0 \quad A_f(\omega_0) = 1 \quad A_1^4(\omega_0)A_f(\omega_0) = \frac{1}{4} \qquad (\text{for } \omega_{cf} = +\infty)$$

As we begin to lower ω_{cf} back from the infinity, the value of $\varphi_f(\omega_0)$ grows from zero into the positive value range. The graph in Fig. 5.8 plots the total amplitude response of the five 1-pole lowpasses in the feedback loop against the growing $\varphi_f(\omega_0)$. We see that the amplitude response grows for quite a while. As long as it is above 1/4, the filter will explode at $k = 4$. The zero amplitude response at $\varphi_f = \pi/2$ corresponds to $\omega_{cf} = 0$, where the extra lowpass is fully closed, thus the entire feedback loop is muted.

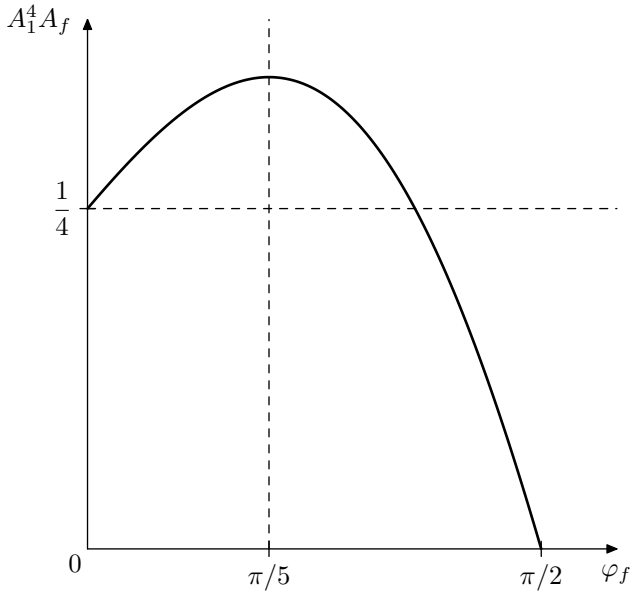


Figure 5.8: Total amplitude response of the four feedforward lowpass 1-poles plus the feedback lowpass 1-pole at the 180° phase shift point, plotted against the phase shift by the feedback 1-pole.

At $\omega_{cf} = 1$ (equal cutoffs of all 1-poles) from (5.11) we have $\varphi_f(\omega_0) = \varphi_1(\omega_0) = \pi/5$. In Fig. 5.8 one can see that this is the "most unstable" situation among all possible ω_{cf}.

In comparison, if we had a 1-pole highpass in the feedback, then we would have $\arg H_f(j\omega) > 0$ and respectively $\varphi_f(\omega) < 0\ \forall\omega$. Therefore the 180° point would be shifted to the right: $\omega_0 > 1$. Therefore $A_1^4(\omega_0) < A_1^4(1) < 1/4$, while $A_f(\omega) < 1\ \forall\omega$, thus the total amplitude response $A_1^4 A_f$ at the 180° point would decrease and the filter won't become "more unstable" than it was before the introduction of the extra highpass filter.

5.5 Multimode ladder filter

Warning! *The multimode functionality of the ladder filter is a somewhat special feature. There are more straightforward ways to build bandpass and highpass ladders, discussed later in this chapter.*

By picking up intermediate signals of the ladder filter as in Fig. 5.9 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.[6]

Suppose $k = 0$. Apparently, in this case, the respective transfer functions

[6] 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.

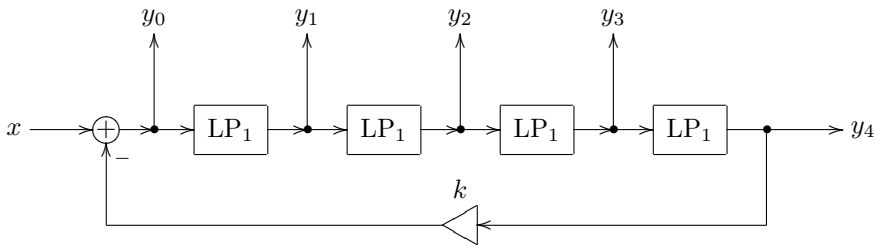


Figure 5.9: Multimode ladder filter.

associated with each of the y_n outputs are

$$H_n(s) = \frac{1}{(1+s)^n} \qquad (n = 0, \ldots, 4) \tag{5.13}$$

If $k \neq 0$ then from

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

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

$$H_n(s) = \frac{(1+s)^{4-n}}{k + (1+s)^4} \tag{5.14}$$

4-pole highpass mode

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 $1/(1+s)$

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

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

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

Let's attempt to build $H_{\mathrm{HP}}(s)$ 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

$$H_{\mathrm{HP}}(s) = \frac{s^4}{k + (1+s)^4} \tag{5.15}$$

which at $k = 0$ will turn into $s^4/(1+s)^4$. We also have $H_{\mathrm{HP}}(\infty) = 1$ while the four zeros at $s = 0$ provide a 24dB/oct rolloff at $\omega \to 0$, thus we are still having a more or less reasonable highpass. In order to express $H_{\mathrm{HP}}(s)$ as a sum of the modes we write

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

that is

$$s^4 = a_0(1+s)^4 + a_1(1+s)^3 + a_2(1+s)^2 + a_3(1+s) + a_4$$

We need to find a_n from the above equation, which generally can be done by equating the coefficients at equal powers of s in the left- and right-hand sides. However, for the specific equation that we're having here we could do a shortcut by simply formally replacing $s+1$ by s (and respectively s by $s-1$):

$$(s-1)^4 = a_0 s^4 + a_1 s^3 + a_2 s^2 + a_3 s + a_4$$

from where immediately

$$a_0 = 1,\ a_1 = -4,\ a_2 = 6,\ a_3 = -4,\ a_4 = 1$$

The amplitude response corresponding to (5.15) is plotted in Fig. 5.10.

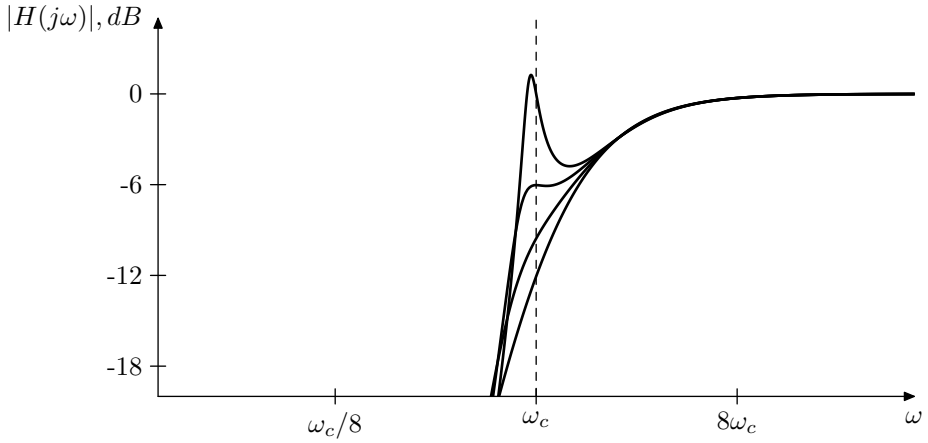


Figure 5.10: Amplitude response of the highpass mode of the ladder filter for various k.

4-pole bandpass mode

A bandpass filter can be built as

$$H_{\mathrm{BP}}(s) = \frac{s^2}{k + (1+s)^4} \tag{5.16}$$

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:

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

The coefficients are found from

$$\begin{aligned} s^2 &= a_0(1+s)^4 + a_1(1+s)^3 + a_2(1+s)^2 + a_3(1+s) + a_4 \\ (s-1)^2 &= a_0 s^4 + a_1 s^3 + a_2 s^2 + a_3 s + a_4 \end{aligned}$$

The amplitude response corresponding to (5.16) is plotted in Fig. 5.11.

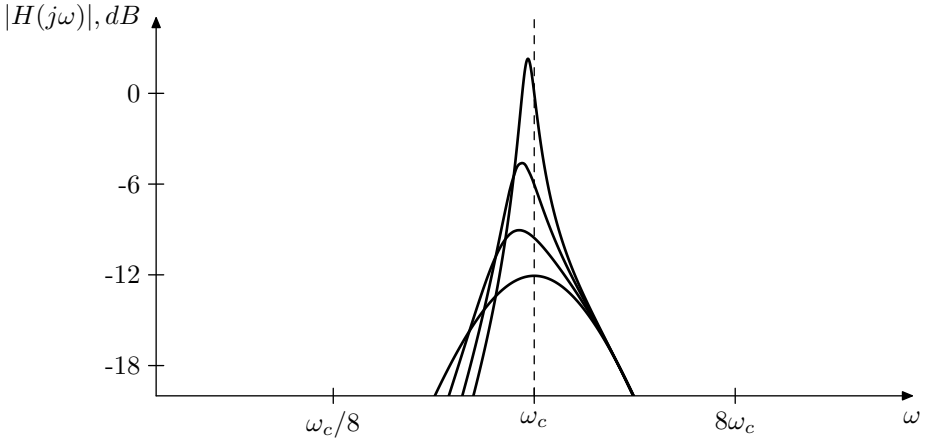


Figure 5.11: Amplitude response of the bandpass mode of the ladder filter for various k.

Lower-order modes

Recalling the transfer functions of the modal outputs y_n in the absence of the resonance (5.13), we can consider the modal signals y_n and their respective transfer functions (5.14) as a kind of "n-pole lowpass filters with 4-pole resonance".

"Lower-order" highpasses can be build by considering the zero-resonance transfer functions

$$H_{\mathrm{HP}}(s) = \frac{s^N}{(s+1)^N} = \frac{(s+1)^{4-N}s^N}{(s+1)^4}$$

which for $k \neq 0$ turn into

$$H_{\mathrm{HP}}(s) = \frac{(s+1)^{4-N}s^N}{k+(s+1)^4}$$

In a similar way we can build a "2-pole" bandpass

$$H_{\mathrm{BP}}(s) = \frac{s}{(s+1)^2} = \frac{(s+1)^2 s}{(s+1)^4} \qquad (k=0)$$

$$H_{\mathrm{BP}}(s) = \frac{(s+1)^2 s}{k+(s+1)^4} \qquad (k \neq 0)$$

Other modes

Continuing in the same fashion we can build further modes (the transfer functions are given for $k = 0$):

$$\frac{s}{(s+1)^3}$$ 3-pole bandpass, 6/12 dB/oct

$$\frac{s^2}{(s+1)^3}$$ 3-pole bandpass, 12/6 dB/oct

$\frac{(s+1)^4 + Ks^2}{(s+1)^4}$	band-shelving
$\frac{s^4 - 1}{(s+1)^4}$	notch
$\frac{(s^2+1)^2}{(s+1)^4}$	notch
$\frac{(s^2+2Rs+1)^2 + (s^2-2Rs+1)^2}{2(s+1)^4}$	2 notches, neutral setting $R = 1$
$\frac{s^2+1}{(s+1)^4}$	2-pole lowpass + notch
$\frac{(1+1/s^2)s^4}{(s+1)^4}$	2-pole highpass + notch
$\frac{(s+1/s)s^2}{(s+1)^4}$	2-pole bandpass + notch

etc. The principles are more or less similar. We are trying to attain a desired asymptotic behavior at $\omega \to 0$ and $\omega \to +\infty$ by having the necessary orders and coefficients of the lowest-order and highest-order terms in the numerator. E.g. by having s^2 as the lowest-order term of the numerator we ensure a 12dB/oct rolloff at $\omega \to 0$, or by having s^4 as the highest-order term we ensure $H(\infty) = 1$. The notch at $\omega = 1$ is generated by placing a zero at $s = \pm j$. The 2-notch version is obtained by explicitly writing out the transfer function of a 4-pole multinotch described in Section 11.3.

5.6 HP ladder

Performing an LP to HP transformation on the lowpass ladder filter we effectively perform it on each of the underlying 1-pole lowpasses, thus turning them into 1-pole highpasses. Thereby we obtain a "true" highpass ladder filter (Fig. 5.12). Obviously, the amplitude response of the ladder highpass is symmetric to the amplitude response of the ladder lowpass (Fig. 5.13).

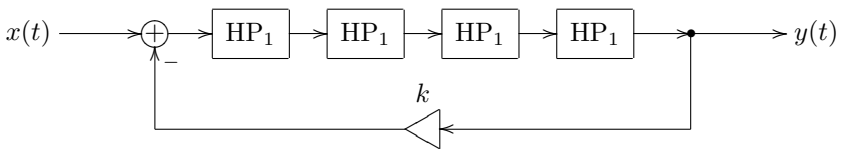


Figure 5.12: A "true" highpass ladder filter.

The instantaneous gain of a 1-pole highpass is complementary to the instantaneous gain of the 1-pole lowpass:

$$1 - \frac{g}{1+g} = \frac{1}{1+g}$$

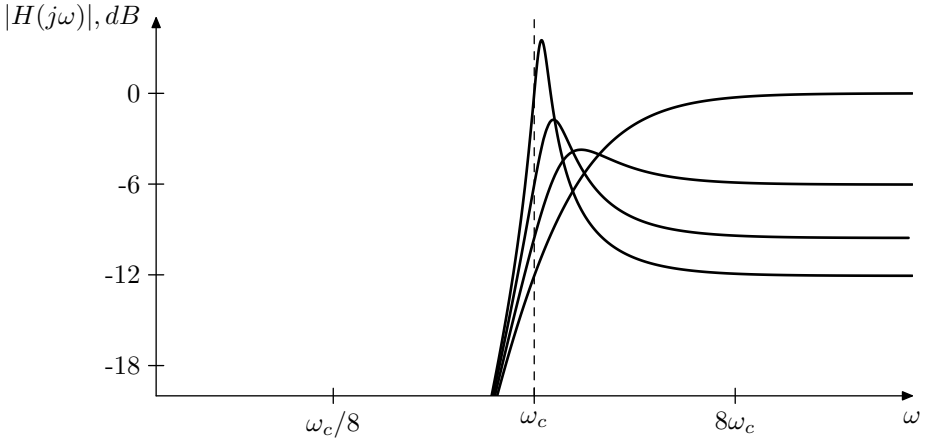


Figure 5.13: Amplitude response of the 4-pole highpass ladder filter for various k.

where $g = \omega_c T/2$. Thus the instantaneous gain of a single 1-pole highpass is varying within the range (0,1) and so does the gain of the chain of four highpasses: $0 < G < 1$. Therefore, 4-pole highpass ladder doesn't get instantaneously unstable for $k > -1$.

5.7 BP ladder

In order to build a "true" 4-pole bandpass ladder, we replace only half of the lowpasses with highpasses (it doesn't matter which two of the four 1-pole lowpasses are replaced). The total transfer function of the feedforward path is thereby

$$\frac{s^2}{(1+s)^4} = \frac{s}{(1+s)^2} \cdot \frac{s}{(1+s)^2}$$

where each of the $s/(1+s)^2$ factors is built from a serial combination of a 1-pole lowpass and a 1-pole highpass:

$$\frac{s}{(1+s)^2} = \frac{s}{1+s} \cdot \frac{1}{1+s}$$

Apparently $s/(1+s)^2 = s/(1+2s+s^2)$ is a 2-pole bandpass with damping $R = 1$ and a serial combination of two of them makes a 4-pole bandpass. The frequency response of $s/(1+s)^2$ at $\omega = 1$ is $1/2$, that is there is no phase-shift. Respectively the frequency response of $s^2/(1+s)^4$ at $\omega = 1$ is $1/4$, also without a phase shift. Therefore we need to use positive rather than negative feedback (Fig. 5.14), the selfoscillation still occuring at $k = 4$, the same as with lowpass and highpass ladders.

Noticing that the filter structure is invariant relative to the LP to HP transformation, we conclude that its amplitude response must be symmetric (around $\omega = 1$) in the logarithmic frequency scale (Fig. 5.15).

The question of instantaneous instability is more critical for the bandpass ladder, since the feedback is positive. The instantaneous gain of a lowpass-highpass pair is a product of the instantaneous gains of a 1-pole lowpass and a

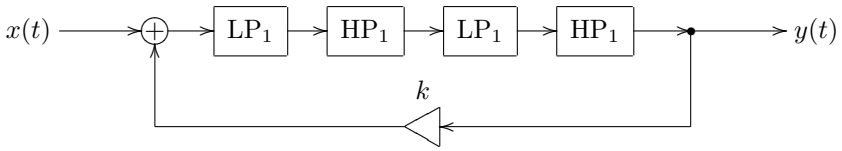


Figure 5.14: A "true" bandpass ladder filter.

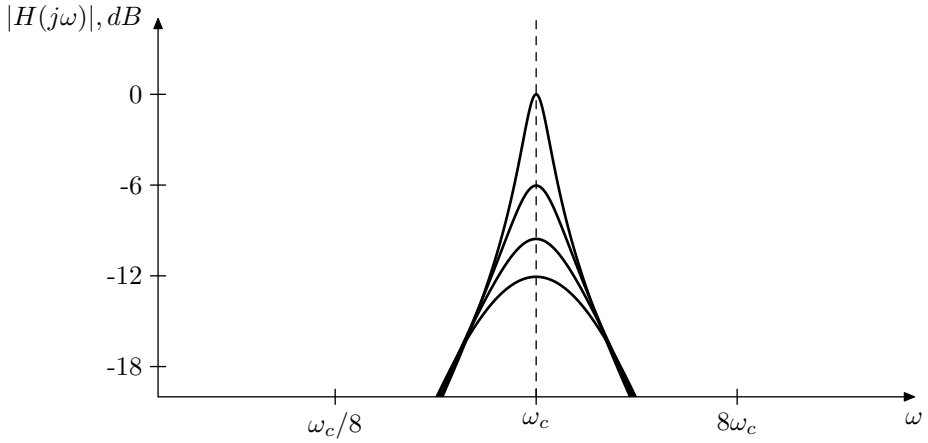


Figure 5.15: Amplitude response of the 4-pole bandpass ladder filter for various k.

1-pole highpass:

$$\frac{g}{1+g}\cdot\frac{1}{1+g}$$

(where $g = \omega_c T/2$). It's not difficult to verify that the maximum gain of this pair is attained at $g = 1$ and is equal to $1/4$. The maximum instantaneous gain of two of these pairs is therefore $1/16$, and thus the instantaneously unstable case doesn't occur provided $k < 16$.

Bandwidth control

Using (5.3) and the fact that the frequency response of $s^2/(1+s)^4$ at $\omega = 1$ is $1/4$ we obtain the frequency response of the 4-pole bandpass ladder at the cutoff

$$H(j) = \frac{1}{4-k}$$

Therefore, by multiplying the output (or the input signal) of the 4-pole bandpass ladder by $4-k$ we can turn it into a normalized bandpass, where the bandwidth is controlled by varying k.

There is another way, however. Recall that the normalized 2-pole bandpass (4.15) is an LP to BP transformation of the 1-pole lowpass $1/(1+s)$. At the same time,

$$\frac{1}{1+s}\cdot\frac{s}{1+s} = \frac{s}{(1+s)^2} = \frac{s}{1+2s+s^2} = \frac{1}{2}\cdot\frac{2s}{1+2s+s^2}$$

is simply a halved version of (4.15) taken at $R = 1$ and therefore is an LP to BP transformation fo the halved 1-pole lowpass $1/2(1+s)$. This means that Fig. 5.14 can be replaced by Fig. 5.16 which in turn is an LP to BP transformation of Fig. 5.17.

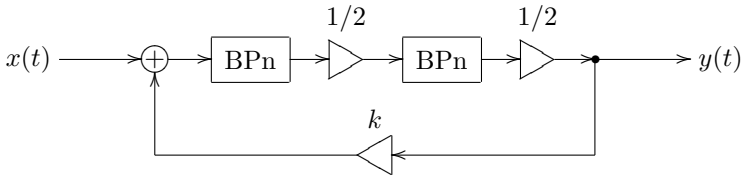


Figure 5.16: 4-pole bandpass ladder filter expressed in terms of normalized 2-pole bandpasses.

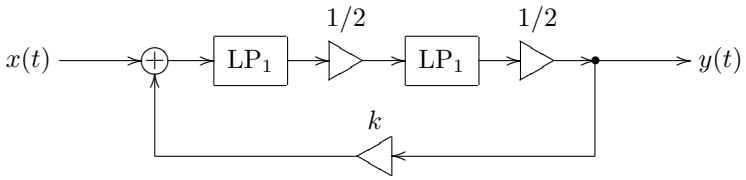


Figure 5.17: LP to BP transformation applied to this structure produces the 4-pole bandpass ladder in Fig. 5.16.

We don't even specifically care to analyse the structure Fig. 5.17. What is important is that the damping parameter of the LP to BP transformation controls the transformation bandwidth and thereby the bandwidth of the bandpass ladder in Fig. 5.16. Thus, introducing the damping control into the normalized 2-pole bandpasses in Fig. 5.16 we can control the bandpass ladder's bandwidth by simply varying the damping parameter of the underlying 2-pole bandpasses.

At the same time we still have the k parameter available, which we still can use to control the bandwidth of the normalized bandpass (Fig. 5.18). Thus, k and R provide two different ways of bandwidth control, resulting in somewhat different amplitude response shapes (Fig. 5.19).[7]

Obviously, normalized 2-pole bandpasses with damping control could be implemented using an SVF. If nonlinearities are involved, however, using TSK/SKF 2-pole bandpasses might be a better option. Since we didn't introduce the latter yet, we need to postpone the respective discussion. We will return to this question, however, in the discussion of 8-pole bandpass ladder in Section 5.9, where the bandwidth control via the 2-pole bandpass damping will be a particularly desired feature compared to being somewhat academic in the case of a 4-pole bandpass.

[7]In principle, k and R have very similar effects. Fundamentally, they both affect the bandwidth and the resonance peak height. In Fig. 5.18 their effect on the resonance peak height is compensated, the compensation for k being the $4 - k$ gain at the output, the compensation for R being embedded into the normalized bandpasses. By removing the normalization from the bandpasses we effectively introduce the $1/R^2$ gain into the feedback, and the damping R thereby will control the resonance peak height too.

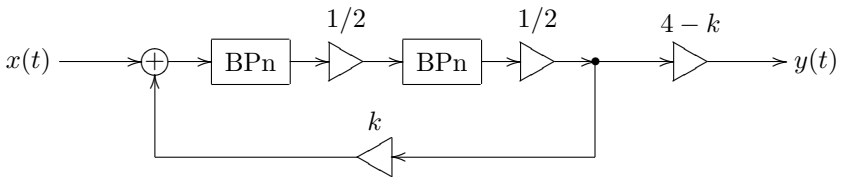


Figure 5.18: 4-pole normalized bandpass ladder filter expressed in terms of normalized 2-pole bandpasses.

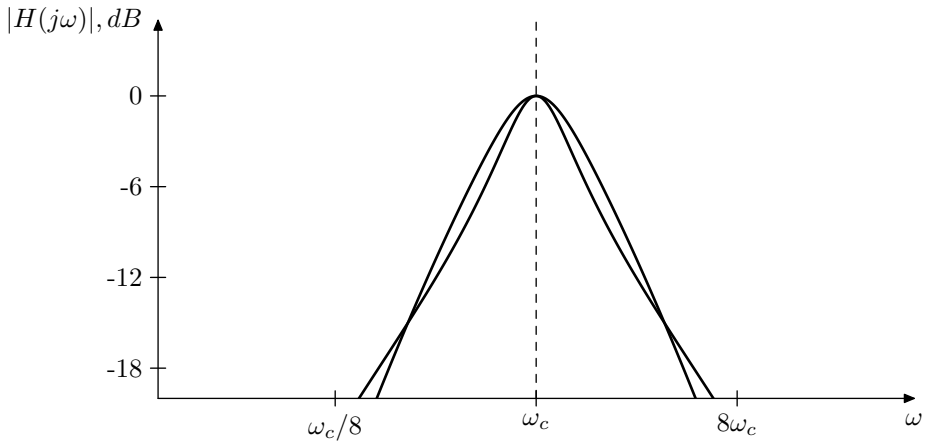


Figure 5.19: Amplitude response of the 4-pole normalized bandpass ladder filter in Fig. 5.18 for two different combinations of k and R resulting in comparable bandwidths.

5.8 Sallen–Key filters

In this section we are going to introduce two special kinds of 2-pole bandpass ladder filters, the Sallen–Key filter and its transpose.[8] They are important because of their nonlinear versions, since, as linear digital 2-pole filters go, the SVF filter could be sufficient for most applications, and it also provides probably the best performance among different TPT 2-poles.

For now we shall develop the linear versions of these filters. The Sallen–Key filter is more famous than its transpose, but we'll start with the transpose, for the sake of a more systematic presentation of the material.

Transposed Sallen–Key (TSK) filters

Attempting to build a 2-pole lowpass ladder filter (Fig. 5.20) we don't end up with a useful filter.

[8] Despite essentially being bandpass ladder filters, the Sallen–Key filter and its transpose can be (and are) used to deliver lowpass and highpass responses as well.

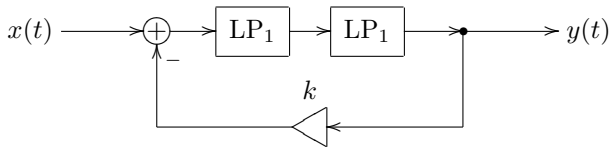


Figure 5.20: 2-pole lowpass ladder filter (not very useful).

Indeed, the transfer function of this filter is

$$H(s) = \frac{1}{k + (1+s)^2}$$

and the poles are respectively at

$$s = -1 \pm \sqrt{-k} = -1 \pm j\sqrt{k} \qquad (k \geq 0)$$

Interpreting these pole positions in terms of 2-pole cutoff and damping (which we can do using (4.13)), we obtain

$$\begin{cases} \omega_c = \left|-1 \pm j\sqrt{k}\right| = \sqrt{1+k} \\ R = \dfrac{-\operatorname{Re}\left(-1 \pm j\sqrt{k}\right)}{\left|-1 \pm j\sqrt{k}\right|} = \dfrac{1}{\sqrt{1+k}} \end{cases}$$

Thus, firstly, there is coupling between the feedback amount and the effective cutoff of the filter. Secondly, as k grows, R stays strictly positive, thus the filter poles never go into the right semiplane (and, as with the 4-pole ladder filter, this would be quite desired once we make the filter nonlinear). So, all in all, not a very useful structure.

A similar situation occurs in an attempt to use two 1-pole highpasses instead of two 1-pole lowpass in the same structure (the readers may wish verify this on their own as an exercise).

This result is no wonder, considering that the transfer function of a chain of two 1-pole lowpasses is $1/(1+s)^2$, with the phase response being $0°$ only at $\omega = 0$ and being $180°$ only at $\omega = \infty$ (for the highpasses the situation is opposite, we have $180°$ only at $\omega = 0$ and $0°$ only at $\omega = \infty$, which doesn't make a big difference for our purposes). Thus we don't get a good resonance peak at any finite location. This however hints at the idea that we might still try to build a 2-pole bandpass ladder filter from a chain of a 1-pole lowpass and a 1-pole highpass, as the total phase shift at the cutoff would be $0°$ in this case:

$$\left(\frac{1}{1+s} \cdot \frac{s}{1+s}\right)\Bigg|_{s=j} = \frac{s}{(1+s)^2}\Bigg|_{s=j} = \frac{j}{(1+j)^2} = \frac{1}{2}$$

The respective structure is shown in Fig. 5.21. Notice that we don't invert the feedback.

Computing the transfer function of this filter we have

$$H(s) = \frac{\dfrac{s}{(1+s)^2}}{1 - k\dfrac{s}{(1+s)^2}} = \frac{s}{(1+s)^2 - ks} = \frac{s}{s^2 + (2-k)s + 1}$$

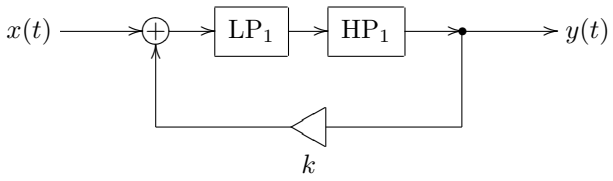


Figure 5.21: 2-pole bandpass ladder filter.

The obtained expression is identical to the transfer function of a 2-pole bandpass filter with a damping gain $2R = 2 - k$. That is, the filter in Fig. 5.21 is pretty much the same as a linear 2-pole SVF bandpass, at least from the frequency response perspective. Notice that $k = 0$ corresponds to the resonance-neutral setting ($R = 1$) while $k = 2$ is the self-oscillation point ($R = 0$). As we should remember from the 4-pole bandpass ladder discussion, the maximum possible instantaneous gain of the lowpass-highpass pair is $1/4$, therefore under the condition $k < 4$ the TPT implementation of Fig. 5.21 doesn't become instantaneously unstable.

It might seem that we have failed to construct a 2-pole lowpass filter using the above approach, but in fact with a slight modification we can obtain one from the bandpass filter in Fig. 5.21. Let's replace the 1-pole highpass with a 1-pole multimode with highpass and lowpass outputs (Fig. 5.22).

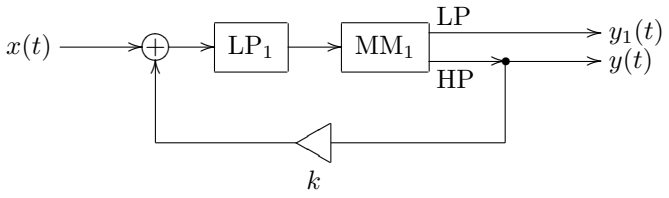


Figure 5.22: 2-pole bandpass ladder filter with an extra output mode.

Obviously, the signal $y(t)$ is not affected by this replacement. Let's find out what kind of signal is $y_1(t)$. In order to simplify the computation of the transfer function of the entire structure at y_1, consider first the transfer functions of the 1-pole multimode filter used in isolation:

$$H_{\mathrm{LP}}(s) = \frac{1}{1+s} \qquad H_{\mathrm{HP}}(s) = \frac{s}{1+s}$$

or, for complex sinusoidal signals of the form e^{st}

$$Y_{\mathrm{LP}}(s) = \frac{1}{1+s}X(s) \qquad Y_{\mathrm{HP}}(s) = \frac{s}{1+s}X(s)$$

where $X(s)e^{st}$ is the input signal of the multimode 1-pole and $Y_{\mathrm{LP}}(s)e^{st}$ and $Y_{\mathrm{HP}}(s)e^{st}$ are the respective output signals. This means that

$$Y_{\mathrm{LP}}(s) = \frac{Y_{\mathrm{HP}}(s)}{s}$$

Therefore a similar relationship exists between the outputs $y_1(t)$ and $y(t)$ of the filter in Fig. 5.22:

$$Y_1(s) = \frac{Y(s)}{s}$$

and there is the same relationship between their respective transfer functions

$$H_1(s) = \frac{H(s)}{s} = \frac{1}{s} \cdot \frac{s}{s^2 + (2-k)s + 1} = \frac{1}{s^2 + (2-k)s + 1}$$

where $H_1(s)$ the the transfer function for the signal $y_1(t)$ in respect to the input signal $x(t)$. Therefore $y_1(t)$ is an ordinary 2-pole lowpass signal with damping gain $2R = 2 - k$.

Thus we have obtained a multimode 2-pole ladder filter with the lowpass and bandpass outputs. We redraw the structure in Fig. 5.22 once again as Fig. 5.23 to reflect what we have just found out about this structure.

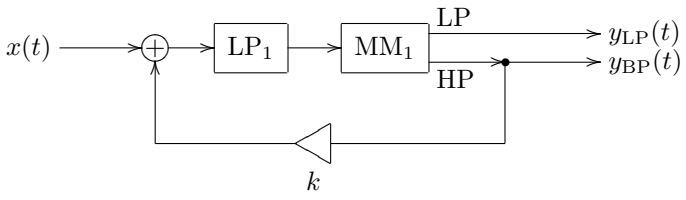


Figure 5.23: Transposed Sallen–Key (TSK) filter.

The structure in Fig. 5.23 happens to be a transpose of the Sallen–Key filter, therefore we will refer to it as the *transposed Sallen–Key* (TSK) filter.[9] The transfer functions of the TSK filter are, as we have found out:

$$H_{\mathrm{LP}}(s) = \frac{1}{s^2 + (2-k)s + 1}$$

$$H_{\mathrm{BP}}(s) = \frac{s}{s^2 + (2-k)s + 1}$$

A 2-pole highpass output mode cannot be picked up in a straightforward way, but can be obtained with some extra effort. Let's also turn the first lowpass into a multimode (Fig. 5.24). It is not difficult to realize that the transfer function for the signal at the LP output of $\mathrm{MM_{1a}}$, which is simultaneously the input signal of $\mathrm{MM_{1b}}$, is

$$H_{\mathrm{MM1aLP}}(s) = H_{\mathrm{LP}}(s) \cdot \left(\frac{1}{s+1}\right)^{-1} = \frac{s+1}{s^2 + (2-k)s + 1}$$

respectively for the signal at the HP output of $\mathrm{MM_{1a}}$ we have

$$H_{\mathrm{MM1aHP}}(s) = s \cdot H_{\mathrm{MM1aLP}}(s) = \frac{(s+1)s}{s^2 + (2-k)s + 1}$$

[9]The author has used the works of Tim Stinchcombe as the information source on the Sallen–Key filter. The idea to introduce TSK filters as a systematic concept arose from discussions with Dr. Julian Parker.

Thus we obtain

$$H_{\mathrm{HP}}(s) = H_{\mathrm{MM1aHP}}(s) - H_{\mathrm{BP}}(s) = \frac{(s+1)s}{s^2+(2-k)s+1} - \frac{s}{s^2+(2-k)s+1} =$$
$$= \frac{s^2}{s^2+(2-k)s+1}$$

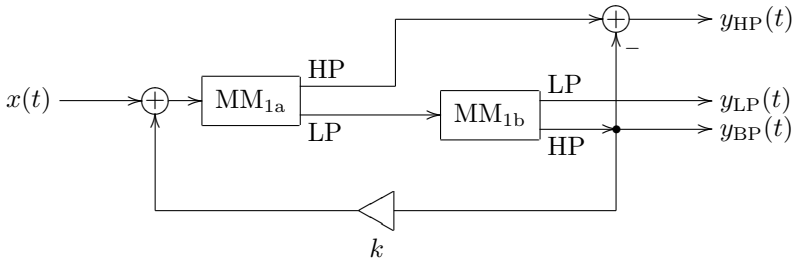


Figure 5.24: Fully multimode TSK filter.

Alternative representations

Recall that 1-pole highpass signal can be obtained as the difference of the 1-pole lowpass filter's input and output signals:

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

Then we can replace the multimode 1-pole in Fig. 5.23 by a 1-pole lowpass, constructing the highpass signal "manually" by subtracting the lowpass output from the lowpass input (Fig. 5.25). A further modification of Fig. 5.25 is formally using negative feedback (Fig. 5.26)

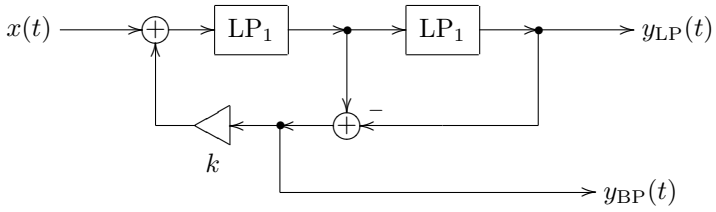


Figure 5.25: TSK filter (alternative representation).

Highpass TSK filter

Let's take the filter in Fig. 5.21 and switch the order of lowpass and highpass 1-pole filters (Fig. 5.27). Since this doesn't change the transfer function of the entire chain of 1-poles, the filter output stays the same, it is still a 2-pole bandpass.

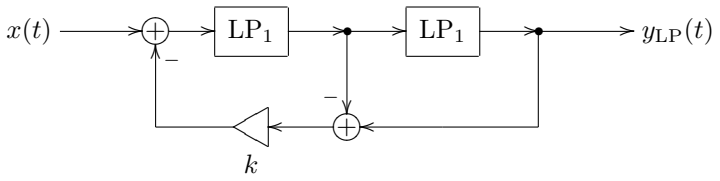


Figure 5.26: TSK filter (alternative representation, negative feedback form).

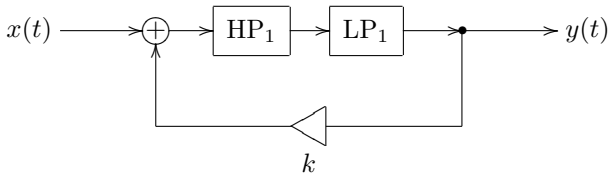


Figure 5.27: 2-pole bandpass ladder filter with a different order of 1-pole lowpass and highpass filters.

Turning the 1-pole lowpass into a multimode we obtain the structure in Fig. 5.28. It's not difficult to see that the signal at the other output of the multimode is a 2-pole highpass one. Therefore, in order to distinguish between the filters in Figs. 5.23 and 5.28 we will refer to the former more specifically as a *lowpass TSK filter* and to the latter as a *highpass TSK filter*. If necessary, we can add the lowpass output, using a way similar to Fig. 5.24.

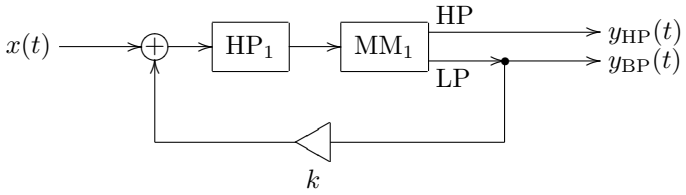


Figure 5.28: Highpass TSK filter.

The highpass versions of Fig. 5.25 and Fig. 5.26 could have been built by performing transformations of Fig. 5.28 similarly to how we did with Fig. 5.23. However it's easier just to apply the *LP to HP* substitution ($s \leftarrow 1/s$) to Figs. 5.25 and 5.26.

Sallen–Key filter (SKF)

We could take the structure in Fig. 5.27 and convert the 1-pole highpass filter into a tranposed multimode 1-pole (Fig. 5.29). By doing this one obtains a transpose of Fig. 5.23 which is (apparently) called *Sallen–Key filter* or shortly *SKF*. If necessary, the highpass input can be added, turning Fig. 5.23 into a transpose of Fig. 5.24.

If instead we take the structure in Fig. 5.21 and convert the lowpass into a

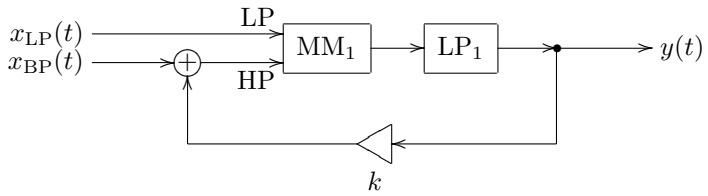


Figure 5.29: Sallen–Key filter.

transposed multimode 1-pole, we can obtain the structure in Fig. 5.30. In order to distinguish between Fig. 5.29 and Fig. 5.30, we will, as we did with their transposes, refer to the structure in Fig. 5.29 more specifically as a *lowpass Sallen–Key filter* and to the structure in Fig. 5.30 as a *highpass Sallen–Key filter*. The lowpass input can be added to the highpass SKF using the transposed version of the idea of Fig. 5.24.

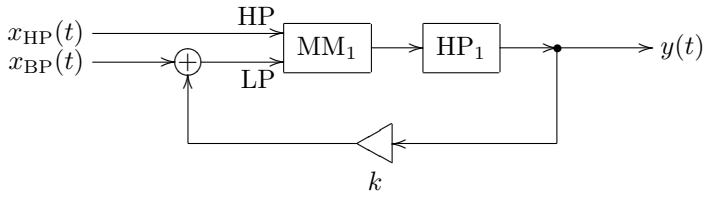


Figure 5.30: Highpass SKF.

The transposed versions of Fig. 5.25 and Fig. 5.26 make alternative representations of the lowpass SKF. E.g. by transposing the structure in Fig. 5.25 we obtain the one in Fig. 5.31.

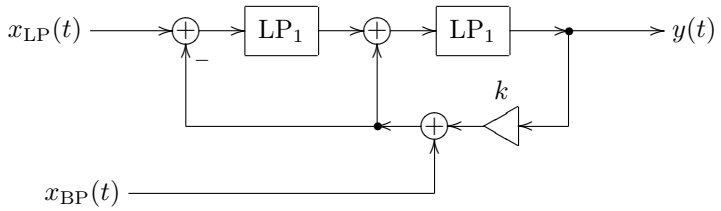


Figure 5.31: Sallen–Key filter (alternative representation).

MIMO Sallen–Key filters

By turning both 1-poles in Fig. 5.27 into multimodes we'll obtain a MIMO (multiple input multiple output) Sallen–Key filter, as illustrated in Fig. 5.32.

Note that the labelling of the inputs and outputs x_{LP}, x_{HP}, y_{LP}, y_{HP} is thereby formal. The actual transfer functions are defined for signal paths from a given input to a given output:

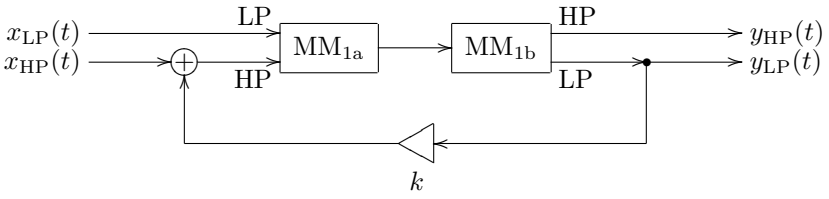


Figure 5.32: MIMO Sallen–Key filter (HP-LP).

	y_{LP}	y_{HP}
x_{LP}	2-pole lowpass	2-pole bandpass
x_{HP}	2-pole bandpass	2-pole highpass

By putting the feedback path around lowpass-highpass chain rather than lowpass-highpass, Fig. 5.32 is turned into Fig. 5.33.

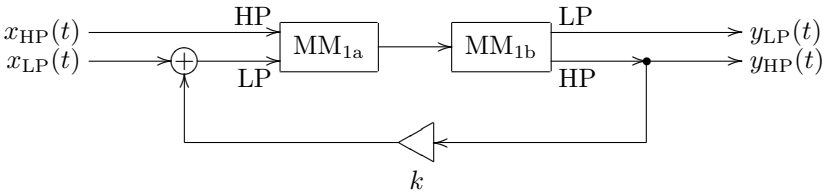


Figure 5.33: MIMO Sallen–Key filter (LP-HP).

Allpass TSK/SKF

Consider again the 2-pole bandpass ladder filter structure in Fig. 5.21. Suppose that we use 1-pole allpasses $(1-s)/(1+s)$ instead of low- and highpass filters. We also use negative, rather than positive feedback, although this is more a matter of convention. The result is shown in Fig. 5.34, where we also prepared the modal outputs.

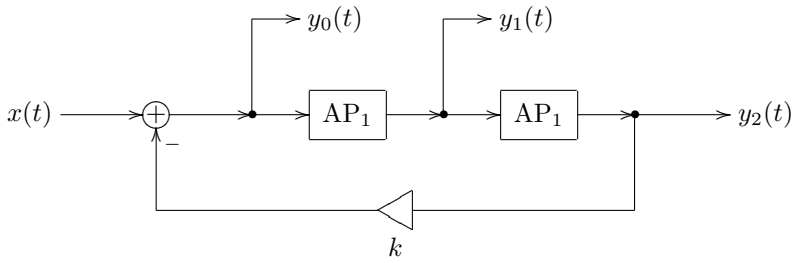


Figure 5.34: 2-pole ladder filter based on allpasses (not so useful).

The transfer function of the main output is

$$H_2(s) = \frac{\left(\frac{1-s}{1+s}\right)^2}{1+k\left(\frac{1-s}{1+s}\right)^2} = \frac{(1-s)^2}{(1+s)^2+k(1-s)^2} =$$
$$= \frac{(1-s)^2}{(1+k)s^2+2(1-k)s+(1+k)} = \frac{1}{1+k}\cdot\frac{(1-s)^2}{s^2+2\frac{1-k}{1+k}s+1}$$

which is not exactly a 2-pole allpass transfer function. The denominator of $H(s)$ however looks pretty usable, it's a classical 2-pole transfer function denominator with damping $R=(1-k)/(1+k)$.

The transfer functions at the other two outputs can be obtained by "reverse application" of the transfer functions of the 1-pole allpasses to $H_2(s)$:

$$H_1(s) = \left(\frac{1-s}{1+s}\right)^{-1}\cdot H_2(s) = \frac{1}{1+k}\cdot\frac{(1+s)(1-s)}{s^2+2\frac{1-k}{1+k}s+1}$$

$$H_0(s) = \left(\frac{1-s}{1+s}\right)^{-1}\cdot H_1(s) = \frac{1}{1+k}\cdot\frac{(1+s)^2}{s^2+2\frac{1-k}{1+k}s+1}$$

We can try building the desired transfer function

$$H(s) = \frac{s^2-2Rs+1}{s^2+2Rs+1} = \frac{s^2-2\frac{1-k}{1+k}s+1}{s^2+2\frac{1-k}{1+k}s+1}$$

as a linear combination of $H_0(s)$, $H_1(s)$ and $H_2(s)$:

$$a_0H_0(s)+a_1H_1(s)+a_2H_2(s) = H(s)$$

Noticing that the denominators of $H_0(s)$, $H_1(s)$, $H_2(s)$ are all identical to the desired denominator already, we can discard the common denominator from the equation and simply write:

$$a_0\frac{(1+s)^2}{1+k}+a_1\frac{(1+s)(1-s)}{1+k}+a_2\frac{(1-s)^2}{1+k} = s^2-2\frac{1-k}{1+k}s+1$$

or

$$a_0(1+2s+s^2)+a_1(1-s^2)+a_2(1-2s+s^2) = (1+k)s^2-2(1-k)s+(1+k)$$

From where $a_0=k$, $a_1=0$, $a_2=1$. Thus

$$H(s) = H_0(s)+kH_2(s) = \frac{s^2-2\frac{1-k}{1+k}s+1}{s^2+2\frac{1-k}{1+k}s+1}$$

and the corresponding structure is shown in Fig. 5.35.[1011] The main idea of this structure is very similar to the one of a TSK filter with some "embedded"

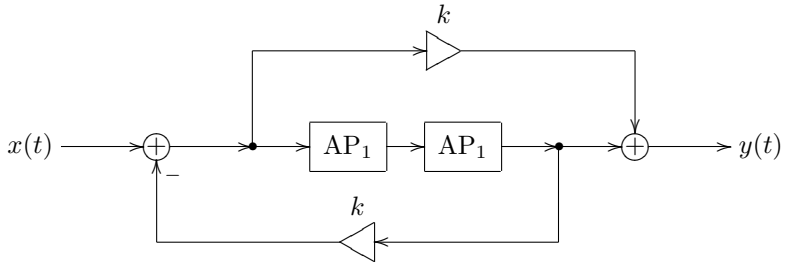


Figure 5.35: Allpass TSK filter.

modal mixture. For that reason we can refer to the filter Fig. 5.35 as a *allpass TSK* filter, or we could call it a *2-pole allpass ladder filter*.

The 2-pole damping parameter R is related to k via

$$R = (1-k)/(1+k)$$
$$k = (1-R)/(1+R)$$

so that for $k = -1 \ldots +\infty$ the damping varies from $+\infty$ to -1. The stable range $R = +\infty \ldots 0$ corresponds to $k = -1 \ldots 1$.

Transposing the structure in Fig. 5.35 we obtain the structure Fig. 5.36 which for obvious reasons we will refer to as an *allpass SKF*.

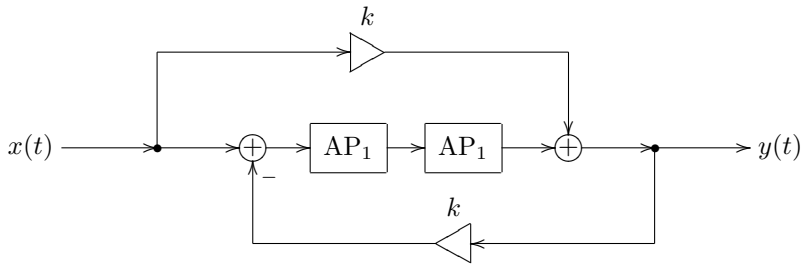


Figure 5.36: Allpass SKF.

5.9 8-pole ladder

Connecting eight 1-pole lowpass filters in series instead of four we can build an 8-pole lowpass ladder filter (Fig. 5.37).

The transfer function of the 8-pole lowpass ladder is obviously

$$H(s) = \frac{1}{k + (1+s)^8}$$

[10]It is easy to notice that this structure is very similar to the one of a multinotch filter with some specific dry/wet mixing ratio.

[11]The same structure can be obtained from a direct form II 1-pole allpass filter by the allpass substitution $z^{-1} \leftarrow (1-s)^2/(1+s)^2$. It is also interesting to notice that, applying the allpass substitution principle to the structure in Fig. 5.35, we can replace the series of the two 1-pole allpass filters in Fig. 5.35 by any other allpass filter, and the modified structure will still be an allpass filter.

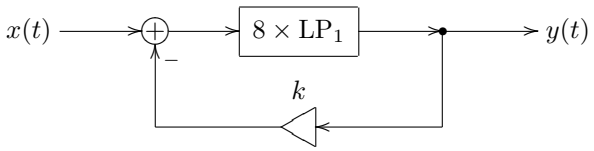


Figure 5.37: 8-pole lowpass ladder filter.

and the pole positions are defined by

$$k + (1 + s)^8 = 0$$

giving

$$s = -1 + (-k)^{1/8}$$

where $(-k)^{1/8}$ is understood in the multivalued complex root sense:

$$(-k)^{1/8} = |k|^{1/8} e^{j\alpha}$$

where

$$\alpha = \frac{\pi + 2\pi n}{8}$$

The main difference from the 4-pole ladder lowpass, besides the steeper cutoff slope, is that the 180° phase shift by the chain of 1-pole lowpasses is no longer occurring at the cutoff. Instead, the phase response of the lowpass chain at the cutoff is 360°. In order to find the frequency at which 180° phase shift is occurring we need to solve

$$\arg\left(\frac{1}{1 + j\omega}\right)^8 = -\pi$$

that is

$$\arg(1 + j\omega) = \pi/8 \quad \text{or} \quad \arg(1 + j\omega) = 3\pi/8$$

(apparently the values $5\pi/8$ an larger cannot be attained by $\arg(1 + j\omega)$). This gives

$$\omega = \tan \pi/8 \quad \text{or} \quad \omega = \tan 3\pi/8$$

The value of $\tan \pi/8$ can be easily found using the formula for the tangent of double angle:

$$\tan 2\alpha = \frac{2 \tan \alpha}{1 - \tan^2 \alpha}$$

where letting $\alpha = \pi/8$ we obtain

$$\frac{2 \tan \pi/8}{1 - \tan^2 \pi/8} = 1$$

$$2 \tan \pi/8 = 1 - \tan^2 \pi/8$$

$$\tan^2 \pi/8 + 2 \tan \pi/8 - 1 = 0$$

$$\omega = \tan \pi/8 = \sqrt{2} - 1 \approx 0.4142$$

For $\tan 3\pi/8$ we can use the formula for the tangent of the complementary angle:

$$\omega = \tan 3\pi/8 = \tan(\pi/2 - \pi/8) = \frac{1}{\tan \pi/8} = \frac{1}{\sqrt{2}-1} = \sqrt{2}+1 \approx 2.4142$$

Thus the resonance peak can occur at $\omega = \tan(\pi/4 \pm \pi/8) = \sqrt{2} \pm 1$. Let's find the values of k at which the respective poles hit the imaginary axis. According to (5.7), k is the reciprocal of the amplitude amplitude response of the chain of eight 1-pole lowpasses at the respective frequencies:

$$k = \left(\left| \frac{1}{1+j\omega} \right|^8 \right)^{-1} = \left(\frac{1}{\sqrt{1+\omega^2}} \right)^{-8} = \left(\sqrt{1+\omega^2} \right)^{-8} =$$
$$= \left(\sqrt{1+\tan^2(\pi/4 \pm \pi/8)} \right)^8 = \cos^{-8}(\pi/4 \pm \pi/8)$$

finally giving

$$\omega_1 \approx 0.4142 \qquad k_1 \approx 1.884$$
$$\omega_2 \approx 2.4142 \qquad k_2 \approx 2174$$

Thus the selfoscillation at ω_1 is occurring way much earlier than the one at ω_2. It is very unlikely that even in a nonlinear version of this filter, which allows going into unstable range of k, we will use k as large as 2174. It also hints to the fact that the second resonance is way much weaker than the first one. Therefore, for practical purposes we will simply ignore the second resonance and say that the infinite resonance is occuring at $\omega = \sqrt{2}-1 \approx 0.4142$ at $k \approx 1.884$. Fig. 5.38 illustrates the amplitude response behavior for various k.

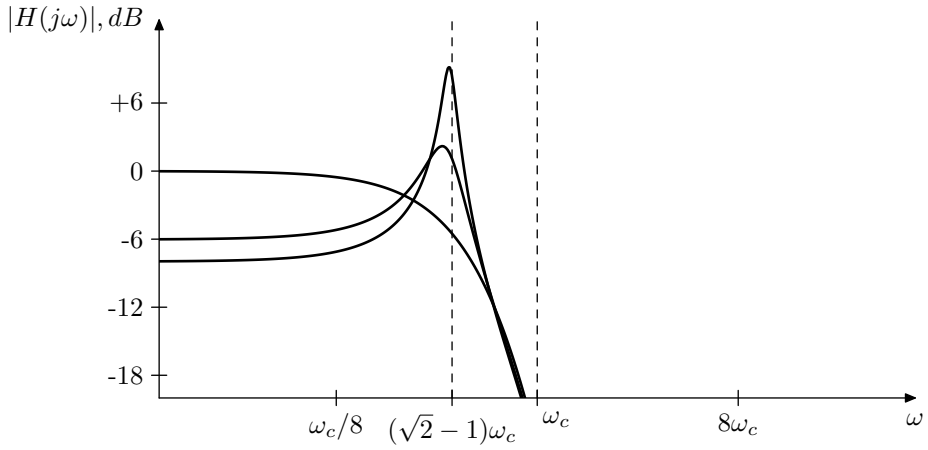


Figure 5.38: Amplitude response of the 8-pole lowpass ladder filter for various k.

Considering that at $k = 0$ the amplitude response of a chain of eight 1-poles at the cutoff is $(1/\sqrt{2})^8 = 1/16$, which is ca. -24dB, we could treat the resonance frequency $\omega = \sqrt{2}-1$ as the "user-facing" cutoff frequency instead, and in practical implementations of the filter let the cutoff of the underlying 1-poles equal the "user-facing" cutoff multiplied by $1/(\sqrt{2}-1) = \sqrt{2}+1 \approx 2.4142$.

One could ask the following question: the phase response of the chain of eight 1-poles at $\omega = 1$ is $0°$, therefore why don't we simply use positive feedback to create the resonance peak at $\omega = 1$? The problem is that the phase response at $\omega = 0$ is also $0°$. Since the amplitude response at $\omega = 0$ is 1, the selfoscillation will occur already at $k = 1$, whereas at $\omega = 1$ it will occur only at $k = 1/(1/\sqrt{2})^8 = 16$.

The instantaneously unstable range of k is found similarly to the 4-pole lowpass ladder and is $k < -1$.

Various modal mixtures for the 8-pole lowpass ladder filter can be built in a similar way to the 4-pole ladder filter. However the fact that the resonance frequency is noticeably lower than the cutoff frequency of the underlying 1-poles will affect the shapes of the resulting modal mixtures. Some smart playing around with the modal mixture coefficients can sometimes reduce the effect of this discrepancy.

8-pole highpass ladder

Replacing the 1-pole lowpasses with highpasses we obtain an 8-pole highpass ladder filter (Fig. 5.39). As we already know from the discussion of the 4-pole highpass, it essentially the same as lowpass except for the $s \leftarrow 1/s$ substitution. The instantaneously unstable range of k is found similarly to the 4-pole highpass ladder and is $k < -1$.

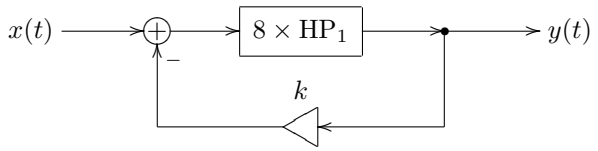


Figure 5.39: 8-pole highpass ladder filter.

8-pole bandpass ladder

Replacing half of the lowpasses with highpasses in Fig. 5.37 we obtain the 8-pole bandpass ladder filter, where we shouldn't forget that in a bandpass ladder the feedback shouldn't be inverted (Fig. 5.40).

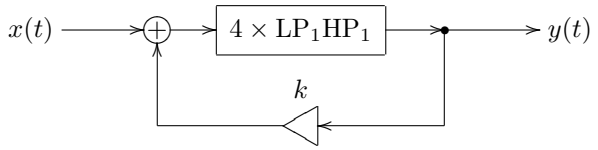


Figure 5.40: 8-pole bandpass ladder filter.

The total gain at the cutoff of the 1-pole chain is

$$\left(\frac{s}{(1+s)^2}\right)^4\Bigg|_{s=j} = \left(\frac{1}{2}\right)^4 = \frac{1}{16}$$

therefore selfoscillation occurs at $k = 16$. Fig. 5.41 illustrates the amplitude response behavior at various k. Note that the amplitude response is pretty low (particularly, for $k = 0$ it peaks at -24dB), therefore additional boosting of the output signal may be necessary in practical usage.

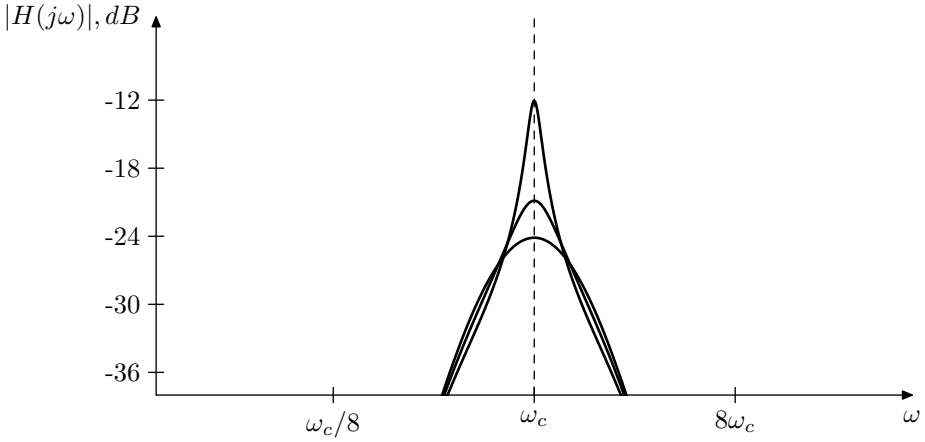


Figure 5.41: Amplitude response of the 8-pole bandpass ladder filter for various $k \geq 0$.

The instantaneously unstable range of k is found similarly to the 4-pole bandpass ladder and is $k \geq 2^8 = 256$.

An interesting feature of the 8-pole bandpass ladder is that at negative k the filter obtains two resonance peaks (Fig. 5.42).[12] Indeed, notice that the phase response of the 8-pole lowpass-highpass chain is the same as the one of the 8-pole lowpass chain:

$$\arg \frac{s^4}{(1+s)^8} = \arg \frac{1}{(1+s)^8} \qquad s = j\omega,\ \omega \in \mathbb{R}$$

Thus we still have a 180° phase shift at $\omega = \sqrt{2} \pm 1$.

The amplitude response of a single lowpass-highpass pair at $\omega = \sqrt{2} \pm 1$ is

$$\left| \left(\frac{s}{(1+s)^2} \bigg|_{s=j(\sqrt{2}\pm 1)} \right) \right| = \frac{\sqrt{2} \pm 1}{1 + (\sqrt{2} \pm 1)^2} = \frac{1}{(\sqrt{2} \mp 1) + (\sqrt{2} \pm 1)} = \frac{1}{2\sqrt{2}}$$

therefore selfoscillation occurs at $k = -(2\sqrt{2})^4 = -64$.

8-pole bandpass ladder with bandwidth control

The occurence of two resonance peaks in an 8-pole bandpass ladder at $k < 0$ motivates the introduction of the possibility to control the distance between these two peaks. In Section 5.7 we have introduced two different approaches to control the 4-pole bandpass ladder's bandwidth. Apparently, the approach using the k parameter is not good for our goal here, since we don't want to affect

[12] In nonlinear versions of this filter this can generate a particularly complex sound, as the two resonance peaks and the input signal fight for the saturation headroom.

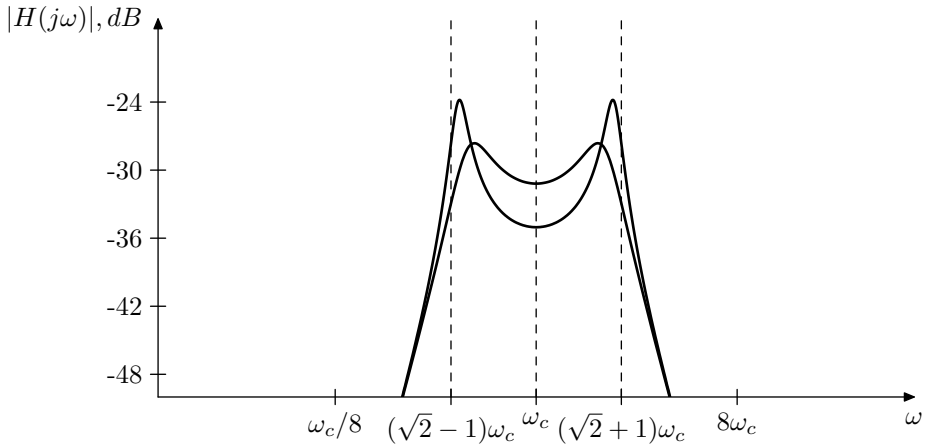


Figure 5.42: Amplitude response of the 8-pole bandpass ladder filter for various $k < 0$.

the amplitude response shape in the vertical direction. Also, from Fig. 5.42 it seems that the variation of k in the negative range has little effect on the actual bandwidth. On the other hand, the approach using the damping of the underlying 2-pole bandpasses looks much more promising.

Representing the 8-pole bandpass ladder in terms of normalized 2-pole bandpasses (Fig. 5.43) we notice that it is an LP to BP transformation of the filter in Fig. 5.44. The filter in Fig. 5.44 is essentially the same as the ordinary 4-pole lowpass ladder (Fig. 5.1), except that

- the feedback is positive, so that selfoscillation at $\omega = 1$ occurs at some negative value of k
- the output signal amplitude and the feedback amount are 16 times lower, thus selfoscillation at $\omega = 1$ doesn't occur at $k = -4$ but at $k = -64$ (which matches the already established fact of selfoscillation of Fig. 5.40 and equivalently Fig. 5.43 at $k = -64$).

Therefore by controlling the bandwidth of the LP to BP transformation, we will control the distance between the resonance peaks in Fig. 5.42.

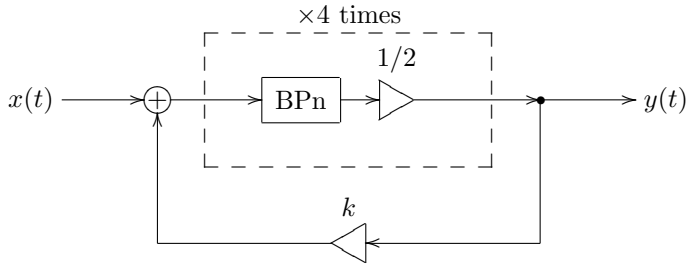


Figure 5.43: 8-pole bandpass ladder filter expressed in terms of normalized 2-pole bandpasses.

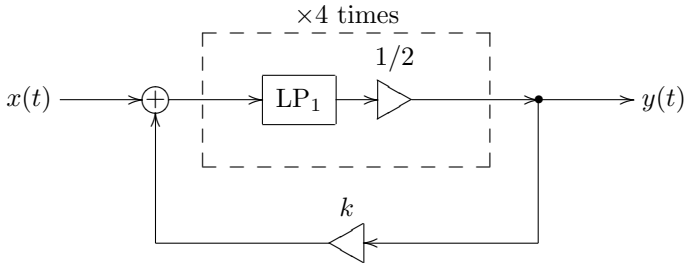


Figure 5.44: 4-pole lowpass ladder filter with positive feedback and additional gains of 1/2. The LP to BP substitution applied to this filter produces the filter in Fig. 5.43.

Since the resonance peak in Fig. 5.44 is occurring at $\omega = 1$, the formula (4.20) expresses R in terms of the distance between the two images of this peak after the LP to BP transformation. Therefore we can directly use the formula (4.20) to control the distance between the resonance peaks in Fig. 5.43. The prewarping techniques described in Section 4.6 also apply, thereby allowing us to achieve the exact positioning of the resonance peaks (in the limit $k \to -64$).

There is an important question concerning the choice of the specific topology for the normalized bandpasses BPn. Of course, the most obvious choice would be to use an SVF. This should work completely fine in the linear case. In a nonlinear case, however, we might want to use a different topology. Particularly, we might want that at $R = 1$ our controlled-bandwidth topology becomes fully identical to Fig. 5.40 (therefore obtaining the sound, which is identical to the one of the structure in Fig. 5.40 even in the presence of nonlinear effects).

Assuming that Fig. 5.40 implies interleaved 1-pole low- and highpasses (as shown in Fig. 5.45), a good solution is provided by the TSK/SKF filters. E.g. considering the structure in Fig. 5.21 (which is essentially the TSK filter from Fig. 5.23), we can notice that at $k = 0$ it becomes fully equivalent to a single lowpass-highpass pair. This suggests that we could use this structure to construct a halved normalized bandpass (Fig. 5.46), where expressing the TSK feedback k in terms of damping R we have $k = 2(1 - R)$. Note that at $R = 1$ not only the feedback path in Fig. 5.46 is disabled, but also the output gain element R is becoming transparent. Using the halved normalized bandpass in Fig. 5.46, we could reimplement Fig. 5.45 as Fig. 5.47.

5.10 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 block diagram of the diode ladder is shown in Fig. 5.48, while the diode ladder filter adds the feedback loop around that structure, feeding the fourth output of the diode ladder into the diode ladder's input (Fig. 5.49).

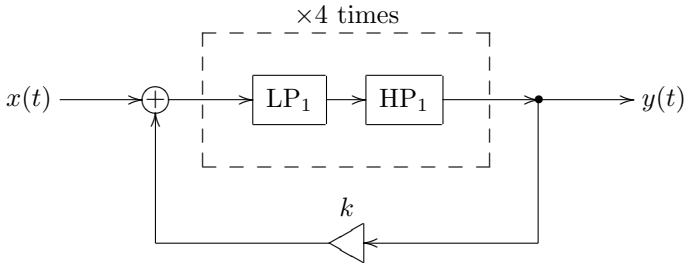


Figure 5.45: Fig. 5.40 implemented by interleaved 1-pole low- and high-passes.

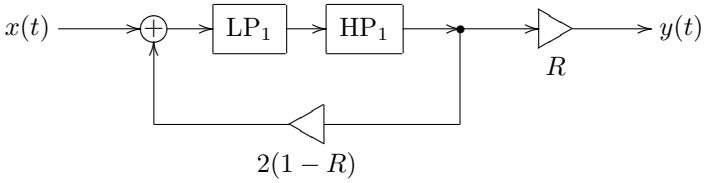


Figure 5.46: Halved normalized TSK bandpass.

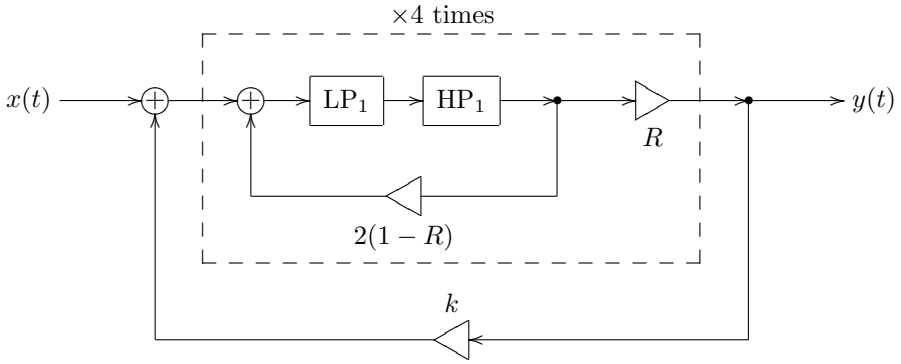


Figure 5.47: 8-pole bandpass ladder filter expressed in terms of halved normalized TSK bandpasses.

It is instructive to write out the 1-pole equations implied by Fig. 5.48:

$$\begin{aligned}
\dot{y}_1 &= \omega_c\big((x + y_2) - y_1\big) \\
\dot{y}_2 &= \omega_c\big((y_1 + y_3)/2 - y_2\big) \\
\dot{y}_3 &= \omega_c\big((y_2 + y_4)/2 - y_3\big) \\
\dot{y}_4 &= \omega_c\big(y_3/2 - y_4\big)
\end{aligned} \tag{5.18}$$

In this form it's easier to guess the reason for the gain elements 1/2 used in Fig. 5.48, they perform the averaging between the feedforward and feedback signals. However this averaging in (5.18) and Fig. 5.48 is not done fully consis-

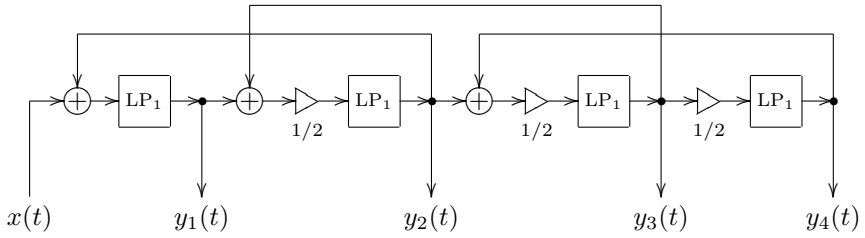


Figure 5.48: Diode ladder.

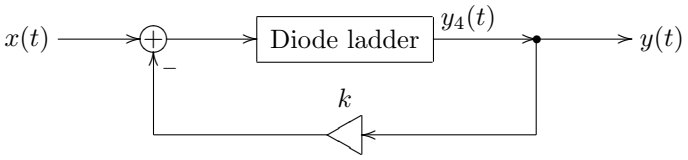


Figure 5.49: Diode ladder filter.

tently. It would have been more consistent to have no 1/2 gain element at the input of the fourth lowpass, rather than of the first one:

$$\begin{aligned}\dot{y}_1 &= \omega_c\big((x + y_2)/2 - y_1\big)\\ \dot{y}_2 &= \omega_c\big((y_1 + y_3)/2 - y_2\big)\\ \dot{y}_3 &= \omega_c\big((y_2 + y_4)/2 - y_3\big)\\ \dot{y}_4 &= \omega_c\big(y_3 - y_4\big)\end{aligned} \tag{5.19}$$

in which case the first lowpass would take $(x + y_2)/2$ as its input, the second lowpass would take $(y_1 + y_3)/2$ as its input, the third lowpass would take $(y_2 + y_4)/2$ as its input, and the fourth lowpass would take y_3 as its input. However, (5.18) is a more traditional way to implement a diode ladder filter. Anyway, the difference between (5.18) and (5.19) is actually not that large, since (as we are going to show below) they result in one and the same transfer function,

The more complicated connections between the 1-pole lowpasses present in the diode ladder "destroy" the frequency response of the ladder in a remarkable form, which, is responsible for the characteristic diode ladder filter sound.[13] Generally, the behavior of the diode ladder filter is less "straightforward" than the one of the transistor ladder filter.

Transfer function

We are going to develop the transfer function for the diode ladder in a generalized form (Fig. 5.50), where $H_n(s)$ denote blocks with respective transfer functions. In the case of Fig. 5.48 and (5.18) we would have

$$H_1(s) = G(s) \qquad H_2(s) = H_3(s) = H_4(s) = \frac{G(s)}{2} \tag{5.20}$$

[13]One could argue that the characteristic sound of diode ladder filters is due to nonlinear behavior, however the nonlinear aspects do not show up unless the filter is driven hot enough.

while in the case of (5.19) we would respectively have

$$H_1(s) = H_2(s) = H_3(s) = \frac{G(s)}{2} \qquad H_4(s) = G(s) \tag{5.21}$$

where

$$G(s) = \frac{1}{1+s} \tag{5.22}$$

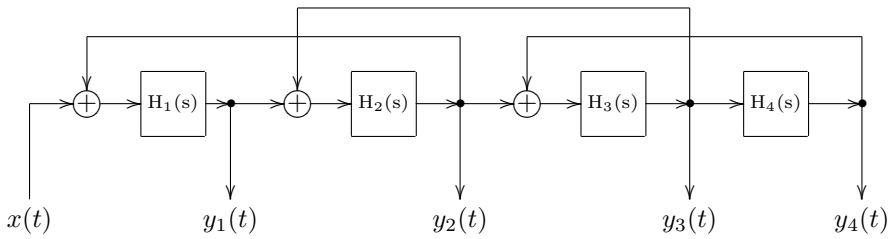


Figure 5.50: Generalized diode ladder in transfer function form.

Assuming complex exponential signals e^{st}, for the $H_4(s)$ block we have

$$y_4 = H_4 y_3$$

(where H_4 is short for $H_4(s)$), therefore

$$\frac{1}{H_4} y_4 = y_3 \tag{5.23}$$

For the $H_3(s)$ block we have

$$y_3 = H_3(y_2 + y_4)$$

Substituting (5.23) we have

$$\frac{1}{H_4} y_4 = H_3(y_2 + y_4)$$

$$\frac{1}{H_{34}} y_4 = y_2 + y_4$$

$$\frac{1 - H_{34}}{H_{34}} y_4 = y_2 \tag{5.24}$$

where H_34 is a short notation for $H_3 H_4$.

For the $H_2(s)$ block we have

$$y_2 = H_2(y_1 + y_3)$$

Substituting (5.23) and (5.24) we have

$$\frac{1 - H_{34}}{H_{34}} y_4 = H_2 \left(y_1 + \frac{1}{H_4} y_4 \right)$$

$$\frac{1-H_{34}}{H_{234}}y_4 = y_1 + \frac{1}{H_4}y_4$$
$$\frac{1-H_{34}-H_{23}}{H_{234}}y_4 = y_1 \tag{5.25}$$

For the $H_1(s)$ block we have

$$y_1 = H_1(x + y_2)$$

Substituting (5.24) and (5.25) we have

$$\frac{1-H_{34}-H_{23}}{H_{234}}y_4 = H_1\left(x + \frac{1-H_{34}}{H_{34}}y_4\right)$$
$$\frac{1-H_{34}-H_{23}}{H_{1234}}y_4 = x + \frac{1-H_{34}}{H_{34}}y_4$$
$$\frac{1-H_{34}-H_{23}-H_{12}(1-H_{34})}{H_{1234}}y_4 = x$$
$$\frac{1-H_{12}-H_{23}-H_{34}+H_{1234}}{H_{1234}}y_4 = x$$
$$\Delta(s) = \frac{y_4}{x} = \frac{H_{1234}}{1-H_{12}-H_{23}-H_{34}+H_{1234}} \tag{5.26}$$

where $\Delta(s)$ is the diode ladder's transfer function. It is easy to see that substituting (5.20) or (5.21) into (5.26) gives identical results, therefore transfer functions arising out of (5.18) and (5.19) are identical. Formula (5.26) also gives one more hint at the reason to use a 1/2 gain with all 1-poles except the first or the last one, as in this case we get unit amplitude response at $\omega = 0$:

$$\Delta(0) = \frac{\frac{1}{8}}{1-\frac{1}{2}-\frac{1}{4}-\frac{1}{4}+\frac{1}{8}} = 1$$

Since we are specifically interested in Fig. 5.48, let's write its transfer function in a more detailed form. Substituting first (5.20) and then (5.22) into (5.26) we have

$$\Delta(s) = \frac{G^4/8}{1-G^2+G^4/8} = \frac{1}{8G^{-4}-8G^{-2}+1} =$$
$$= \frac{1}{8(1+s)^4-8(1+s)^2+1} = \frac{1}{T_4(s+1)} \tag{5.27}$$

where $T_4(x) = 8x^4 - 8x^2 + 1$ is the fourth-order Chebyshev polynomial.[14] The poles of $\Delta(s)$ are therefore found from $s + 1 = x_n$ or $s = -1 + x_n$ where $x_n \in (-1, 1)$ are the roots of the Chebyshev polynomial $T_4(x)$:

$$x_n = \pm\frac{1}{2} \pm \frac{1}{2\sqrt{2}}$$

[14] Although the denominator of $\Delta(s)$ is a Chebyshev polynomial, this has nothing to do with Chebyshev filters, despite the name.

Therefore the poles of $\Delta(s)$ are purely real and located within $(-2, 0)$:

$$p_n = -1 \pm \frac{1}{2} \pm \frac{1}{2\sqrt{2}} \tag{5.28}$$

Since the poles of $\Delta(s)$ are located on the negative real semiaxis and there are no zeros, $|\Delta(j\omega)|$ is monotonically decreasing to zero on $\omega \in [0, +\infty)$. Thus $\Delta(s)$ is a lowpass.[15]

In Section 2.16 we have seen that two linear systems sharing the same transfer function are equivalent as long as the only modulation which is happening is the cutoff modulation. Therefore, as long as our implementation is purely linear, we could replace the complicated diode ladder feedback system in Fig. 5.50 with simply a serial connection of four 1-poles, whose cutoffs are defined by (5.28).[16] Further details of replacement of the diode ladder by a series of 1-poles can be taken from Section 8.2 where general principles of building serial filter chains are discussed.

The transfer function of the diode ladder filter is obtained from (5.27) giving

$$H(s) = \frac{\Delta}{1 + k\Delta} = \frac{1}{k + \Delta^{-1}} = \frac{1}{k + T^4(1+s)} = \frac{1}{8(1+s)^4 - 8(1+s)^2 + 1 + k} \tag{5.29}$$

The corresponding amplitude response is plotted in Fig. 5.51.

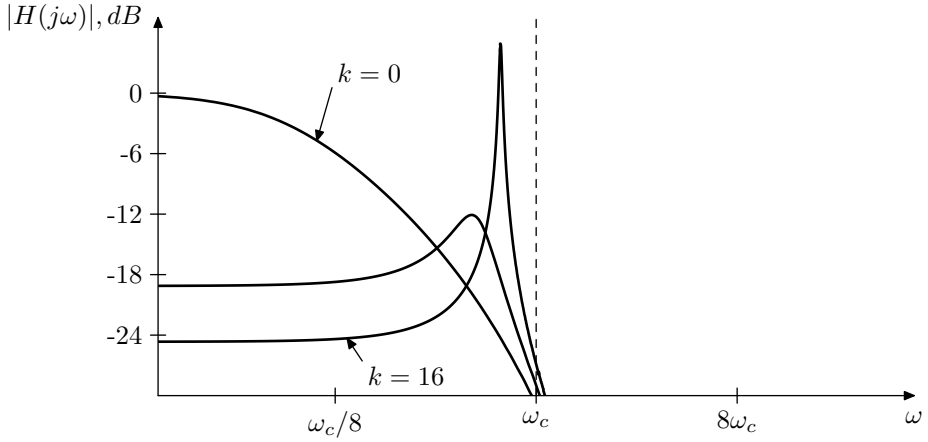


Figure 5.51: Amplitude response of the diode ladder filter for various k.

The poles of the diode ladder filter, if necessary, can be obtained by solving

$$8(1+s)^4 - 8(1+s)^2 + 1 + k = 0$$

which is a biquadratic equation in $(1 + s)$.

[15] The amplitude response of $\Delta(s)$ can be seen in Fig. 5.51 at $k = 0$.

[16] Note that such replacement only gives a correct modal output y_4, which is the one we usually need. Other modal outputs, if needed at all, would have to be obtained in a more complicated way by combining the output signals of the 1-poles.

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.

Resonance

In order to obtain the information about the resonating peak, we need to find frequencies at which the phase response of $\Delta(s)$ is $0°$ or $180°$. Therefore we are interested in the solutions to the equation

$$\operatorname{Im}\big(8(1+s)^4 - 8(1+s)^2 + 1\big) = 0 \qquad \text{where } s = j\omega,\ \omega \in \mathbb{R}$$

Substituting $j\omega$ for s we have

$$\begin{aligned}\operatorname{Im}\big(8(1+s)^4 - 8(1+s)^2 + 1\big) &= 8\operatorname{Im}\big((1+j\omega)^4 - (1+j\omega)^2\big) = \\ &= \operatorname{Im}\big((1-\omega^2+2j\omega)^2 - (1-\omega^2+2j\omega)\big) = 4(1-\omega^2)\omega - 2\omega = 0\end{aligned}$$

The solution $\omega = 0$ is not very interesting. Therefore we cancel the common factor 2ω obtaining

$$2(1-\omega^2) = 1$$

and therefore

$$\omega = \pm\frac{1}{\sqrt{2}}$$

Now, in order to find the selfoscillation boundary value of k we need to find the frequency response of $\Delta(s)$ at $\omega = 1/\sqrt{2}$. Substituting $s = j/\sqrt{2}$ into (5.27) and using (5.6) we have

$$\begin{aligned}k &= 8(1+s)^4 - 8(1+s)^2 + 1 = 8\left(1+\frac{j}{\sqrt{2}}\right)^4 - 8\left(1+\frac{j}{\sqrt{2}}\right)^2 + 1 = \\ &= 8\left(\frac{1}{2}+j\sqrt{2}\right)^2 - 8\left(\frac{1}{2}+j\sqrt{2}\right) + 1 = \\ &= 8\left(-\frac{7}{4}+j\sqrt{2}\right) - 8\left(\frac{1}{2}+j\sqrt{2}\right) + 1 = 1 - 14 - 4 = -17\end{aligned}$$

Now, since we are already having negative feedback in Fig. 5.48, the selfoscillation occurs at $k = 17$.

Note that the amplitude response in Fig. 5.51 is matching the above analysis results.

TPT model

Converting Fig. 5.48 to the instantaneous response form we obtain the structure in Fig. 5.52. From Fig. 5.52 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. 5.49.

The structure in Fig. 5.52 looks a bit complicated to solve. Of course we could always write a system of linear equations and solve it in a general way, e.g. using Gauss elimination, but this has its own complications. Therefore we

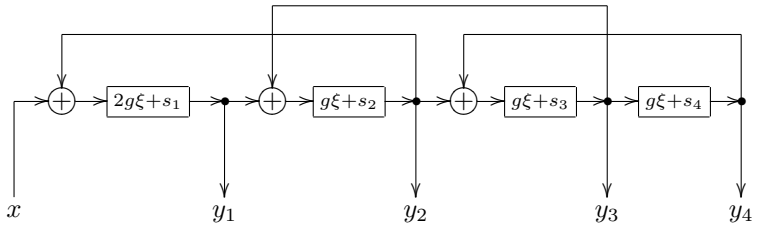


Figure 5.52: Diode ladder in the instantaneous response form.

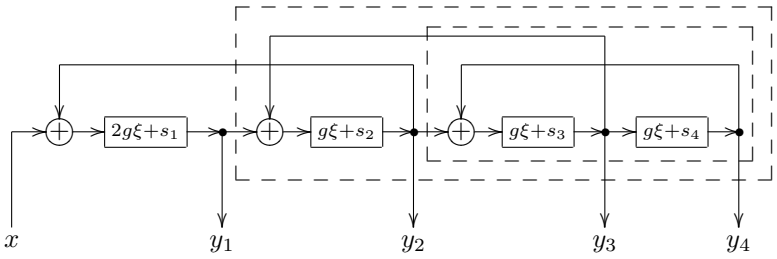


Figure 5.53: Diode ladder in the nested instantaneous response form.

would rather like to see if we somehow could still use the approach of nested zero-delay feedback loops, as we have been doing with other filters until now.

Introducing the nested systems, as shown in Fig. 5.53 by dashed lines, we can first treat the innermost system which has input y_2 and outputs y_3 and y_4. The equations for this system are

$$\begin{aligned} y_3 &= g(y_2 + y_4) + s_3 \\ y_4 &= gy_3 + s_4 \end{aligned}$$

Solving for y_3, we obtain

$$y_3 = \frac{g}{1-g^2}y_2 + \frac{gs_4 + s_3}{1-g^2} = g_{23}y_2 + s_{23}$$

where g_{23} and s_{23} are new variables introduced as shown above. Since $g\xi + s_n$ denote 1-pole lowpasses with halved input signals, $0 < g < 1/2$. Respectively $0 < g^2 < 1/4$ and thus the zero-delay feedback loop doesn't get instantaneously unstable. The range of g_{23} is

$$0 < g_{23} = \frac{g}{1-g^2} < \frac{1/2}{1-(1/2)^2} = \frac{1/2}{3/4} = \frac{2}{3}$$

Going outside to the next nesting level we have

$$y_2 = g(y_1 + y_3) + s_2 = gy_3 + gy_1 + s_2 = g(g_{23}y_2 + s_{23}) + gy_1 + s_2$$

Solving for y_2:

$$y_2 = \frac{g}{1-gg_{23}}y_1 + \frac{gs_{23} + s_2}{1-gg_{23}} = g_{12}y_1 + s_{12}$$

where $0 < gg_{23} < 1/2 \cdot 2/3 = 1/3$, thus the zero-delay feedback loop doesn't get instantaneously unstable. The range of g_{12} is

$$0 < g_{12} = \frac{g}{1 - gg_{23}} < \frac{1/2}{1 - \frac{1}{2} \cdot \frac{2}{3}} = \frac{1/2}{1 - 1/3} = \frac{1/2}{2/3} = \frac{3}{4}$$

Going outside to the outermost level we have

$$y_1 = 2g(x + y_2) + s_1 = 2gy_2 + 2gx + s_1 = 2g(g_{12}y_1 + s_{12}) + 2gx + s_1$$

Solving for y_1:

$$y_1 = \frac{2g}{1 - 2gg_{12}}x + \frac{2gs_{12} + s_1}{1 - 2gg_{12}} = g_{01}x + s_{01}$$

where $0 < 2gg_{12} < 2 \cdot 1/2 \cdot 3/4 = 3/4$, thus the zero-delay feedback loop doesn't get instantaneously unstable. The range of g_{01} is

$$0 < g_{01} = \frac{2g}{1 - 2gg_{12}} < \frac{1}{1 - 2 \cdot \frac{1}{2} \cdot \frac{3}{4}} = \frac{1}{1 - 3/4} = \frac{1}{1/4} = 4$$

Introducing for consistency the notation $y_4 = gy_3 + s_4 = g_{34}y_3 + s_{34}$, we obtain the instantaneous response for the entire ladder

$$\begin{aligned}
y_4 &= g_{34}y_3 + s_{34} = \\
&= g_{34}(g_{23}y_2 + s_{23}) + s_{34} = g_{34}g_{23}y_2 + (g_{34}s_{23} + s_{34}) = g_{24}y_2 + s_{24} = \\
&= g_{24}(g_{12}y_1 + s_{12}) + s_{24} = g_{24}g_{12}y_1 + (g_{24}s_{12} + s_{24}) = g_{14}y_1 + s_{14} = \\
&= g_{14}(g_{01}x + s_{01}) + s_{14} = g_{14}g_{01}x + (g_{14}s_{01} + s_{14}) = g_{04}x + s_{04}
\end{aligned}$$

it's not difficult to realize that

$$0 < g_{04} = g_{01}g_{12}g_{23}g_{34} < 4 \cdot \frac{3}{4} \cdot \frac{2}{3} \cdot \frac{1}{2} = 1$$

Now g_{04} is the instantaneous gain of the entire diode ladder. Respectively the total gain of the of the zero-delay feedback loop in Fig. 5.49 is $-kg_{04}$ and thus the feedback doesn't get instantaneously unstable provided $k \geq -1$.

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.

The same idea of a feedback loop around a chain of several filters also results in further filter types such as 8-pole ladder, diode ladder and SKF/TSK.

Chapter 6

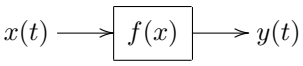
Nonlinearities

The filters which we were discussing until now were all linear. Formally this means that if we consider a filter as an operator, this operator is a linear one. Practically this meant that the structures of our filters were consisting of gains, summators and integrators. However, filters used in synthesizers often show noticeably nonlinear behavior. In terms of block diagrams, introducing nonlinear behavior means that we should add nonlinear elements to the set of our block diagram primitives.

Nonlinear filters have more complicated behavior and are capable of producing richer sound than the linear ones. Usually they exhibit complex overdriving effects, when driven with an input signal of a sufficiently high level. Another special feature of many nonlinear filters is their ability to increase the resonance beyond a formally infinite amount, entering the so-called *self-oscillation*.

6.1 Waveshaping

We just mentioned that in order to build non-linear filters we need to introduce nonlinear elements into the set of our block diagram primitives. In fact we are going to introduce just one new type of element, the *waveshaper*:

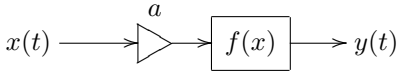


A waveshaper is simply applying a given function to its input signal, and sends the respective function value as its output signal:

$$y(t) = f(x(t))$$

The function $f(x)$ can be any "reasonable" function, e.g. $f(x) = |x|$ or $f(x) = \sin x$ etc.

Usually the function f cannot vary with time, that is, the function's parameters, if it has any, are fixed. E.g. if $f(x) = \sin ax$, then a is usually fixed to some particular value, e.g. $a = 2$, which doesn't vary. Often, this is just a matter of convention. e.g. the waveshaper $\sin ax$ can be represented as a serial connection of a gain element and the waveshaper itself:



in which case the waveshaper itself is time-invariant.

Still, if necessary, it's no problem for the waveshaper to contain time-varying parameters, as long as the time-varying parameters are "externally controlled" (in the same way how e.g. filter cutoff is controlled). That is, the waveshaper's parameters cannot depend on the values of the signals within the block diagram. If one needs the parameter dependency on the signals of the block diagram, then one should consider such dependencies as additional inputs of the nonlinear element and we end up with a multi-input element of the block diagram. It is no problem to use such elements, but normally we should not refer to them as waveshapers, since commonly, waveshapers have one input and one output.

In order to be representable as a function of the input signal, a waveshaper clearly shouldn't have any dependency on its own the past. That is waveshaper is a *memoryless* element.

6.2 Saturators

The probably most commonly used category of waveshapers is *saturators*. There is no precise definition of what kind of waveshaper is referred to as saturator, it's easier to give an idea of what a saturator is by means of example.

Bounded saturators

One of the most classical saturators is the hyperbolic tangent function:

$$y(t) = \tanh x(t) \tag{6.1}$$

(Fig. 6.1). Even if the input signal of this saturator is very large, the output never exceeds ± 1. Thus, this element *saturates* the signal, which is the origin of the term *saturator*.

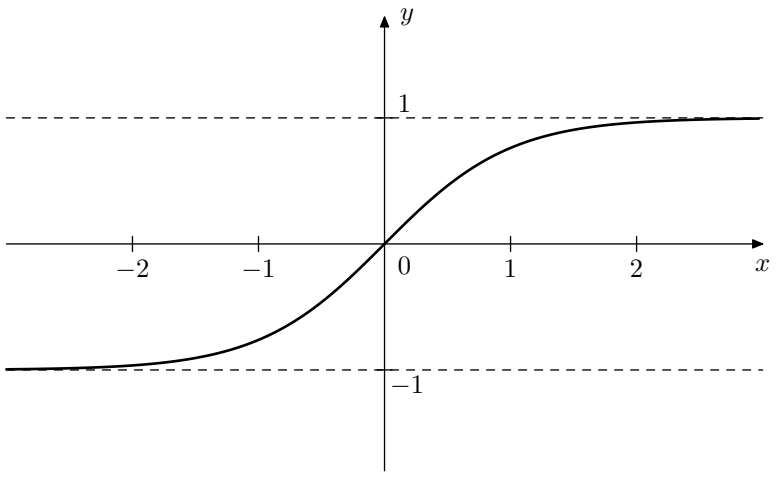


Figure 6.1: Hyperbolic tangent $y = \tanh x$.

Other saturators with shapes similar to the hyperbolic tangent include:

$$y = \sin \arctan x = x/\sqrt{1 + x^2} \tag{6.2a}$$

$$y = \begin{cases} x \cdot (1 - |x|/4) & \text{if } |x| \le 2 \\ \operatorname{sgn} x & \text{if } |x| \ge 2 \end{cases} \qquad \text{(Parabolic saturator)} \tag{6.2b}$$

$$y = x/(1 + |x|) \qquad \text{(Hyperbolic saturator)} \tag{6.2c}$$

(this list is by no means exhaustive). Is is not difficult to see that the values of the hyperbolic tangent (6.1) and the saturators (6.2) do not exceed 1 in absolute magnitude. That is, their ranges are bounded. We are going to refer to such saturators as *bounded-range saturators* or simply *bounded saturators*.

From the four introduced saturation functions the parabolic saturator (6.2b) stands out in that the full saturation is achieved at $|x| = 2$, whereas for other shapes it's not achieved at finite input signal levels. Thus, the range of (6.2b) is $[-1, 1]$, therefore being compact. We will refer to such saturators as *compact-range monotonic saturators*.

Another important distinction of the parabolic saturator is that it has three discontinuities of the second derivative (at $x = 0$ and $x = \pm 2$) and the hyperbolic saturator has one discontinuity of the second derivative (at $x = 0$). Even though such discontinuities are not easily visible on the graph, they affect the character of the saturator's output signal. Usually such discontinuities are rather undesired, as they represent abrupt irregularities in the saturator's shape, so it's generally better to avoid those.[1] A common reason to tolerate derivative discontinuities in a saturator, though, is performance optimization.

Transparency at low signal levels

A property commonly found with saturators is that at low levels of input signals the saturator is transparent: $f(x) \approx x$ for $x \approx 0$. Equivalently this condition can be written as

$$\begin{aligned} f(0) &= 0 \\ f'(0) &= 1 \end{aligned} \tag{6.3}$$

Visually it manifests itself as the function's graph going at 45° through the origin. Clearly, all the previously introduced saturators have this property.

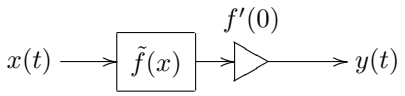
The property (6.3) is not really a must, but it's quite convenient if the saturators have it, particularly for the analysis of system behavior at low signal levels. For that reason it's common to represent a non-unit derivative at the origin via a separate gain. Given a saturation function $f(x)$ such that $f(0) = 0$ but $f'(0) \ne 1$ we introduce a different saturation function $\tilde{f}(x)$ such that $\tilde{f}(0) = 0$ and $\tilde{f}'(0) = 1$. E.g. we can take

$$\tilde{f}(x) = \frac{f(x)}{f'(0)}$$

so that

$$f(x) = f'(0)\tilde{f}(x)$$

The coefficient $f'(0)$ is then represented as a separate gain element.



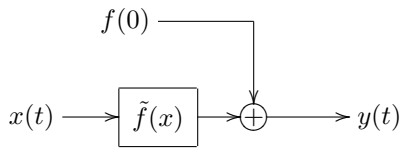
[1] Sometimes the effect created by discontinuities is explicitly being sought after, e.g. in a rectification waveshaper $f(x) = |x|$.

Apparently this representation is not available if $f'(0) = 0$, however in such cases the saturator effectively breaks the connection at low signal levels, working as a zero gain.

Saturators with $f(0) \neq 0$ can be represented by separation of the value $f(0)$ into a DC offset signal:

$$f(x) = f(0) + \tilde{f}(x)$$

which is treated as another input signal with a fixed value $f(0)$:



Unbounded saturators

Sometimes we want saturation behavior, but do not want a hard bound on the output signal's level. One function with this property is inverse hyperbolic sine:

$$y = \sinh^{-1} x = \ln\left(x + \sqrt{x^2 + 1}\right) \tag{6.4}$$

(Fig. 6.2) While having the usual transparency property (6.3), it is not bounded. The asymptotic behavior of the hyperbolic sine is similar to the one of the logarithm function:

$$\sinh^{-1} x \sim \operatorname{sgn} x \cdot \ln|2x| \qquad x \to \infty$$

Another saturator with a similar behavior can be obtained as an inverse of $y = x(1 + |x|)$, which is

$$y = \frac{2x}{1 + \sqrt{1 + |4x|}} \tag{6.5}$$

behaving as $\sqrt{|x|}$ at $x \to \infty$.

Such kind of waveshapers are also referred to saturators, even though the saturation doesn't have a bound. We will refer to them as *unbounded-range* or *unbounded* saturators.

Apparently, unbounded saturators represent a weaker kind of saturation than bounded ones. The weakest possible kind of saturation is achieved if y grows as a linear function of x at $x \to \infty$. Such saturators can be built by introducing a linear term into the saturator's function. Given a saturator $f(x)$ where $f(x)$ can be any of the previously discussed saturators, we build a new saturator by taking a mixture of $y = f(x)$ and $y = x$:

$$y = (1 - \alpha) f(x) + \alpha x \qquad (0 < \alpha < 1) \tag{6.6}$$

where we needed to multiply $f(x)$ by $1 - \alpha$ to keep the transparency property (6.3) (provided it was holding for $f(x)$). Apparently $y \sim \alpha x$ for $x \to \infty$. We can refer to such saturators as *asymptotically linear* saturators. The previously discussed saturators such that $y = o(x)$ for $x \to \infty$ can be respectively referred to as *slower-than-linear* saturators.

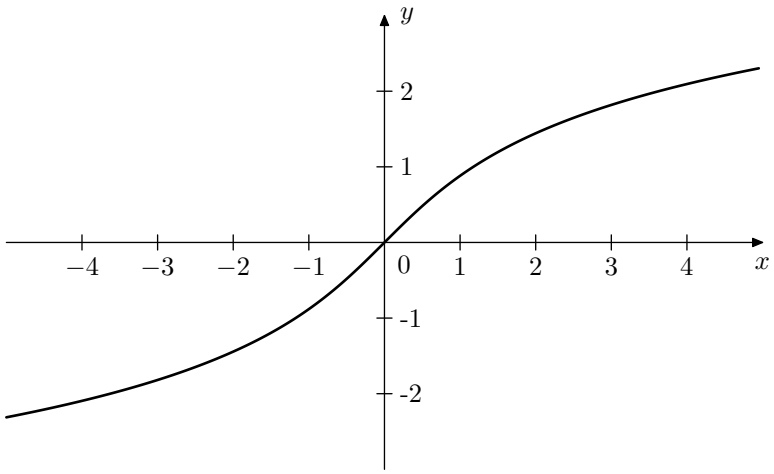


Figure 6.2: Inverse hyperbolic sine $y = \sinh^{-1} x$.

Soft- and hard-clippers

One special but important example of a saturator is the *hard clipper*, shown in Fig. 6.3.[2] In contrast, we will be referring to all previously discussed saturators as *soft clippers*.[3]

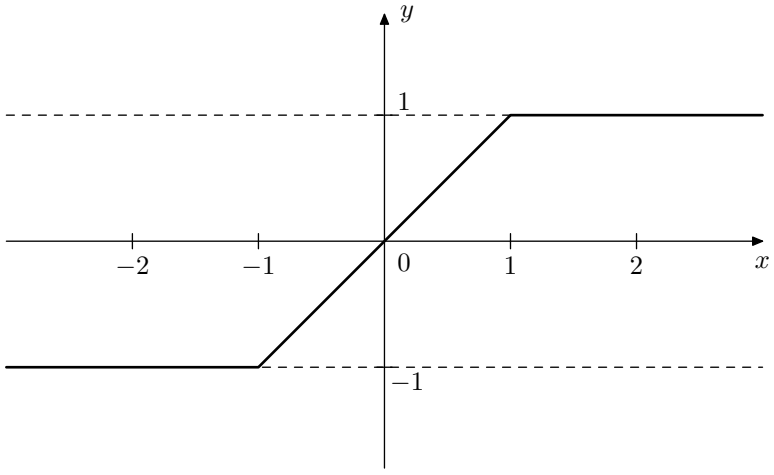


Figure 6.3: Hard clipper.

[2]Apparently, hard clipper is a compact-range saturator.

[3]There doesn't seem to be a universally accepted definition of which kinds of saturators are referred to as soft clippers, and which aren't. E.g. the set of soft clippers could be restricted to contain only bounded saturators. In this book we will understand the term *soft clipper* in the widest possible sense.

Saturation level

The previously introduced bounded saturators (6.1) and (6.2) were all saturating at $y = \pm 1$. But that is not always desirable. Given a bounded saturator $f(x)$ with the saturation level $y = \pm 1$ we can change the saturation level to $y = \pm L$ by simultaneouly scaling the x and y coordinates:

$$y(t) = L \cdot f(x/L) \tag{6.7}$$

(Fig. 6.4). The simultaneous scaling of x and y preserves the transparency property (6.3).

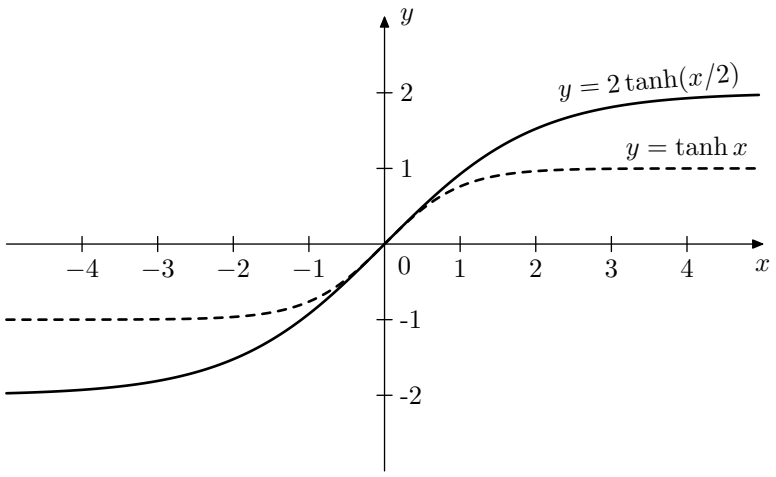


Figure 6.4: Changing the saturation level.

Saturator as variable gain

Sometimes it is useful to look at saturators as at variable gain elements. E.g. we can rewrite $y = \tanh x$ as

$$y = \tanh x = x \cdot \frac{\tanh x}{x} = g(x) \cdot x \tag{6.8}$$

The graph of the function $g(x) = \frac{\tanh x}{x}$ is shown Fig. 6.5. Thus

$$g(x) \approx 1 \qquad \text{for } x \approx 0 \tag{6.9a}$$
$$g(x) \sim 1/|x| \qquad \text{for } x \to \infty \tag{6.9b}$$

That is at low signal levels the saturator is transparent, at high signal levels is reduces the input signal's amplitude by a factor of approximately $1/|x|$. Apparently, this kind of behavior is shown by all bounded saturators. Unbounded saturators give a similar picture, as long as they are slower than linear. For asymptotically linear saturators, such as (6.6), we have $g(x) \to \alpha$ instead.

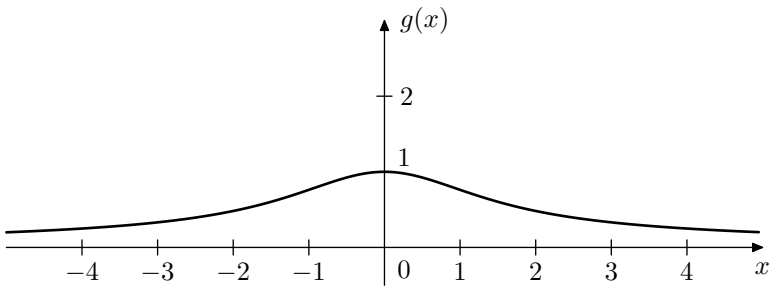


Figure 6.5: $g(x) = \dfrac{\tanh x}{x}$.

6.3 Feedback loop saturation

In the ladder filter and its variations, such as 4-pole and 8-pole ladders and SKF/TSK filters, the resonance is implemented by means of a feedback loop. By this we mean that when the feedback loop is disabled (by setting the feedback gain to zero), there is no resonance, and the resonance amount is increased by increasing the amount of the feedback (thus e.g. the SVF filter doesn't fall into this category). With such filter structures, when the feedback amount goes above a certain threshold (e.g. $k = 4$ for the 4-pole lowpass ladder or $k = 2$ for the SKF) the filter becomes unstable and "explodes" (the filter's state and the output signal indefinitely grow). By putting a saturator anywhere within such feedback loop we can prevent the signals in the feedback loop from the infinite growth, making the filter stable again.

Feedforward path saturation

One of the common positions for the feedback loop saturator is in the feedforward path right after the feedback merge point (Fig. 6.6). Given that the saturator is a bounded one (such as $\tanh x$), the output signal of such saturator is guaranteed to be bounded. Since the rest of the feedforward path in Fig. 6.6 is known to be BIBO-stable (independently of the feedback setting), the output of the filter is bounded too and thus the entire filter is BIBO-stable.

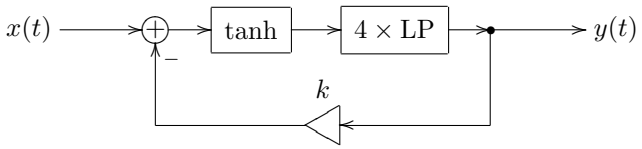


Figure 6.6: Ladder filter with a saturator in the feedforward path.

We could also view the saturator in Fig. 6.6 as a variable gain g (6.8). Apparently, as the amplitude of the signal grows, the average value of g is decreasing to zero. In those terms, the saturator is effectively reducing the feedback gain from k to $k \cdot \langle g \rangle$ (where $\langle g \rangle$ is the average value of g). Since at large signal amplitudes $\langle g \rangle$ can get arbitrarily close to zero, the value of $k \cdot \langle g \rangle$ goes below 4, which, intuitively, prevents the filter from exploding. Unbounded

saturators are therefore having the same effect, as long as they are slower than linear.

With asymptotically linear saturators the filter will still explode at some point. Assuming the saturator has the form (6.6) we have $g(x) \to \alpha$ and thus $\langle g \rangle \to \alpha$ and $k \cdot \langle g \rangle \to \alpha k$. Thus, we could expect that for something like $\alpha k < 4$ the filter should not explode.

Effects of transient response

As we should remember, one possible way to look at a filter getting unstable is that its transient response grows instead of decaying with time. Each pair of conjugate poles p_n and p_n^* of the filter contributes a transient component of the form

$$Ae^{p_n t} + A^* e^{p_n^* t} = ae^{t \operatorname{Re} p_n} \cos(t \operatorname{Im} p_n + \varphi_n)$$

Thus at $\operatorname{Re} p_n = 0$ we have a sinusoid of frequency $\omega = \operatorname{Im} p_n$. At $\operatorname{Re} p_n > 0$ we have the same sinusoid of an exponentially growing amplitude. This sinusoid will be present in the filter's output even in the absence of the input signal.[4] Since at $\operatorname{Re} p_n \geq 0$ this sinusoid is self-sustaining, the filter is said to *self-oscillate*. The saturator in the feedback loop prevents the self-oscillation from infinite growth.[5]

Suppose the system in Fig. 6.6 is at $k \approx 4$, that is it is selfoscillating or at least strongly resonating. Let

$$u(t) = x(t) - ky(t) \tag{6.10}$$

denote the input signal of the saturator and recall the representation of a saturator as a variable gain element (6.8). Then the output signal of the saturator is

$$v(t) = \tanh u(t) = g(u) \cdot u = g(u)x(t) - g(u)k \cdot y(t) = g(u)x(t) - \tilde{k}(u)y(t) \tag{6.11}$$

where $g(u) = \frac{\tanh u}{u}$. Comparing (6.10) to the last expression in (6.11) we see that the effect of the saturator can be seen as the "replacing" $x(t)$ with $g(u)x(t)$ and $ky(t)$ with $\tilde{k}(u)y(t)$. Thus, $\tilde{k}(u) = g(u)k$ is the new "effective feedback amount". Now, by increasing the amplitude of the input signal $x(t)$ we increase the amplitude of $u(t)$ and thus reduce the magnitude of $g(u)$ and thereby reduce the effective feedback amount $\tilde{k}$, which in turn shows up as reduction of resonance. That is, at high amplitudes of the input signal the resonance oscillations kind of disappear.

One intuitive way to look at this is to say that the input signal and the resonance are "fighting" for the saturator's headroom, and if the input signal

[4] If the system is in the zero state, then in the absence of the input signal it will stay forever in this state of "unstable equilibrium". In analog circuits, however, there are always noise signals present in the system, which will excite the transient response components, thus destroying the equilibrium. In the digital implementation such excitations need to be performed manually. This can be done by initializing the system to a not-exactly-zero state, or by sending a short excitation impulse into the system at the initialization, or by mixing some low-level noise at one or multiple points into the system. Often a very small constant DC offset will suffice instead of such noise.

[5] As the saturator is effectively reducing the total gain of the feedback loop, at $k = 4$ the selfoscillation will first have an infinitely small signal level, where the saturator is transparent. Increasing the value of k further we can bring the selfoscillation to an audible level.

has a very high level, it "pushes" the resonating component of the signal out. On the other hand, if the input signal level is low, then the entire headroom is taken by the resonating component which will be therefore much louder than the input signal. There is usually some "sweet spot" in the input signal's level, where the fighting doesn't kill the resonance, but results in a nice interaction between the input signal and the resonance.

Feedback path saturation

The amount of fighting (at the same input signal level) can be decreased by putting the saturator into the feedback path, either prior to the feedback gain (Fig. 6.7) or past the feedback gain (Fig. 6.8).[6] In this case the input signal $x(t)$ doesn't directly enter the saturator but first goes through the four 1-pole lowpass filters, which somewhat reduces its amplitude (depending on the signal and on the filter's cutoff). The difference between Figs. 6.7 and Fig. 6.8 is obviously that in one case the effective saturation function is $y = k \tanh x$ whereas in the other one it's $y = \tanh kx$. This means that in the first case the saturation level is $\pm k$ whereas in the second one it's fixed to ± 1 (Fig. 6.9).

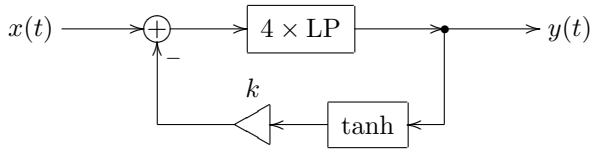


Figure 6.7: Ladder filter with a saturator in the feedback path (pre-gain).

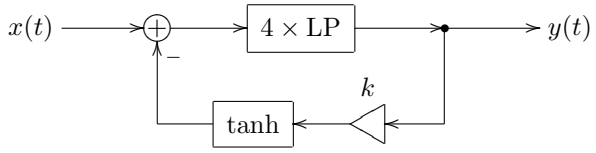


Figure 6.8: Ladder filter with a saturator in the feedback path (post-gain).

The amount of fighting will also be decreased by using a weaker saturation curve. E.g. using an unbounded saturator instead of a bounded one, in the most extreme case having an asymptotically linear saturator. A classical example of this approach is ecountered in the nonlinear Sallen–Key filter (Fig. 6.10). It is an interesting observation that the sound of nonlinear Sallen–Key filter significantly differs from the sound of nonlinear transposed Sallen–Key filter (Fig. 6.11) since in one case the saturator's output goes through a highpass and a lowpass, while in the other case it goes through two lowpasses before reaching the filter's output.[7]

[6] Notice that, with the saturator positioned in the feedback path, at $k = 0$ the filter effectively becomes linear.

[7] This observation was made by Dr. Julian Parker.

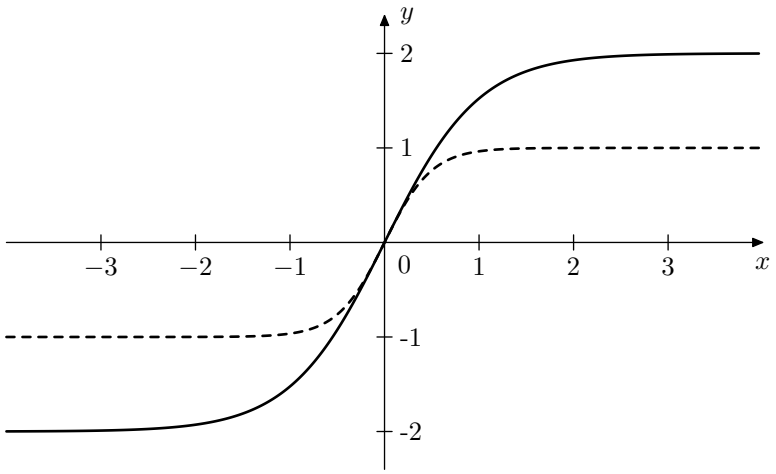


Figure 6.9: Pre-gain ($y = k \tanh x$, solid) vs. post-gain saturation ($y = \tanh kx$, dashed).

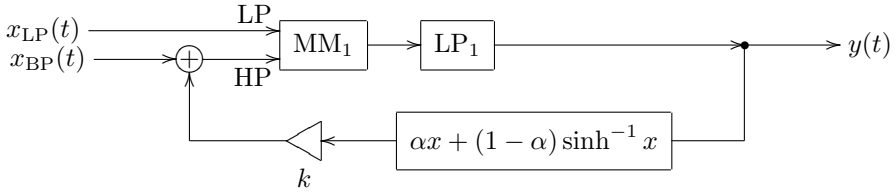


Figure 6.10: Sallen-Key filter with an asymptotically linear saturator.

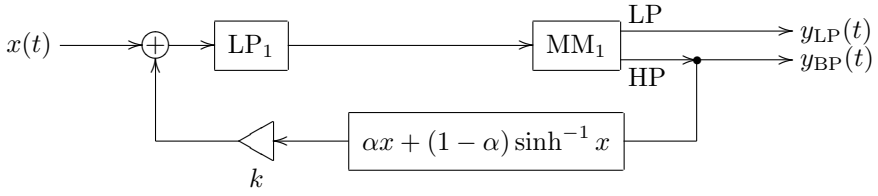


Figure 6.11: Transposed Sallen-Key filter with an asymptotically linear saturator.

Transfer function

For systems containing nonlinear elements the complex exponentials e^{st} are no longer system eigensignals. That is, given an input signal of the form Ae^{st} the output will not have a similar form. Therefore the idea of the transfer function as well as amplitude and phase responses doen't work anymore.

Still, given that the nonlinear elements satisfy the transparency condition

(6.3), at low signal levels the nonlinearities have almost no effect and the system is approximately linear. In that sense the transfer function stays applicable to a certain degree and still can be used to analyse the filter's behavior, although the error is growing stronger at higher signal levels. Nevertheless, as a rule, qualitatively the filters retain their main properties also in the presence of saturators. The lowpass filters stay lowpass, bandpass filters stay bandpass etc.

6.4 Nonlinear zero-delay feedback equation

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. 5.5 redrawn to contain the feedback nonlinearity (Fig. 6.12).

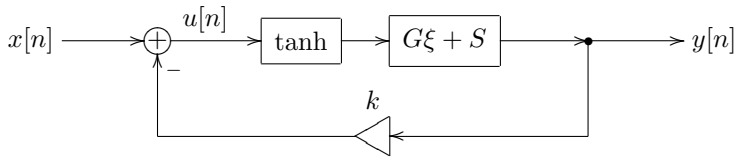


Figure 6.12: Nonlinear TPT ladder filter in the instantaneous response form.

Writing the zero-delay feedback equation we obtain

$$u = x - k(G \tanh u + S) \tag{6.12}$$

Apparently, the equation (6.12) is a transcendental one. It can be solved only using numerical methods. Also, the linear zero-delay feedback equation had only one solution, but how many solutions does (6.12) have? In order to answer the latter question, let's rewrite (6.12) as

$$(x - kS) - u = kG \tanh u \tag{6.13}$$

If $k \geq 0$ then $v(u) = kG \tanh u$ is a nonstrictly increasing function of u,[8] while $v(u) = (x - kS) - u$ is a strictly decreasing function of u. Thus, (6.13) (and respectively (6.12)) has a single solution (Fig. 6.13). At $k < 0$ we also typically have one solution (Fig. 6.14) unless $kG > -1$, in which case (6.13) has three solutions Fig. 6.15. Fortunately, $kG > -1$ corresponds to instantaneously unstable feedback, and thus normally we are not so much interested in this case anyway. However, if needed, one could use the concept of the instantenous smoothing to find out the applicable solution among the three formal ones.

Having found the zero-delay equation solution u, we proceed in the usual way, first letting u through the tanh waveshaper and then letting it through the 1-pole lowpasses (denoted as $G\xi + S$ in Fig. 6.12), updating the 1-pole states along the way and ultimately obtaining the value of y.

Now we are going to discuss some possible approaches for finding u. This discussion is by no means exhaustive and the reader is advised to consult the literature on numerical methods for further information.

[8] Recall that for a series of 1-pole lowpasses (which $G\xi + S$ denotes in Fig. 6.12) $0 < G < 1$.

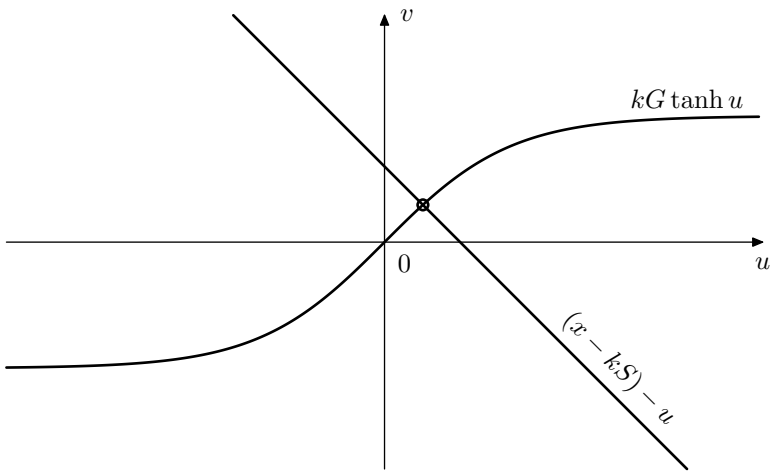


Figure 6.13: The solution of (6.13) for $k > 0$.

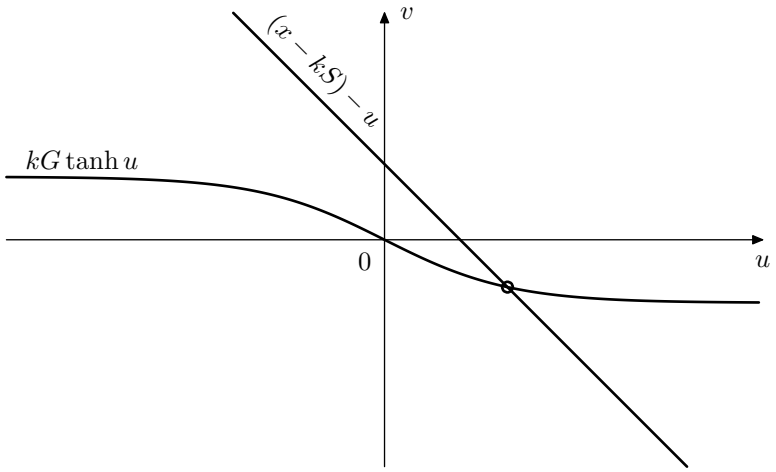


Figure 6.14: The solution of (6.13) for $-1 < kG < 0$.

6.5 Iterative methods

Fixed-point iteration

Starting with some initial value $u = u_0$ we compute iteratively the left-hand side of (6.12) from the right-hand side:

$$u_{n+1} = x - k(G \tanh u_n + S) \tag{6.14}$$

and hope that this sequence converges quickly enough.[9] Intuitively, the convergence gets worse at larger absolute magnitudes of kG, that is at high cutoffs (large G) and/or high resonance values (large k). Conversely, it gets better as

[9] In a realtime situation it would be a good idea to artificially bound the number of iterations from above.

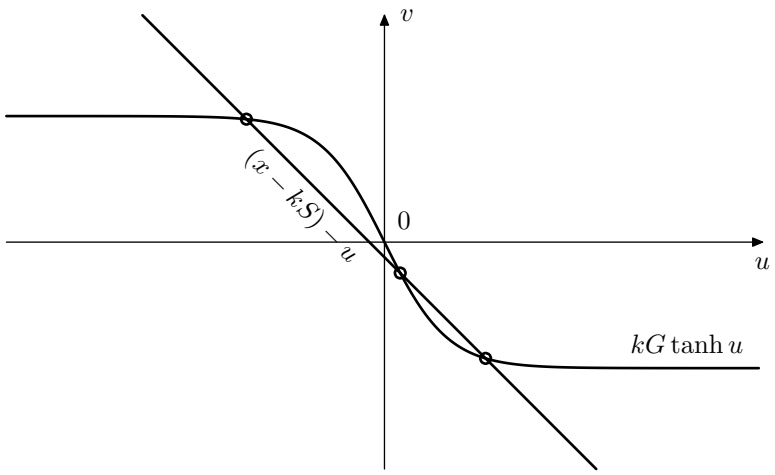


Figure 6.15: Solutions of (6.13) for $kG < -1$.

the sampling rate increases (since G becomes smaller in this case). Generally, the convergence fails for $|kG| \geq 1$.

The process defined by (6.14) has a strong similarity to the naive approach to time discretization. Indeed, for the frozen values of G and S one can treat Fig. 6.12 as stateless zero-delay feedback system (Fig. 6.16). And then we simply implement this system in the naive way by introducing a unit delay at the point of the signal u (Fig. 6.17) and letting this system run for some number of discrete-time ticks. This is a bit like oversampling of the instantaneous feedback loop part of the system.[10]

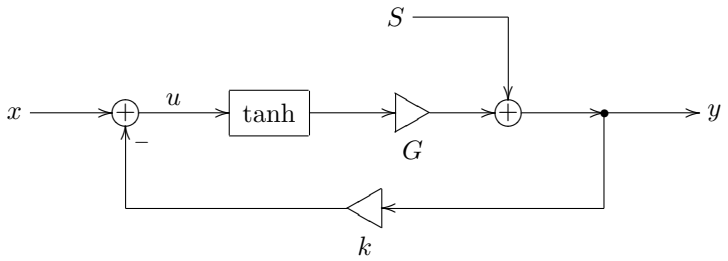


Figure 6.16: Zero-delay feedback equation (6.12) as a stateless zero-delay feedback system.

So, it is as if we introduce a "nested" discrete time into a single tick of the "main" discrete time. This suggests a natural choice of the initial value of u for (6.14), namely, taking the previous value of u (that is the value from the previous sample of the "main" discrete time) as the iteration's initial value u_0.

Under the consideration of the concept of the instantaneous smoothing (introduced in Section 3.13), the interpretation in Fig. 6.17 also suggests a way to

[10]Of course this is not exacty oversampling, because the state of the system (manfesting itself in the S variable) is frozen.

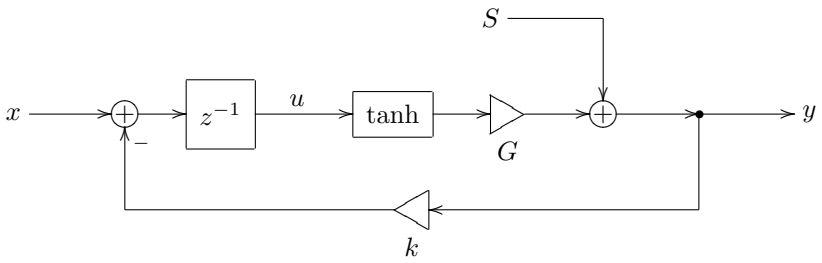


Figure 6.17: Interpretation of the equation (6.14) as an "oversampled" naive discrete-time model of the stateless feedback loop in Fig 6.16.

improve the convergence of the method by introducing a smoother into the feedback loop of Fig. 6.17. In a practical implementation such smoother can be a naive 1-pole lowpass, like in Fig. 6.18, which effectively lowers the total feedback gain from kG to a smaller value.[11] However, even though such smoother may improve the convergence at high kG, obviously it can deteriorate the convergence in good situations. Particularly at $k = 0$ the iteration process is supposed to immediately converge, however in the presence of the lowpass smoother it will converge exponentially instead.

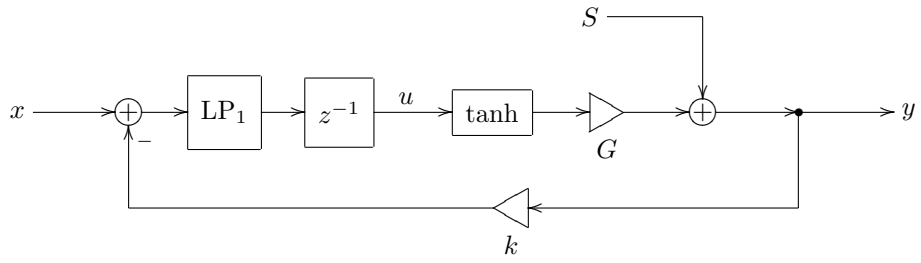


Figure 6.18: Using a 1-pole lowpass smoother to improve convergence of signals in Fig. 6.17.

Newton–Raphson iteration

A very popular approach in practical DSP is Newton–Raphson method, which is based on the idea of linearization of the function around the current point u_n by the tangent line. Instead of solving (6.13) we solve

$$(x - kS) - u_{n+1} = kG(\tanh u_n + (u_{n+1} - u_n) \tanh' u_n) \qquad (6.15)$$

for u_{n+1} to obtain the next guess and repeat the iteration (6.15) until it converges. Fig. 6.19 illustrates the idea.

[11] Clearly, the smoother will not help in the instantaneously unstable case, occurring when $kG \leq -1$.

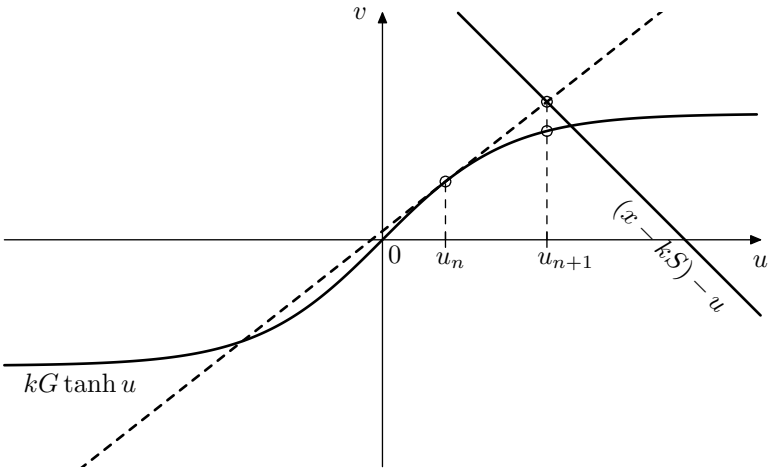


Figure 6.19: Newton–Raphson method: linearization by the tangent line.

The textbook version of Newton–Raphson method is formulated in terms of searching for a zero-crossing of a function (Fig. 6.20). By subtracting the left-hand side of (6.13) from the right-hand side we obtain the equation

$$f(u) = u + k(G \tanh u + S) - x = 0 \tag{6.16}$$

Respectively, the iterations are generated by solving

$$f(u_n) + (u_{n+1} - u_n) f'(u_n) = 0 \tag{6.17}$$

Apparently (6.15) and (6.17) (and respectively Figs. 6.19 and 6.20) are equivalent, both giving

$$\begin{aligned} u_{n+1} = u_n - \frac{f(u_n)}{f'(u_n)} = u_n - \frac{u_n + k(G \tanh u_n + S) - x}{1 + kG/\cosh^2 u_n} = \\ = u_n - \frac{u_n + k(G \tanh u_n + S) - x}{1 + kG(1 - \tanh^2 u_n)} \end{aligned}$$

Newton–Raphson method converges very nicely in almost linear areas of $f(u)$, the convergence getting worse as $f(u)$ becomes more nonlinear. As with fixed-point iteration, the convergence deteriorates at large $|kG|$, as the prediction error of u_n increases.[12]

As in the fixed-point iteration method, the value of u from the previous sample tick is a natural choice for the iteration's initial value as well. This choice usually leads to fast convergence if the new solution lies close to the value of u on the previous sample. However in excessive situations (such as high cutoff and/or high input signal frequency) the old solution could lie within the right-hand side saturation range of $\tanh u$ (that is $u \gg 0$) and the new solution could lie within the left-hand side saturation range of $\tanh u$ (that is $u \ll 0$).

[12]There are a number of tricks which can be employed to improve the convergence of Newton–Raphson, but even those might not help in all situations. The specific tricks can be found in the literature on numerical methods and fall outside the scope of this book.

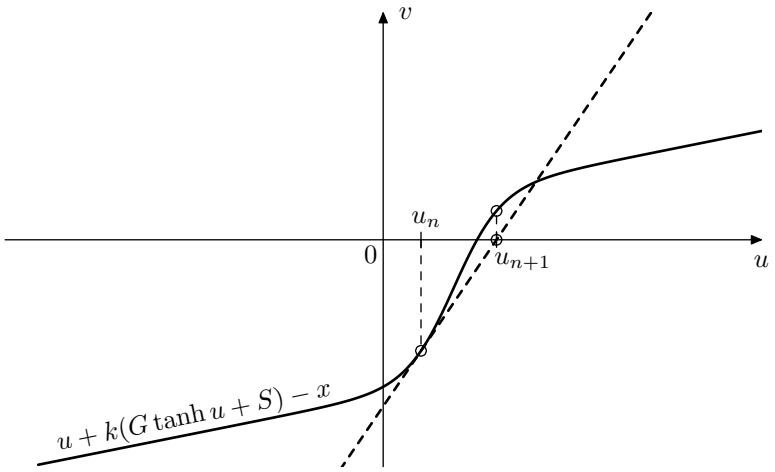


Figure 6.20: Texbook version of Newton–Raphson method (note that the aspect ratio of the graph is not 1:1)

The solution search by Newton–Raphson iterations will need to traverse both "knees" (areas of higher curvature) of tanh along the way, which usually has a negative impact on the convergence. The neutral choice of $u = 0$ as initial value might somewhat improve this worst-case scenario, while simultaneously deteriorating the convergence in "nice" situation.

Other, more advanced approached to the choice of the initial point may be used. Often one uses Newton–Raphson to refine the result of another method, so that the initial point is alredy sufficiently close to the true solution.

There is also some freedom of the choice of the variable to solve for. E.g. in Fig. 6.19 we could have been solving for v instead of u. This means that we are having

$$\begin{aligned} v &= (x - kS) - u \\ v &= kG \tanh u \end{aligned}$$

from where

$$\begin{aligned} u &= (x - kS) - v \\ u &= \tanh^{-1}(v/kG) \end{aligned}$$

and (6.13) turns into

$$(x - kS) - v = \tanh^{-1}(v/kG)$$

In this specific case v is hardly a better choice compared to u. For one we have a division by zero if $k = 0$.[13] Worse, one could see in Fig. 6.19 that v_{n+1} is located above the horizontal asymptote of $kG \tanh u$, which means that we are getting outside of the domain of $\tanh^{-1}(v/kG)$. And even if we're not outside of the domain, there still could be large precision losses when evaluating $\tanh^{-1}$ at points close to ± 1. The convergence speed is likely to be affected too. Therefore a good choice of the variable to solve for is important.

[13] Taking $\bar{v} = \tanh u$ as the unknown to solve for addresses the division by zero issue, but the other issues are similar to the choice of $v = kG \tanh u$.

Bisection

Newton–Raphson method usually converges better than fixed-point iteration, but the potential convergence problems of the former can be difficult to predict. Often there can be good ways to address the convergence issues in Newton–Raphson method, but it might be worth it to have an alternative approach, which is not suffering from such issues at all.

From Fig. 6.13 we could notice that for $k \geq 0$ we are looking for an intersection point of a monotonically decreasing straight line with a (nonstrictly) monotonically increasing curve. Therefore, if we somehow initially bracket the solution point of (6.13) we can search for it using bisection.

Given the bracketing range $u \in [a_n, b_n]$ we take the middle point $u_{n+1} = (a_n + b_n)/2$ and compare the values of the left- and right-hand sides of (6.13) at u_{n+1}. Depending on which of the two sides has a larger value, we take either $[u_{n+1}, b_n]$ or $[a_n, u_{n+1}]$ as the new bracketing range $[a_{n+1}, b_{n+1}]$. Fig. 6.21 illustrates.

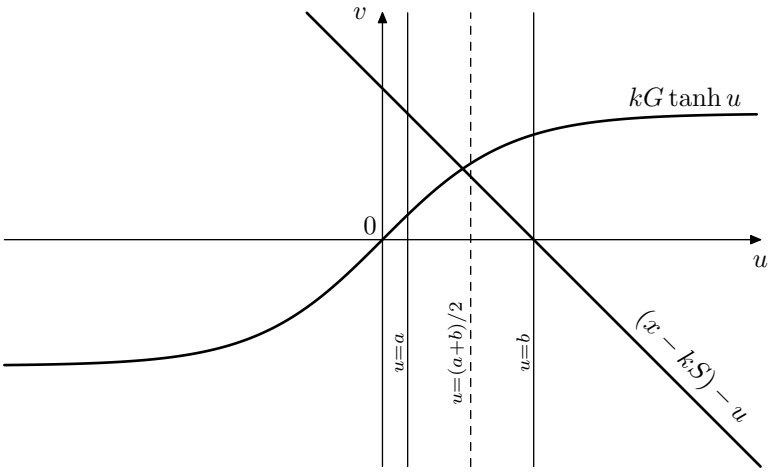


Figure 6.21: Bisection method.

Obviously the size of the bracketing range halves on each step and we repeat the procedure until the bracketing range becomes sufficiently small. The convergence speed therefore doesn't depend on values of filter's parameters or signals and the iteration is guaranteed to converge. However we need to be able to somehow find the initial bracketing range $[a_0, b_0]$.

Fortunately, with monotonic saturation shapes such as $\tanh u$ this is not very difficult. We can construct the initial bracketing range by noticing that the graph of the function $v = kG \tanh u$ lies between $v \equiv 0$ and $v = kG \operatorname{sgn} u$ (Fig. 6.22).

With unbounded saturators such as inverse hyperbolic sine one needs to get slightly more inventive. One possible idea is shown in Fig. 6.23. This however doesn't work for $k < 0$. In that case we could reuse the approach of Fig. 6.22 by taking a vertically offset version of (6.5) as a bound on $\sinh^{-1} u$ (Fig. 6.24). The intersection on $v = (x - kS) - u$ with this bound can be found by solving a quadratic equation (more on this in Section 6.7). Obviously, the same idea works for $k \geq 0$ too.

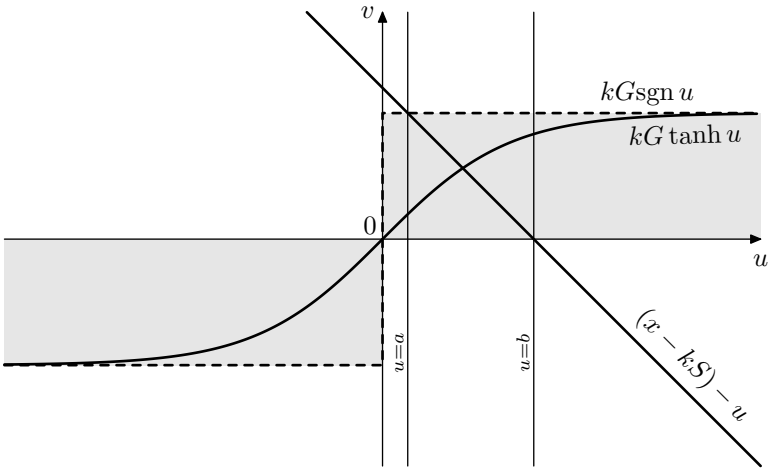


Figure 6.22: Initial bracketing for bisection.

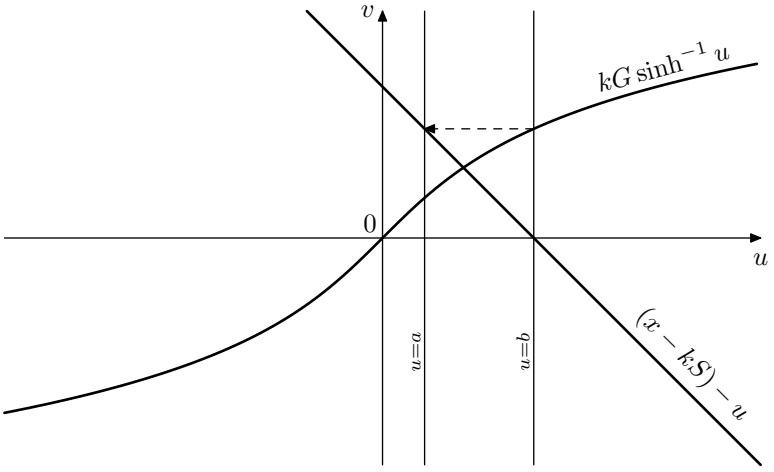


Figure 6.23: Initial bracketing for bisection in the case of an unbounded saturator and $k \geq 0$. First we find the right bracket b and then use $v = kG \sinh^{-1} b$ to find the left bracket a.

If nothing else helps to find the initial bracketing range for a (monotonic) nonlinearity $f(u)$, one could simply start at some point, such as e.g. the zero-crossing of $v = (x - kS) - u$, determine the direction of other bracket by comparing $v = (x - kS) - u$ to $kG \cdot f(u)$ and then take steps of progressively increasing size (exponential increasing of steps is usually a good idea) until the comparison result of $v = (x - kS) - u$ and $kG \cdot f(u)$ flips.

Even though bisection method guarantees convergence, the convergence speed might be a bit too low for our purposes. Let's assume that the magnitude order of the signals in the filter is 10^0 and let's assume that the length of the initial bracketing segment $[a_0, b_0]$ has about the same order of magnitude. Then, to

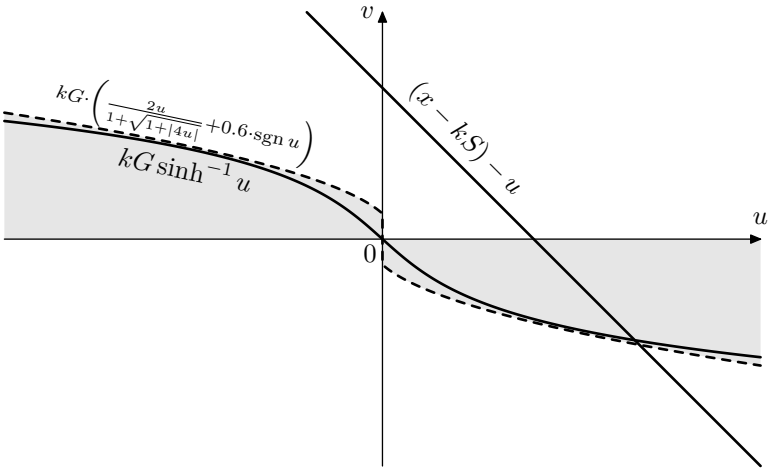


Figure 6.24: Initial bracketing for bisection in the case of $f(u) = \sinh^{-1} u$ and $k < 0$.

reach a -60dB SNR[14] corresponding to the order of magnitude of 10^{-3}, we'll need about 10 iterations. This might be a bit too expensive for a realtime audio processing algorithm on modern computers.[15]

6.6 Approximate methods

We might also attempt to find a rough approximate solution of (6.12) without running an iterative scheme. Having found u, we would simply pretend it's a true solution, and proceed as usual in the zero-delay feedback solution scheme, sending u through the tanh waveshaper and further through the 1-pole lowpasses, updating their state along the way. Several approximation approaches seem to be in (more or less) common use:

Linearization at zero. At small signal levels the nonlinearity is almost transparent:

$$\tanh u \approx u$$

Hoping that our signal level is "sufficiently small", whatever that means, we could replace $\tanh u$ by u and solve the resulting linear equation:

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

Note that this is equivalent to one step of Newton–Raphson with $u = 0$ as the initial guess.

[14] Treating the error in the numerically computed solution as noise, we can define the signal-to-noise ratio (SNR) as the ratio of the absolute magnitudes of the error and the signal, expressed in decibels.

[15] Whether this is too expensive or not depends on a number of factors. E.g. in Newton–Raphson method we needed to compute both $\tanh u$ and $\tanh' u$. With the hyperbolic tangent function we were quite fortunate in that the derivative of the function is trivially computable from the function value ($\tanh' u = 1 - \tanh^2 u$) and thus doesn't create significant computation cost. Had the derivative computation been expensive, the computation cost of 10 iterations of bisection could have been comparable to 5 iterations of Newton–Raphson.

Linearization at operating point. Hoping that the signals within the filter do not change much during one sample tick, we replace $\tanh u$ with its tangent line at the current point:

$$\tanh u \approx \tanh u_{-1} + (u - u_{-1}) \cdot \tanh' u_{-1}$$

where u_{-1} is the value of u at the previous discrete time moment. This is equivalent to one step of Newton–Raphson with u_{-1} as the initial guess. Usually this approximation provides a better result, however in the excessive (but not so unusual) situations of high cutoff, high feedback amount and/or high signal frequencies this can work worse than the linearization at zero. Thus, the linearization at zero might provide a better "worst case performance".

Linearization by secant line[16] On the graph of $\tanh u$ we draw a straight line going through the origin $(0, 0)$ and the operating point $(u_{-1}, \tanh u_{-1})$ and use this line as our linearization to obtain the value of u. Being a mixture of the previous two approaches, in moderately excessive situations this could work better than the linearization at the operating point, but at more excessive settings could work worse than the linearization at zero. The readers are however encouraged to gain their own experience and judgement in the choice of the initial guess approach.

All the above quick approximation approaches share the same idea of replacing the nonlinearity with a straight line. In that regard it is important that we have chosen to solve for the signal u at the saturator's input, so that the signal obtained through the approximation is then really sent through the nonlinearity before reaching the 1-poles and the output. One can view this as if, after having obtained the approximated result, we are doing one step of fixed-point iteration.[17] Had we instead chosen to solve for the signal at the saturator's output, the results would have been more questinable. Particularly, in the case of linearization at zero there would have been no difference to the linear case whatsoever.

The above approximation approaches work reasonably well with saturation type of nonlinearities. Obviously, the error increases as kG becomes larger and thus the system becomes "more non-linear". Notably, G, being monotonically growing in respect to $\omega_c T$, decreases as the sampling rate grows, thus the approximation error is smaller at higher sampling rates.

6.7 2nd-order saturation curves

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. $f(x) = \tanh x$ can be replaced by $f(x) = x/(1 + |x|)$ (which consists of two hyperbolic segments), thereby turning (6.13) into:

$$(x - kS) - u = kG\frac{u}{1 + |u|} \tag{6.18}$$

[16] Proposed for usage in the zero-delay feedback context by Teemu Voipio.

[17] This is a particular case of a more general idea, where we would use the result obtained by one of the above approximations as an initial point for an iterative algorithm.

The inverse of $f(x) = \sinh x$ can be replaced by the inverse of $f(x) = x(1+|x|)$, consisting of two parabolic segments.

In order to solve (6.18), which graphically is an itersection between the lines $v = (x - kS) - u$ and $v = kG \cdot f(u)$ (same as in Figs. 6.13, 6.14, 6.15), we first need to find out, whether the intersection is occuring at $u > 0$ or $u < 0$ (the case $u = 0$ can be included into either of the cases). Looking at Figs. 6.13 and 6.14, it's not difficult to realize that for $kG > -1$ this is defined solely by the sign of the value which $(x - kS) - u$ takes at $u = 0$. Thus (6.18) turns into

$$(x - kS) - u = kG\frac{u}{1+u} \qquad \text{if } x - kS \geq 0 \tag{6.19a}$$

$$(x - kS) - u = kG\frac{u}{1-u} \qquad \text{if } x - kS \leq 0 \tag{6.19b}$$

Each of the equations (6.19) is a quadratic equation in respect to u.

Choosing the appropriate one of the two solutions of the quadratic equation is easy. E.g. for (6.19a) the choice can be made with the help of Fig. 6.25. Taking into account the restriction $x - kS \geq 0$, we see that we should be alway interested in the larger of the two solutions u_1, u_2. The choice of the appropriate solution for (6.19b) can be done using similar considerations.

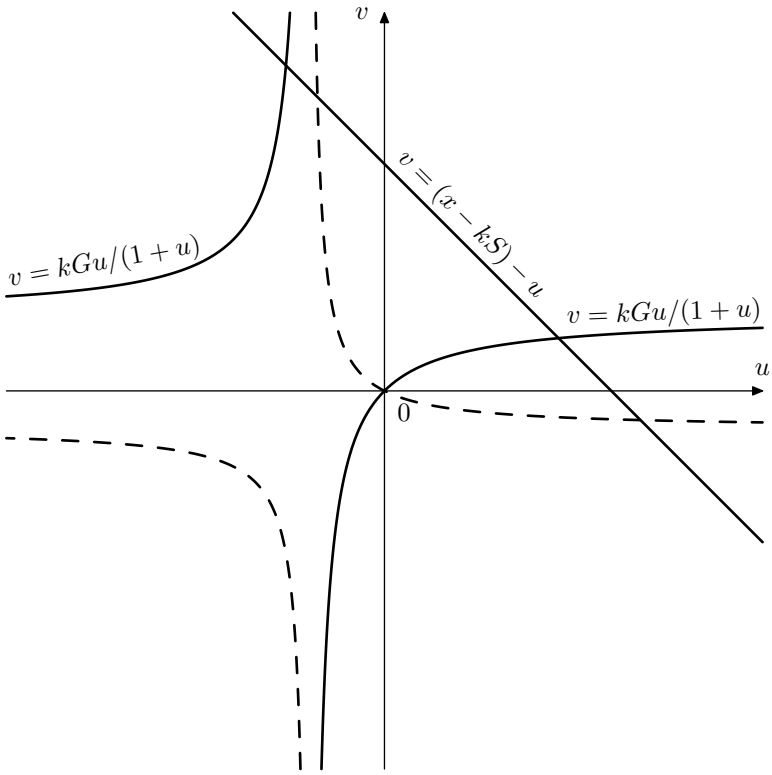


Figure 6.25: Choice of the solution of the quadratic equation for $f(u) = u/(1+u)$. The dashed line shows the graph of $v = kGu/(1+u)$ for $kG < 0$.

In solving the quadratic equation $Ax^2 - 2Bx + C = 0$ one has not only to choose the appropriate one of the two roots of the equation, but also to choose

the appropriate one of the two solution formulas:

$$x = \frac{B \pm \sqrt{B^2 - AC}}{A} = \frac{C}{B \mp \sqrt{B^2 - AC}} \tag{6.20}$$

Mathematically the two formulas are equivalent, however numerically there is a precision loss (which may become very strong) if $B \pm \sqrt{B^2 - AC}$ results in addition of two values of opposite sign, or, conversely, subtraction of two values of the same sign. This consideration yields the following formulas for the solutions of the quadratic equation:

$$x_1 = \frac{B + \operatorname{sgn} B \cdot \sqrt{B^2 - AC}}{A} \qquad x_2 = \frac{C}{B + \operatorname{sgn} B \cdot \sqrt{B^2 - AC}}$$

2nd-order soft clippers of the most general form

We could generalize the previously used idea of turning the nonlinear zero-delay feedback equation into a quadratic one by considering a waveshaper made of the most general form of a second-order curve $y = f(x)$ defined by[18]

$$\Phi(x, f(x)) = \Phi(x, y) = ax^2 - 2bxy + cy^2 - 2px - 2qy + r = 0 \tag{6.21}$$

Equation (6.21) has 6 parameters and 5 degress of freedom. After subtituting the nonlinearity (6.21) into (6.13), the equation (6.13) turns into

$$\Phi\left(u, \frac{(x - kS) - u}{kG}\right) = 0$$

or, equivalently,

$$\begin{aligned} k^2G^2au^2 - 2kGbu((x - kS) - u) + c((x - kS) - u)^2 - \\ - 2k^2G^2pu - 2kGq((x - kS) - u) + k^2G^2r = 0 \end{aligned} \tag{6.22}$$

Obviously, (6.22) is a quadratic equation in respect to u. Particularly, under the "typical soft clipping curve" conditions

$$f(0) = 0 \quad f'(0) = 1 \quad f(\infty) = 1 \quad f'(\infty) = 0 \tag{6.23}$$

equation (6.21) turns into a family of hyperbolas: with a single parameter:

$$\frac{2y_1 - 1}{y_1^2}y^2 - xy + x - y = 0 \tag{6.24}$$

Four of five freedom degrees in (6.21) has been taken by the conditions (6.23). The fifth remaining degree is represented by the parameter y_1, which is the value[19] of y that the curve has at $x = 1$ (Fig. 6.26). A reasonable choice for the range of y_1 is $[0.5, 1]$, where at $y_1 = 0.5$ we obtain the already familiar $y = x/(1 + x)$ curve, at $y_1 = 1$ the curve (6.24) turns into a hardclipper.

[18] We use the implicit form, because the explicit form has some ill-conditioning issues. Besides, in order to solve (6.13) for the specific second-order shaper function $f(x)$, we will need to effectively go from explicit to implicit form during the algebraic transformations of the resulting equation anyway, thus using the explicit form wouldn't have simplified the solution, but on the contrary, would have made it longer.

[19] More precisely, one of the two values.

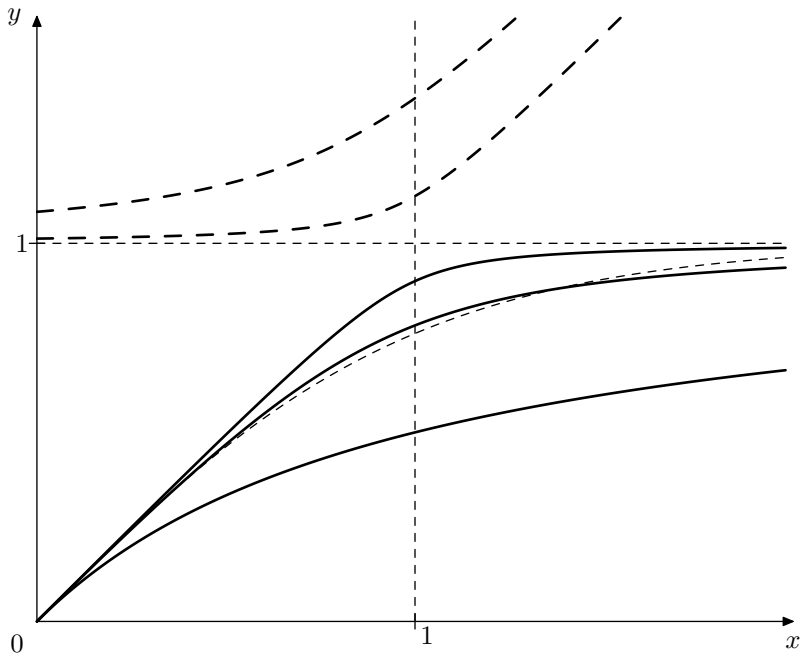


Figure 6.26: A family of soft clippers generated by (6.24) for $y_1 = 0.5$, $y_1 = 0.7829$ and $y_1 = 0.9$. The two dashed curves above the line $y = 1$ are the second (unused) branches of the respective curves (the second branch for $y_1 = 0.5$ is not visible because it is outside the picture boundaries). The thin dashed curve close to the main branch of the curve for $y_1 = 0.7829$ is the hyperbolic tangent $y = \tanh x$.

By making the odd extension of the curve:

$$f_{\text{ext}}(x) = \begin{cases} f(x) & \text{if } x \geq 0 \\ -f(-x) & \text{if } x \leq 0 \end{cases}$$

we obtain a proper soft clipping saturator shape, where we should remember to pick the appropriate branch of the curve, when solving the quadratic zero-delay feedback equation (6.22).

This time the selection of the appropriate solution of the quadratic equation is still simple for $k \geq 0$, where we can just pick the larger of the two solutions u_1, u_2, however for $k < 0$ it becomes more complicated (Fig. 6.27). From Fig. 6.27 one can see that our choice of the larger or smaller of the two solutions is switched once when kG changes sign and once again when the oblique asymptote of $kG \cdot f(u)$[20] goes at $-45°$, thereby becoming parallel to to the line $v = (x - kS) - u$.[21]

[20] It can be shown, that $f(u) \sim u \cdot \left((2y_1 - 1)/y_1^2\right)^{-1}$ at $u \to \infty$ which defines the steepness of the asymptote.

[21] In the previously discussed case $f(u) = u/(1 + u)$ we didn't have a switch between larger and smaller solutions. But $f(u) = u/(1 + u)$ is a limiting case of (6.24) at $y_1 \to 0.5$, so why is there no switch? It turns out that both switches occur simultaneously at $kG = 0$ (since

By writing out the expressions for the solutions of the resulting quadratic equation, one could see that, if we define the choice of the solution in terms of the choice of the plus or minus sign in (6.20) in front of $\sqrt{B^2 - AC}$ (which is actually what we care about), then the solution is switched only when $kG = 0$, at which moment $B^2 - AC = 0$ and respectively both solutions become equal to each other. The (negative) value of kG, at which the oblique asymptote of $kG \cdot f(u)$ goes at $-45°$, doesn't correspond to another solution switch but solely to the unused solution disappearing into the infinity from one side and reappering from the other.

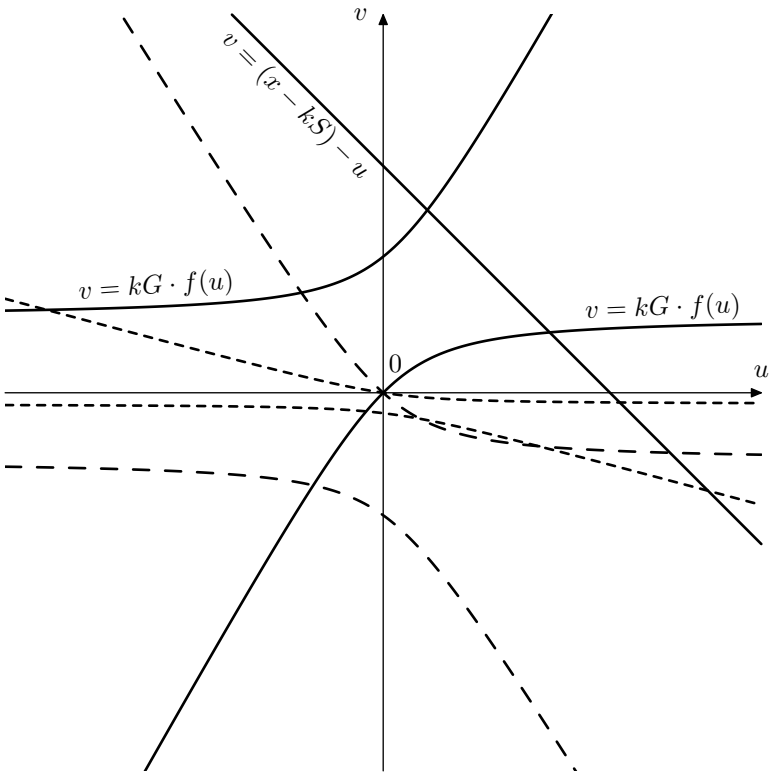


Figure 6.27: Choice of the solution of the quadratic equation for $f(u)$ which is a member of the family of hyperbolas (6.24). Solid line corresponds to $kG > 0$, dashed lines correspond to two different values of $kG < 0$.

Other 2nd-order saturators

Apparently, mixing in a linear component (6.6) into a saturator defined by (6.21) still can be expressed in the general form (6.21), thus the zero-delay feedback equation is still quadratic equation and we can use the same solution techniques.

Instead of using hyperbolas, we could also use parabolas, such as the one in (6.5) or its mixture with a linear term. Ellipses, having finite support in terms

the oblique asymptote of $f(u)$ becomes vertical), and thus we simply always choose the larger solution.

of both x and y, are not lending themselves for this kind of usage, unless used in a piecewise approximation, which we discuss later.

6.8 Tabulation

Tabulation is one of the standard ways of reducing the computation cost of functions. Instead of computing the function using some numerical method (which might be too expensive) we store function values at certain points in a lookup table. To compute the function value in between the points, interpolation (most commonly linear) is used.

Tabulation is worth a dedicated discussion in the context of nonlinear zero-delay feedback equations, because in this case it can be combined with the bisection method in a special way, making this combination more efficient. Also the same ideas provide a general framework for applying piecewise saturation curves in a zero-delay feedback context, even if the number of segments is so low that using a real table is not practical.

Imagine the saturator function in Fig. 6.13 was represented by tabulation combined with linear interpolation, which effectively means that we are having a piecewise-linear function $f(u)$ (Fig. 6.28). In order to solve (6.13) we first would need to determine the applicable segment of $f(u)$. Having found the linear segment we just need to solve a linear zero-delay equation.

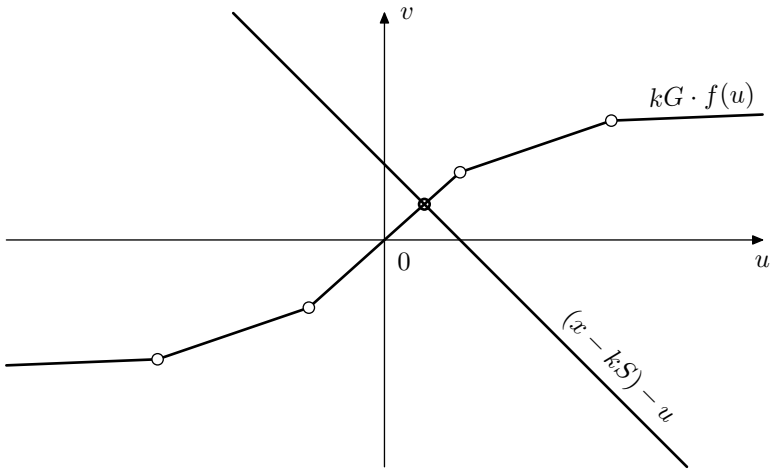


Figure 6.28: The solution of (6.13) for a piecewise-linear saturator.

From Fig. 6.28 is should be clear that the bisection method for a piecewise-linear curve can be implemented by simply comparing the values of $v = (x - kS) - u$ and $v = kG \cdot f(u)$ at the breakpoints u_n, thereby sparing the need for linear interpolation. We would start with some initial bracketing of the breakpoint range $n \in [L, R]$ and then compare the two curves at the breakpoint in the middle of the range u_M (where $M = (L + R)/2$, rounding the result of division by 2 up or down, if necessary). Depending on the comparison outcome we pick either $[L, M]$ or $[M, R]$ as the next range. We repeat until we are left with a single segment, and then simply solve the linear zero-delay feedback

equation.[22]

The very first and very last linear segments will require special care, because they do not go from one table point to the other, but extend from the outermost entries of the table to $u = \pm\infty$. We can either assume that they horizontally extend from the first and last points in the table, or store their slope separately.

As a very simple example of the just introduced concepts we could consider a hard clipper

$$f(x) = \begin{cases} 1 & \text{if } x \geq 1 \\ x & \text{if } -1 \leq x \leq 1 \\ -1 & \text{if } x \leq -1 \end{cases}$$

(Fig. 6.29). We don't need a real table to store the breakpoints, but the same ideas apply. First comparing $v = (x - kS) - u$ and $v = kG \cdot f(u)$ at $u = 1$ we find out whether the intersection occurs in the right-hand saturation segment $u \geq 1$. If not, then we perform the same comparison at $u = -1$, thereby finding out whether the intersection occurs in the left-hand saturation segment $u \leq -1$. Otherwise the interesection occurs in the middle segment $-1 \leq u \leq 1$.[23].

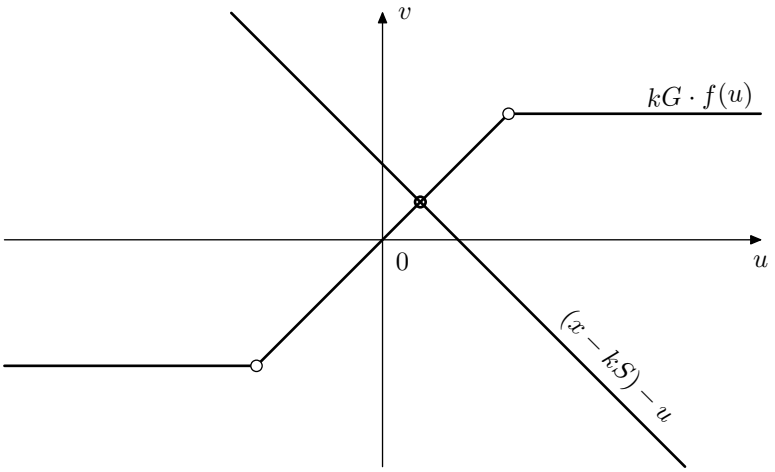


Figure 6.29: The solution of (6.13) for a hard clipper.

The tabulation approach is not limited to piecewise-linear segments. We could e.g. use the 2nd-order segments of the form (6.21). Since the latter have 5 degrees of freedom, we could use 4 of those to specify the values of the function and its first derivative at the segments ends (like we would do for a Hermitian interpolating segment and like we did for (6.24)) and use the 5th degree of freedom e.g. to minimize the remaining error. In fact, in Section 6.7 we have

[22]Note that the described binary search process doesn't rely on the regular spacing of breakpoints u_n along the u axis. This suggests that we might use an irregular spacing, e.g. placing the breakpoints more densely in the areas of higher curvature. Irregularly spaced breakpoints might complicate the initial bracketing a bit, though.

[23]Treating the hard clipper as a piecewise-linear shaper is just a demonstration example. For a hard clipper shape it might be simpler and more practical to simply perform a linearization at zero (thereby treating the hard clipper as a fully transparent shaper $f(u) = u$) to find u. As the very next step after that is sending u through the hard clipper, at the output of the hard clipper we will get the true value, as if we properly solved the equation (6.13))

done exactly this, building a piecewise-2nd-order curve consisting of two segments joined at the breakpoint at the origin. The saturator (6.2b), consisting of four segments of an order not exceeding 2, could be another candidate for this approach.

6.9 Saturation in 1-pole filters

The feedback in the 1-pole filter is not one creating the resonance. Therefore the discussion from Section 6.3 does not apply and we need to address nonlinear 1-poles separately.

We are going now to discuss nonlinear 1-poles with the nonlinearity ideas derived from different analog variations of the 4-pole lowpass ladder filter discussed in Section 5.1 These nonlinear 1-pole filters, however, are of generic nature and are therefore not limited to the usage inside 4-pole lowpass ladder filters (or inside filters of whatever specific kind, for that matter).

Transistor ladder's 1-pole lowpasses

The linear model of transistor ladder discussed in Section 5.1 (Fig. 5.1) is a first level of approximation of the behavior of the respective analog structure, where we ignore all nonlinear effects. If we wish to take nonlinear effects into account, we could replace the underlying linear 1-pole lowpasses of the ladder filter with nonlinear 1-pole lowpasses, the structure of such nonlinear lowpass being shown in Fig. 6.30. In terms of the equations, (2.3) is transformed into

$$y = y(t_0) + \int_{t_0}^{t} \omega_c \big(\tanh x(\tau) - \tanh y(\tau)\big)\, \mathrm{d}t \tag{6.25}$$

The lowpass in (6.25) and Fig. 6.30 is a simple nonlinear model of the underlying 1-pole lowpass of the transistor ladder, directly arising out of the application of Ebers–Moll transistor model.[24]

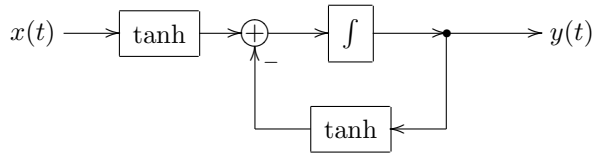


Figure 6.30: A nonlinear 1-pole lowpass element of the transistor ladder filter.

Which effect does the change from (2.3) to (6.25) have? Apparently, $\tanh x - \tanh y$ has a smaller absolute magnitude compared to $x - y$, the drop in magnitude becoming more noticeable of one or both of the signals x and y 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 more slowly than

[24] A famous piece of work describing this specific nonlinear model of the transistor ladder filter is the DAFx'04 paper *Non-linear digital implementation of the Moog ladder filter* by Antti Huovilainen. Therefore this model is sometimes referred to as the "Antti's model".

in (2.3). Intuitively this feels like "cutoff reduction" at large signal levels, or, more precisely this can be seen as audio-rate modulation of the cutoff, where the cutoff is being changed by the factor

$$K = \frac{\tanh x - \tanh y}{x - y} \qquad 0 < K \leq 1$$

where the equality $K = 1$ is attained at $x = y = 0$.

Connecting 1-poles from Fig. 6.30 in series (Fig. 6.31) can be optimized by noticing that we don't need to compute the tanh of the output of the first integrator twice (Fig. 6.32), thus sparing one tanh saturator. The entire ladder filter thereby turns into one in Fig. 6.33.

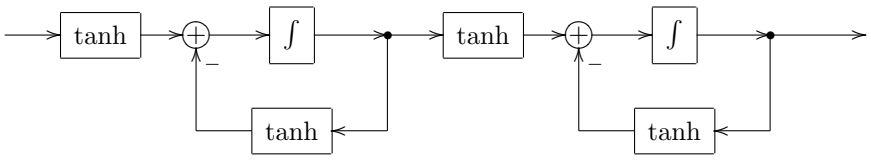


Figure 6.31: Serial connection of two nonlinear 1-pole lowpass elements from Fig. 6.30.

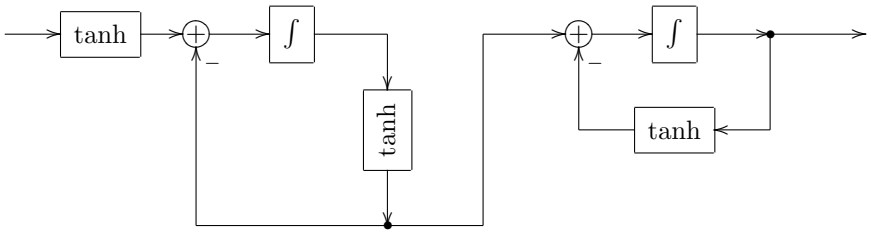


Figure 6.32: Optimized serial connection of two nonlinear 1-pole lowpass elements from Fig. 6.31.

The nonlinear 1-pole in Fig. 6.33 are normally sufficient to prevent the filter from explosion in selfoscillation range. However, obviously, there is nothing which should stop us from introducing additional nonlinearities, such as the ones discussed in Section 6.3, not so much as a means from preventing the filter explosion but rather for giving additional color to the sound. Apparently feedfoward path of Fig. 6.33 already contains many nonlinear elements, therefore adding nonlinearities to the feedback path could make more sense. Note that while there are good reasons to keep the saturation levels of nonlinearities in the feedfoward path of Fig. 6.33 (especially since we are employing the optimization from Fig. 6.32, which shares one nonlinearity between two 1-pole lowpasses), there is much less reason to have the same saturation level (or even the same saturation curve) for the nonlinearity in the main feedback path.

The nonlinear version of the diode ladder filter (Figs. 5.48, 5.49) is using a similar kind of nonlinear 1-poles, resulting in a structure shown in Fig. 6.34.

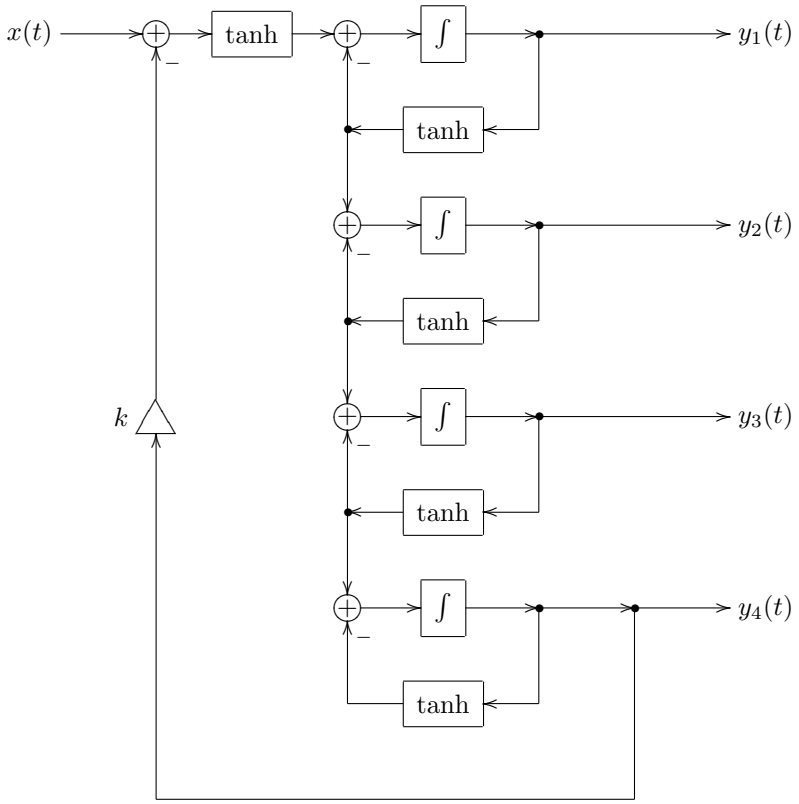


Figure 6.33: Nonlinear transistor ladder filter.

The equations (5.18) are respectively turned into:

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

(compare to (6.25)).

OTA ladder 1-poles

The same idea of the ladder filter discussed in Section 5.1 and shown in Fig. 5.1 has been often implemented in analog form using OTA (operational transconductance amplifiers) instead of transistors. This generates another kind of nonlinear 1-pole structure (Fig. 6.35).

Formally we are having a feedback loop saturator here. However this feedback loop is not responsible for generating the resonance, therefore the effect of the saturator is different from the one discussed in Section 6.3. We are having a saturator at the integrator's input, therefore we are performing soft clipping

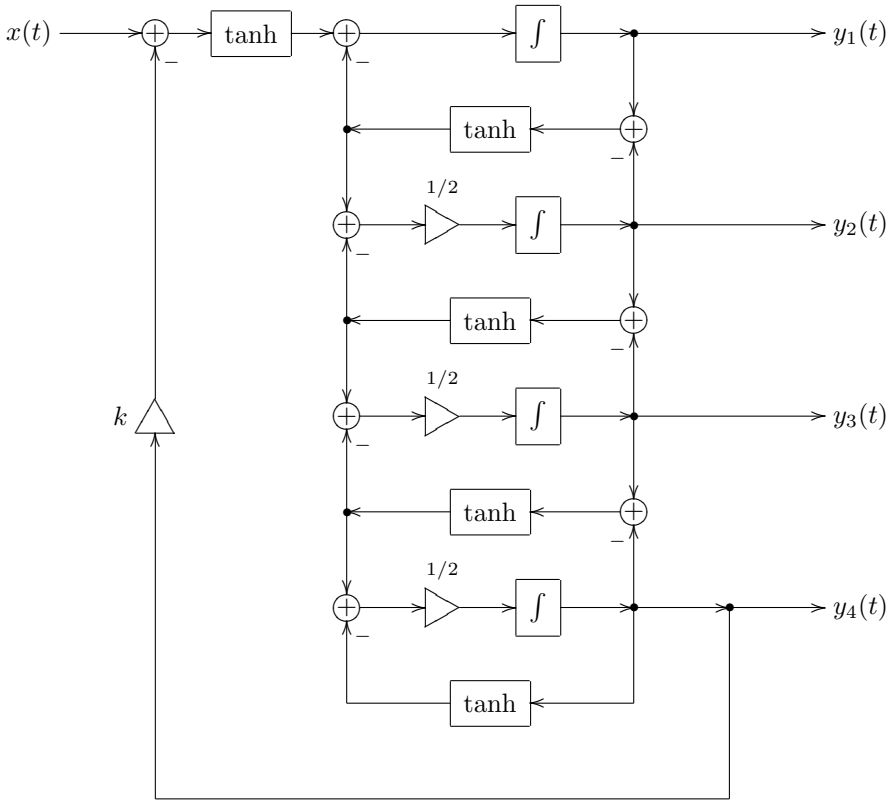


Figure 6.34: Nonlinear diode ladder filter.

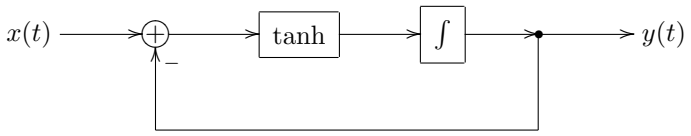


Figure 6.35: OTA-style nonlinear 1-pole lowpass.

on the speed of change of the filter's output value, or, equivalently, we are doing "soft slew limiting". Alternatively, as shown by (6.8), this can be seen as audio-rate cutoff modulation, the cutoff factor varying in agreement with (6.9).

Note that we have two different options for picking the highpass signal in Fig. 6.35. We could do this either before or after the nonlinearity. In the latter case the highpass signal will be saturated (which might be a bit over the top, compared to the lowpass signal), in the former case we have the benefit of preserving the relationship $H_{\mathrm{LP}}(s) + H_{\mathrm{HP}}(s) = 1$. This also makes the former option look like a particulary good candidate not only for a nonlinear 1-pole highpass (and thereby, among other things, for ladder filter structures utilising highpasses) but also for a nonlinear allpass. Fig. 6.36 shows the respective nonlinear 1-pole multimode.

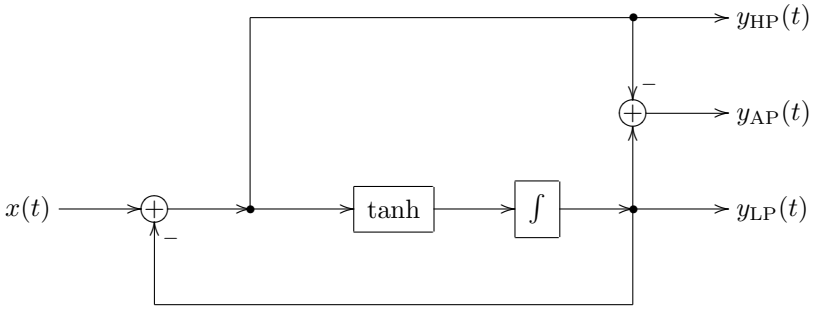


Figure 6.36: OTA-style nonlinear 1-pole multimode.

Saturated integration

The previously discussed ways of introduction of nonlinearities into 1-poles resulted in relatively complicated nonlinear behavior of the filters. But what if we want a simpler behavior? Let's say we want to simply saturate the output. Of course we simply could put a saturator at the output of the filter (Fig. 6.37) but this doesn't really feel like making the filter itself nonlinear.

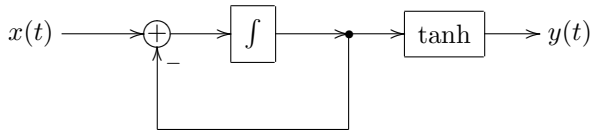


Figure 6.37: Putting a saturator at the filter's output.

We could try putting the output nonlinearity inside the filter's feedback loop (Fig. 6.38). However, comparing this to equation (2.3) we should realize that the main effect of such nonlinearity will be that the difference $x - y$ will be changed to $x - \tanh y$, leading to the capacitor in Fig. 2.1 continuing to charge even after the output value has reached the input value. In other words, the output will still grow even after reaching the input value. This feels more like a mistake.

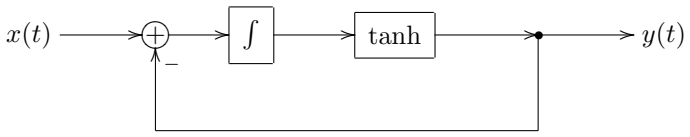


Figure 6.38: Saturating the integrator's output (not really a working idea).

What we rather want is to prevent the 1-pole's capacitor in Fig. 2.1 from charging beyond a certain level (that is we want to prevent the integrator state from going beyond a certain maximum). In order to achieve that in a "proper analog way", we will need to introduce *antisaturators*, which we are going to

do later in this chapter. However we could also do a "hack" and modify the integrator structure, introducing the saturation into its internal accumulation process. This works particularly well with direct form I (Fig. 6.39) and transposed direct form II (Fig. 6.40) integrators. Obviously, this hack is not limited to 1-poles, but can be applied to any structure which is based on integrators, such as e.g. SVF.

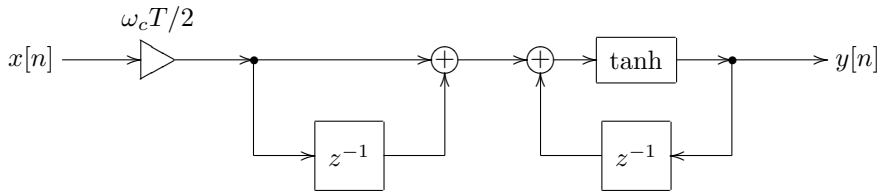


Figure 6.39: Saturating direct form I trapezoidal integrator.

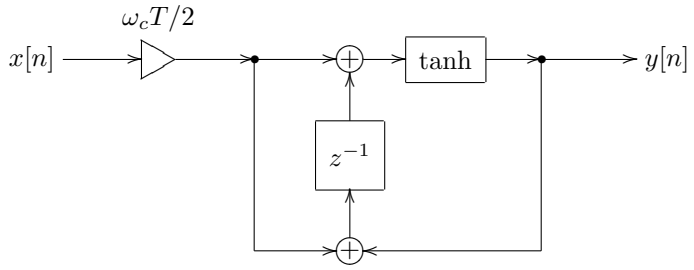


Figure 6.40: Saturating transposed direct form II trapezoidal integrator.

6.10 Multinonlinear feedback

We have seen that instantaneous responses of linear filters are linear functions of their input, such as e.g. in (3.29). It is not difficult to realize, particularly from the previous discussion of the solution of the nonlinear zero-delay feedback equation (6.12), that instantaneous response of a nonlinear filter is some nonlinear function of its input:

$$y = F(x, S) \tag{6.26}$$

(where we also explicitly notated the dependency on the filter's state S, but the dependency of F on the filter's parameters is understood implicitly).

Consider the OTA-style 1-pole lowpass in Fig. 6.35 and imagine we build a 4-pole lowpass ladder filter (as in Fig. 5.1) from four idenitical 1-pole lowpasses of this kind. Assuming (6.26) decribes the instantaneous response of Fig. 6.35, we could redraw Fig. 5.1 in the instantaneous response form as Fig. 6.41.

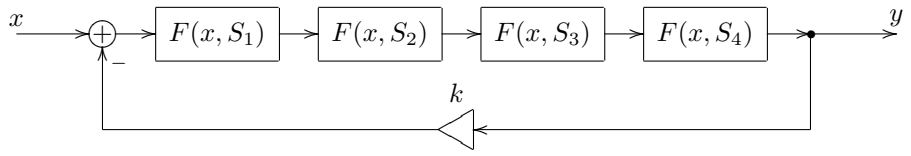


Figure 6.41: Nonlinear ladder filter in the instantaneous response form.

Let u denote the signal at the input of the first 1-pole lowpass in Fig. 6.41. The zero-delay feedback equation for the entire of Fig. 6.41 therefore becomes

$$u = x - k \cdot F(F(F(F(u, S_1), S_2), S_3), S_4) \tag{6.27}$$

Intuitively we can expect $F(x, S)$ to be monotonically increasing with respect to x, thus $F(F(F(F(x, S_1), S_2), S_3), S_4)$ should be monotonically increasing too, and we could use most of the previously described methods of solving nonlinear zero-delay feedback equations to solve (6.27). Theoretically.

Practically the evaluation of $F(x, S)$ is usually very expensive, because it means a numerical solution of the zero-delay feedback equation for the respective 1-pole, possibly running several rounds of an iterative method. Now, if we are going to use an iterative method to solve (6.27), these expenses will be multiplied by the number of the "outer" iterations. Besides, if we are using Newton–Raphson to solve (6.27) then we need not only to evaluate $F(x, S)$ but also its derivative with respect to x, which further increases the computation cost of solving (6.27).

Therefore usually such "nesting" approach, where we express the higher-level zero-delay feedback equation in terms of the solutions of the lower-level zero-delay feedback equations, is not very practical for nonlinear systems. Instead, let's "flatten" the entire structure, and write the equation describing the instantaneous response signals within this structure. E.g. for the 4-pole ladder built out of 1-poles in Fig. 6.35 the flattened structure is shown in Fig. 6.42. Or, representing the integrators by their instantaneous responses (which are fully linear), we obtain Fig. 6.43.

Denoting the input and output signals of each of the 1-poles as x_n and y_n, we write the 1-pole zero-delay feedback equations:

$$y_n = g \tanh(x_n - y_n) + s_n$$

Or, since $x_{n+1} = y_n$ we can denote the input of the first lowpass as y_0 and write

$$y_n = g \tanh(y_{n-1} - y_n) + s_n \qquad n = 1, \ldots, 4$$

Plus, we are having the global feedback loop:

$$y_0 = x - k y_4$$

and thus we are having an equation system:

$$y_0 = x - k y_4$$

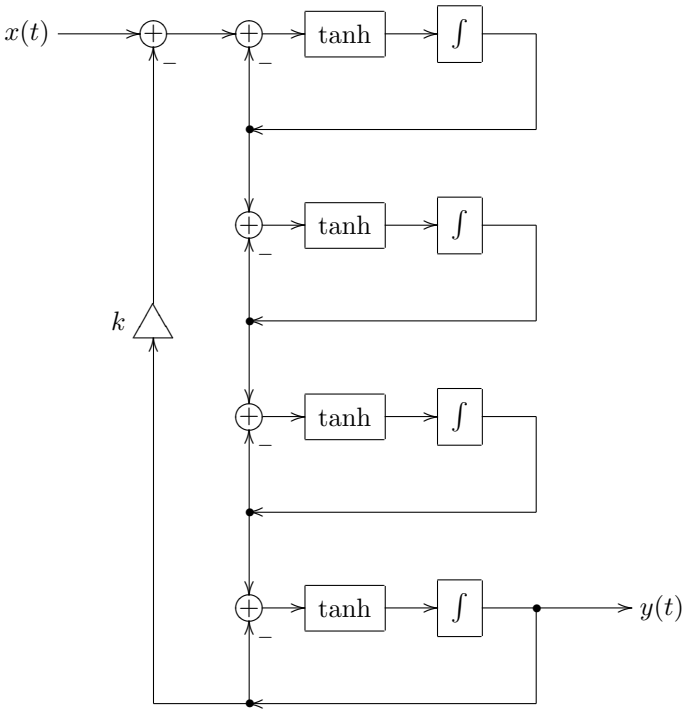


Figure 6.42: Flattened OTA lowpass ladder filter structure.

$$\begin{aligned} y_1 &= g\tanh(y_0 - y_1) + s_1 \\ y_2 &= g\tanh(y_1 - y_2) + s_2 \\ y_3 &= g\tanh(y_2 - y_3) + s_3 \\ y_4 &= g\tanh(y_3 - y_4) + s_4 \end{aligned}$$

We can get rid of the first equation by simply substituting its right-hand side for y_0, obtaining:

$$\begin{aligned} y_1 &= g\tanh(x - ky_4 - y_1) + s_1 \\ y_2 &= g\tanh(y_1 - y_2) + s_2 \\ y_3 &= g\tanh(y_2 - y_3) + s_3 \\ y_4 &= g\tanh(y_3 - y_4) + s_4 \end{aligned} \tag{6.28}$$

Equation (6.28) can be written in a more concise form by introducing the vector

$$\mathbf{y} = \begin{pmatrix} y_1 & y_2 & y_3 & y_4 \end{pmatrix}^\mathsf{T}$$

and the vector-function of a vector argument $\mathbf{\Phi}$:

$$\mathbf{\Phi}(\mathbf{y}) = \begin{pmatrix} g\tanh(x - ky_4 - y_1) + s_1 \\ g\tanh(y_1 - y_2) + s_2 \\ g\tanh(y_2 - y_3) + s_3 \\ g\tanh(y_3 - y_4) + s_4 \end{pmatrix}$$

In this notation (6.28) looks simply like

$$\mathbf{y} = \mathbf{\Phi}(\mathbf{y}) \tag{6.29}$$

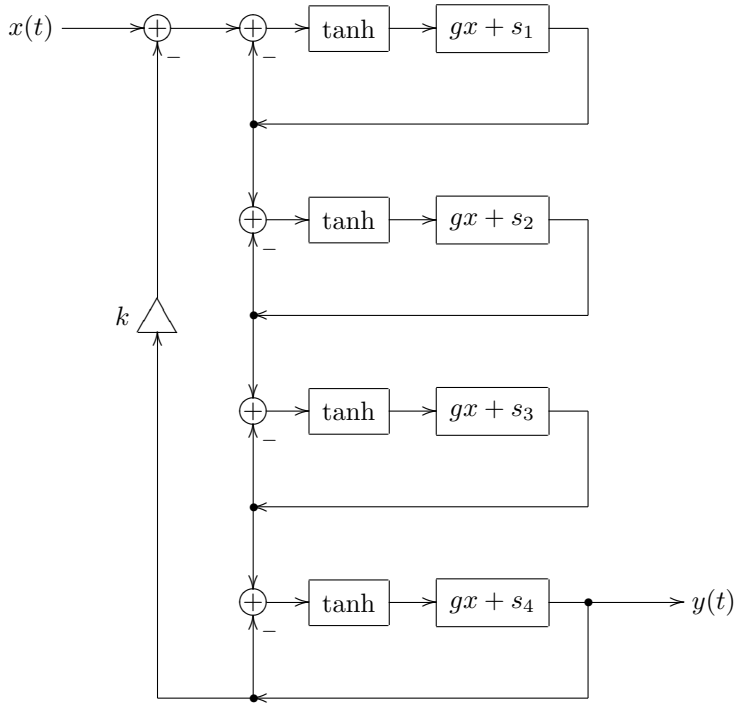


Figure 6.43: Flattened OTA lowpass ladder filter structure in the instantaneous response form.

This is our nonlinear 4-dimensional (since we are having 4 unknowns y_n) zero-delay feedback equation.

The form (6.29) readily offers itself for fixed-point iteration. By rewriting (6.29) as

$$\mathbf{\Phi}(\mathbf{y}) - \mathbf{y} = 0$$

the multidimensional form of Newton–Raphson algorithm can be used:

$$\mathbf{y}_{n+1} = \mathbf{y}_n - \left(\frac{\partial(\Phi(\mathbf{y}) - \mathbf{y})}{\partial \mathbf{y}} (\mathbf{y}_n) \right)^{-1} \cdot (\Phi(\mathbf{y}_n) - \mathbf{y}_n)$$

Also the quick approximate methods of Section 6.6 work out of the box.

The difference of solving (6.29) instead of (6.27) is that in (6.29) we are simultaneously solving all zero-delay feedback equations in the system, thereby not having the problem of nested iterations.

Actually, choosing the 1-pole output signals as the unknowns is not necessarily the best choice. It would have been more convenient to solve for the inputs of the integrators, so that we can directly reuse the obtained signals to update the integrator states.[25] On the other hand, e.g. for the transposed direct form II integrator (Fig. 3.11) one could deduce the new state from the old state and the new output signal, thus y_n also work pretty efficiently (this trick has been used in the digital implementation of an SVF in Section 4.4). A consideration

[25]It might be a good idea to write the equation system in terms of integrator input signals as an exercise.

of a bigger importance therefore could be that the choice of the unknowns may affect the convergence of the iteration scheme.

Usually for multidimensional zero-delay feedback cases the iterative methods need to be further refined and/or a combination of different methods need to be used to have a reliable and quick convergence of an iterative process of finding the solution of (6.29). However, often simply using the approximate methods of Section 6.6, will deliver reasonable results.

6.11 Antisaturators

In Section 6.9 we made some attempts to make the 1-pole lowpass filter state saturate, the most successful attempt being the modification of the internals of an integrator. In a real analog circuit we wouldn't have been able to do the same, as e.g. a capacitor, which is used as an integrator for the current, doesn't have "built-in saturation functionality". Therefore different means have to be used to achieve the integrator state's saturation.

Diode clipper

A common trick is to shorten the 1-pole filter's capacitor with a nonlinear resistance, this resistance being high at low voltages and dropping down at high voltages on the capacitor. That is the short path is disabled at low voltages but progressively "turns on" at higher voltages. This can be done by using a diode pair (Fig. 6.44). The structure in Fig. 6.44 is commonly referred to as *diode clipper*.

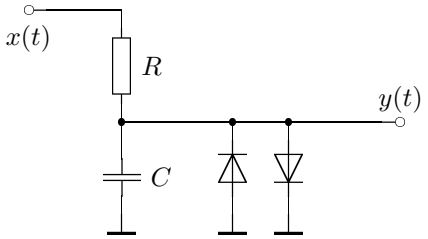


Figure 6.44: Diode clipper.

Using Shockley diode equation we can show that, qualitatively, the current flowing through the diode pair is related to the capacitor voltage as

$$I_D = I_s \sinh \frac{U_C}{U_T}$$

where I_s and U_T are diode parameters (Fig. 6.45 provides a graph of sinh as a reference). This current is then subtracted from the current which is charging the capacitor, thus acting as current leakage:

$$\dot{q}_C = I - I_D = I - I_s \sinh \frac{U_C}{U_T}$$

(please refer to equations (2.1) for the other details of the circuit's model). Since I_s is very small, as long as U_C is below or comparable to U_T the leakage

is negligible. As U_C exceeds U_T, the current grows exponentially and quickly stops being negligible.

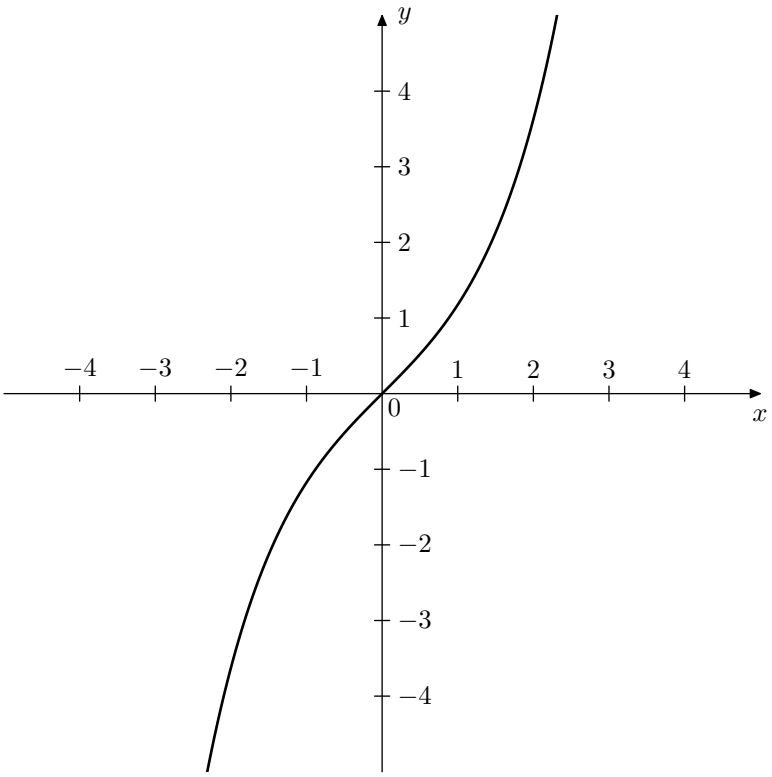


Figure 6.45: Hyperbolic sine $y = \sinh x$.

In terms of the block diagram (Fig. 2.2) this current leakage can be expressed as shown in Fig. 6.46, where we have assumed that the filter cutoff is controlled by the resitance R rather than capacitance C and thus the amount of the current leakage is independent of the cutoff.

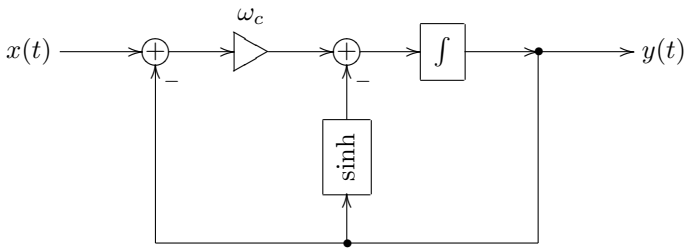


Figure 6.46: Diode clipper in the form of a block diagram. "sinh" stands for some curve of the form "$a \sinh(x/b)$".

The fact that the leakage current is independent of the cutoff is actually having the opposite effect: the effects of the leakage become cutoff-dependent and the leakage more strongly affects the filter at lower cutoffs. Particularly,

given a constant input voltage, the stabilized output level will be larger at larger cutoffs. For the purposes of generic application it is therefore more useful to make the leakage cutoff-independent, as in Fig. 6.47.

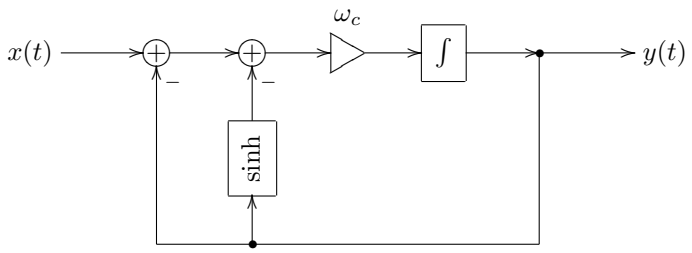


Figure 6.47: Diode clipper with cutoff-independent leakage. "sinh" stands for some curve of the form "$a\sinh(x/b)$".

Or, using implied cutoff notation and combining the two feedback paths into a single one, we obtain the structure Fig. 6.48. Also, in Fig. 6.47 the cutoff parameter ω_c was not the true cutoff of the system, since at low signal levels the gain of the feedback path was $1 + a/b$. This made the system behave as if its cutoff was $(1 + a/b)\omega_c$ and as if its input signal was reduced by $(1 + a/b)$ factor at the same time. In Fig. 6.48 we addressed this issue by scaling the linear path of the feedback by the factor $(1 - a/b)$. This doesn't change the qualitative behavior of the system, but affects only the interpretation of the cutoff ω_c and the input signal scale.

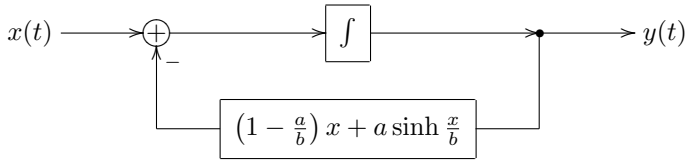


Figure 6.48: Diode clipper with cutoff-independent leakage (simplified diagram).

The structure in Fig. 6.48 is a good illustration of the idea that we could employ to introduce saturation into 1-pole lowpass filter's state: as $\sinh(x/b)$ grows exponentially for large signals, the term $a\sinh(x/b)$ causes the negative feedback to grow as well, thereby causing the integrator to "discharge".

The same effect is obviously obtained by putting any other quickly growing function of a similar shape into the feedback path of a 1-pole lowpass. Good options for such functions are provided by the inverses of the saturator functions introduced in Section 6.2:

$$y = \tanh^{-1} x = \frac{1}{2} \ln \frac{1+x}{1-x} \qquad \text{(inverse of (6.1))}$$

$$y = x/(1 - |x|) \qquad \text{(inverse of (6.2c))}$$

$$y = \sinh x \qquad \text{(inverse of (6.4))}$$

$$y = x(1 + |x|) \qquad \text{(inverse of (6.5))}$$

A particularly important feature of the inverses of the saturators is that, same as with saturators, they are transparent at low signal levels, thereby not affecting the cutoff of the filter.

We will refer to the waveshapers having an inverse saturator kind of shape as *antisaturators*. Fig. 6.49 shows another version of Fig. 6.48, this time using a simpler antisaturator.

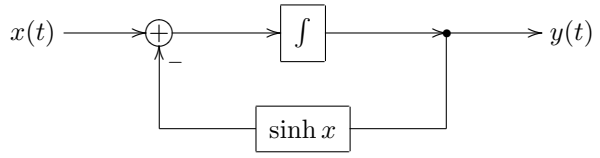


Figure 6.49: Lowpass filter's state saturation by using an antisaturator.

An antisaturator in Fig. 6.49 is having a similar effect on the filter's state saturation as its inverse (the respective saturator) would have had if directly applied to a signal, or if being put in a resonating feedback path. Specifically, using an unbounded saturator's inverse as an antisaturator in Fig. 6.49 would result in an unbounded saturation of the filter's state, in the sense that by making the amplitude of the input signal of the filter larger and larger one can achieve arbitrarily large levels of the filter's state. On the other hand, using a bounded saturator's inverse as an antisaturator (such as e.g. $\tanh^{-1}$) would result in bounding of the filter state, the state not being able to exceed the saturation level.

As with saturators, adding a linear term to an antisaturator $f(x)$ doesn't change its antisaturating behavior, but simply weakens it a bit further, where we assume that the addition should be done under the same considerations of keeping the transparency at low signal levels:

$$y = (1 - \alpha)f(x) + \alpha x \qquad (0 < \alpha < 1)$$

The antisaturator in Fig. 6.48 is a kind of a reverse example of this principle, which can be seen as if the (otherwise fully linear and transparent) shape $y = x$ was modified by an addition of a non-transparent antisaturator $a\sinh(x/b)$, however the resulting curve has been made transparent again.

Antisaturation in SVF

As with 1-pole filters, the feedback in SVF is also not one creating the resonance, respectively the discussion from Section 6.3 does not apply either, and thus we can't simply put a saturator into the feedback loop. Actually, the purpose of the feedback in SVF is kind of an opposite of creating the resonance. The function of the feedback path containing the bandpass signal is to dampen the otherwise self-oscillating structure. This suggests the idea that if we put an antisaturator into the bandpass signal path, this might actually do the trick of preventing the signal levels from getting too high.

Our first attempt to do so is shown in Fig. 6.50. After thinking a bit we, however, realize that it can't work. Indeed, at $R = 0$ there is no damping signal whatsoever, the same as without the antisaturator. Furthermore, probably the

main reason to introduce the antisaturator into the SVF is so that we could go into the selfoscillation range $R < 0$, same as we did e.g. with nonlinear 4-pole ladder by going into the range $k > 4$. However, at $R < 0$ the introduced antisaturator doesn't cause any damping either, quite on the opposite, it amplifies the "antidamping" (the inverted damping signal). Obviously, putting the antisaturator after the $2R$ gain element instead of putting it before doesn't change much in this regard.

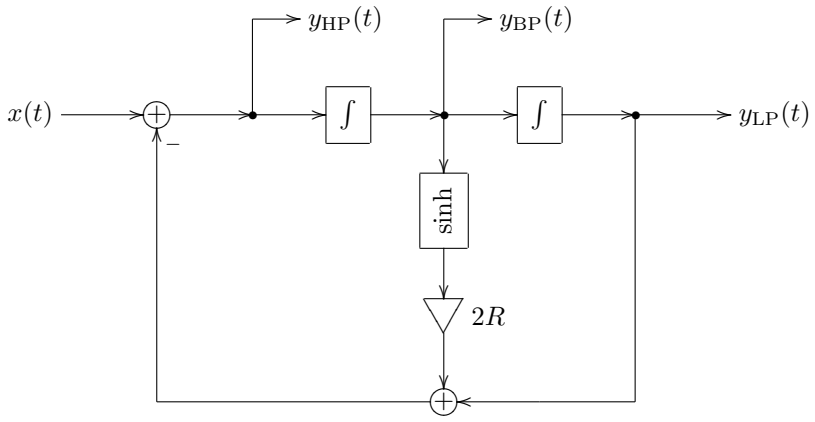


Figure 6.50: An attempt to introduce an antisaturator into an SVF (not really working).

We could get a bit smarter and connect a saturator in parallel with the $2R$ gain element (Fig. 6.51). This now does the job of saturating the signals, as the damping feedback signal will grow exponentially at large levels of y_{BP}, no matter what the value of R is. However now the effective gain of the damping feedback path (at low signal levels, where $\sinh x \approx x$) is $2R+1$, rather than $2R$.

The latter problem is fixed in Fig. 6.52. In this structure, at the neutral setting of $R = 1$ the entire damping signal goes through the antisaturator. This exactly matches the same situation in our first attempt in Fig. 6.50 (and is the reason for the separation of the multiplication by 2 into an additional gain element). As R gets away from 1, we send some of the damping signal through the parallel linear path, still keeping the total gain of the damping path equal to $2R$ at low signal levels.

The antisaturator in Fig. 6.52 effectively makes the state of the first integrator saturate. This might result in the feeling that the level of the bandpass signal y_{BP} becomes too low. Therefore, instead one could pick the bandpass signal from $y_{\mathrm{BP}'}$ output, where the antisaturator has increased the level of y_{BP} back. The y_{BP1} output provides the normalized bandpass signal.

Note that $y_{\mathrm{HP}} + y_{\mathrm{BP1}} + y_{\mathrm{LP}} = x$, as for the linear SVF.

Zero-delay feedback equation with antisaturators

The introduction of antisaturators raises some new considerations for the solution of the zero-delay feedback equation. We will use the nonlinear 1-pole in

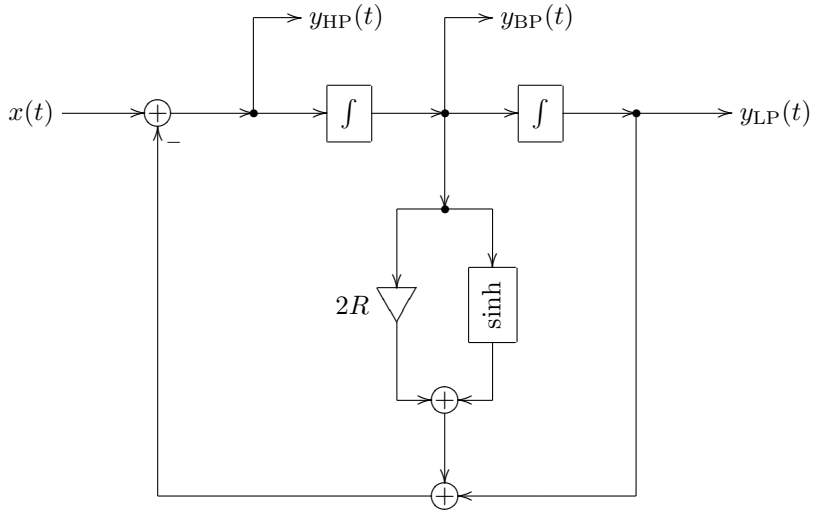


Figure 6.51: A second attempt to introduce an antisaturator into an SVF (works better, but R does no longer directly correspond to damping).

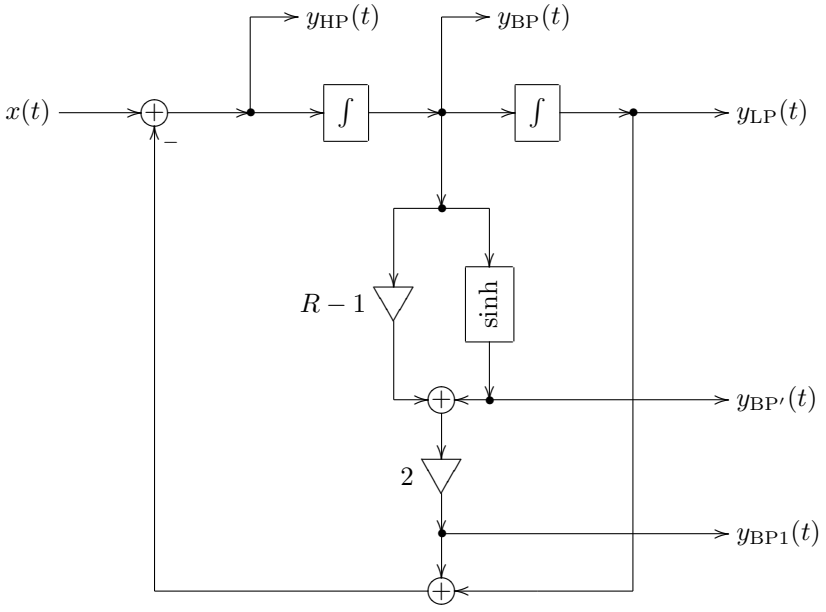


Figure 6.52: An SVF with antisaturator.

Fig. 6.49 as a demonstration example, however it will also be more instructive to consider an inverse hyperbolic tangent (Fig. 6.53) instead of a hyperbolic sine

as an antisaturator.

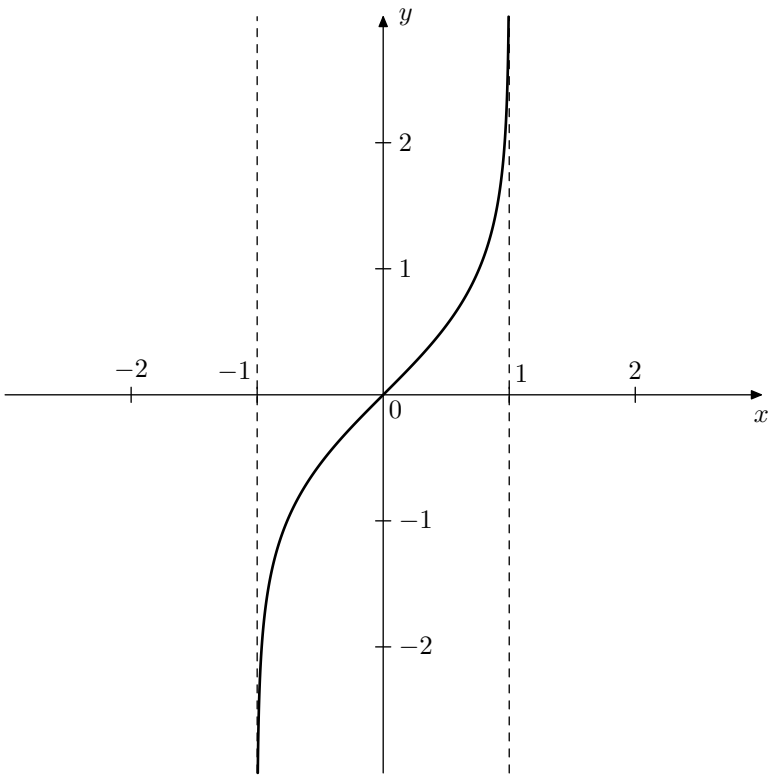


Figure 6.53: Inverse hyperbolic tangent $y = \tanh^{-1} x$.

Introducing the instantaneous response $gx + s$ for the integrator in Fig. 6.49 and replacing sinh with $\tanh^{-1}$ we obtain Fig. 6.54. Writing the zero-delay feedback equation for Fig. 6.54 we obtain

$$y = g(x - \tanh^{-1} y) + s \tag{6.30}$$

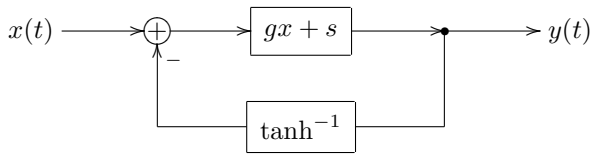


Figure 6.54: Lowpass filter with a $\tanh^{-1}$ antisaturator in the instantaneous response form.

We could start solving (6.30) using the usual methods, such as the ones discussed earlier in this chapter, however notice that $\tanh^{-1}$ has a limited support, being defined only on the $(-1, 1)$ range. This might create serious problems if we somehow arrive at a value of y outside of that range. Such values of y could appear for a number of reasons, such as e.g.:

- from an approximate solution
- from an iterative method's step
- from numerical errors, such as roundoffs.[26]

Even if we formally stay within the range $y \in (-1, 1)$, we could still get out of the range of representable values of $\tanh^{-1} y$ if $\tanh^{-1} y$ gets too large.

There are also related questions of convergence of iterative schemes, particulary of fixed point iteration. Last but not least, close to the boundaries of the range $y \in (-1, 1)$ a small numerical error in the value of y will result in a huge error in the value of $\tanh^{-1} y$, which suggests that *it might be generally a bad idea to explicitly evaluate* $\tanh^{-1} y$ *at all.* Similar issues also of course arise with unbounded antisaturators, even though they are not as bad as with bounded ones.

In order to avoid this kind of problems, we can solve for the antisaturator's output, rather than for the antisaturator's input. Introducing variable u for the antisaturator's output signal:

$$u = \tanh^{-1} y$$

we respectively have $y = \tanh u$ and can rewrite (6.30) in terms of u as

$$\tanh u = g(x - u) + s$$

or, further rewriting it so that the linear function in the right-hand side is more explicitly visible

$$\tanh u = (gx + s) - gu \tag{6.31}$$

Equation (6.31) looks very much like the previously discussed zero-delay feedback equation (6.13). However, there are still important differences. Expressing the left- and right-hand sides of (6.31) graphically in Figs. 6.55 and 6.56, we see that, compared to Figs. 6.13, 6.14 and 6.15, multiple solutions can occur already for $g < 0$. Fortunately, in Fig. 6.54 the value of g cannot get negative, since that would require a negative cutoff value for the integrator.

Having found u from (6.31) we can "send" it further through the feedback loop, first finding the integrator's input value as $x - u$, then updating the integrator's state and finding y as the output value of the integrator. Note that thereby we *never* explicitly evaluated $\tanh^{-1} y$.

For an antisaturator in the SVF (Fig. 6.52) the situation is more complicated. We would like to solve for the antisaturator's output $y_{\mathrm{BP}'}$, but then we would be stuck immediately afterwards: since we don't know the signal on the "$R-1$" path, we can't add the output signals from sinh and $R - 1$. Furthermore, we would have a similar problem of not knowing y_{LP} at the next adder (which computes $y_{\mathrm{BP1}} + y_{\mathrm{LP}}$). These problems are not unexpected, considering that we have been solving for a point in the signal path which is not shared among all zero-delay feedback loops in the structure.

One way around this would be to try to introduce more unknowns into the system and solve several equations at once. However, in this specific case we

[26] Going out of supported range of y due to numerical errors is more likely to happen in more complicated structures than the one in Fig. 6.54. However since we are using Fig. 6.54 as a demonstration of general principles, we should mention this aspect as well.

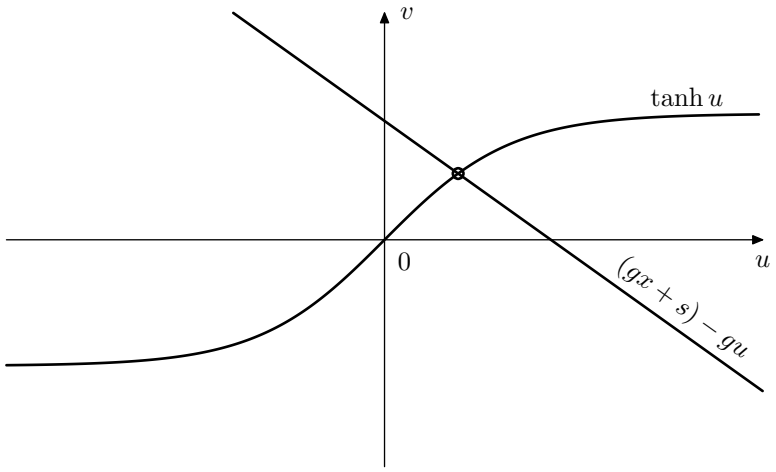


Figure 6.55: The solution of (6.31) for $g > 0$.

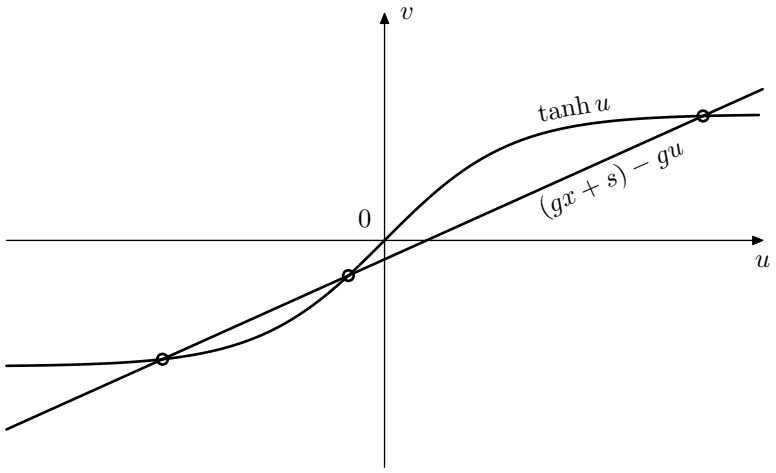


Figure 6.56: The solution of (6.31) for $g < 0$.

could simply "send the obtained signal through the antisaturator in the reverse direction". That is, knowing the antisaturator's output, we can obtain the antisaturator's input by evaluating $\sinh^{-1}$ (which is completely okay, we don't want to explicitly evaluate the antisaturator function because it can increase the computation error by a huge factor, but it is no problem to evaluate its inverse), thereby finding the value of y_{BP}. The signal y_{BP} is shared among all zero-delay feedback loops and therefore is sufficient to find all other signals in the structure.[27]

The general approach of avoiding the explicit evaluation of antisaturators but rather dealing with their inverses instead also allows us to deal with a

[27] Of course, we should remember that we already know the output signal of the antisaturator and not attempt to evaluate it again as $\sinh y_{\mathrm{BP}}$, which was the whole point of solving for $y_{\mathrm{BP}'}$.

certain class of antisaturators which are not functions in the normal sense. An example of this are compact-range monotonic saturators such as (6.2b). The inverse of such saturator is not really a function, since it would have infinitely many different values at $x = \pm 1$ (Fig. 6.57). However we still can use it as an antisaturator, since we never have to deal with the antisaturating function explicitly, but are dealing with the respective saturating function instead.[28]

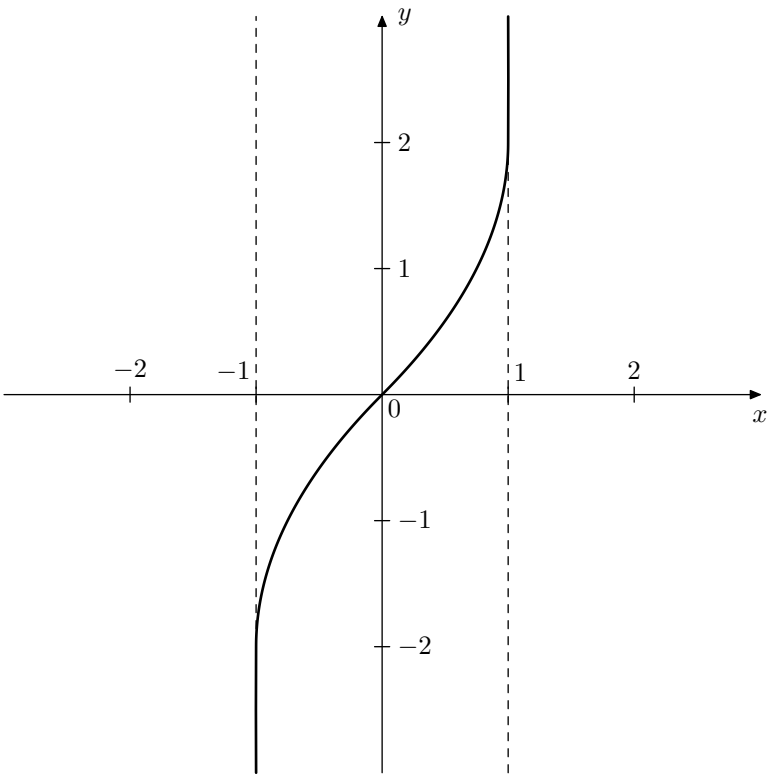


Figure 6.57: The inverse of (6.2b) is not a function in the normal sense.

6.12 Asymmetric saturation

The saturators which we have been using so far were all having the odd symmetry $f(-x) = -f(x)$. A feature of all symmetric saturators is that when its input signal amplitude is very high, the output signal basically alternates between positive and negative saturation levels $f(x)$. If the input signal is something like a sine or a sawtooth, the saturator would produce a square-like output. More generally, such saturators tend to produce signals containing mostly odd harmonics (as the square wave does).

Sometimes this domination of odd harmonics can become too boring[29] and

[28] Note that thereby we can even use an antisaturator which is an inverse of hard clipper.

[29] More likely so for a "standalone" saturator being used as an overdrive effect, rather than for a saturator used in a complicated feedback loop structure in a filter.

asymmetric saturation might be desired. Simply adding an offset to the saturator's input (or, instead, performing a parallel translation of the saturator curve by "sliding" it through the origin, to keep the property $f(0) = 0$) works only for signals of average levels. At high signal levels the same square would be produced for e.g. a sine or a sawtooth input.

The offset idea would have worked, though, if the offset had been somehow made proportional to the input signal's amplitude.[30] It turns out that this is a natural feature of a particular nonlinear 1-pole construct. Consider the OTA-style nonlinear 1-pole in Fig. 6.36 and imagine that instead of a saturator nonlinearity we have used the following shaper function:

$$f(x) = \begin{cases} 2x & \text{if } x \geq 0 \\ x/2 & \text{if } x \leq 0 \end{cases} \tag{6.32}$$

(Fig. 6.58 illustrates). This would mean that whenever $y_{\text{HP}} = x - y_{\text{LP}} > 0$, the cutoff is effectively doubled. When $y_{\text{HP}} = x - y_{\text{LP}} < 0$, the cutoff is effectively halved. Therefore the integrator state will be more "willing" to change in the positive direction than in the negative one.

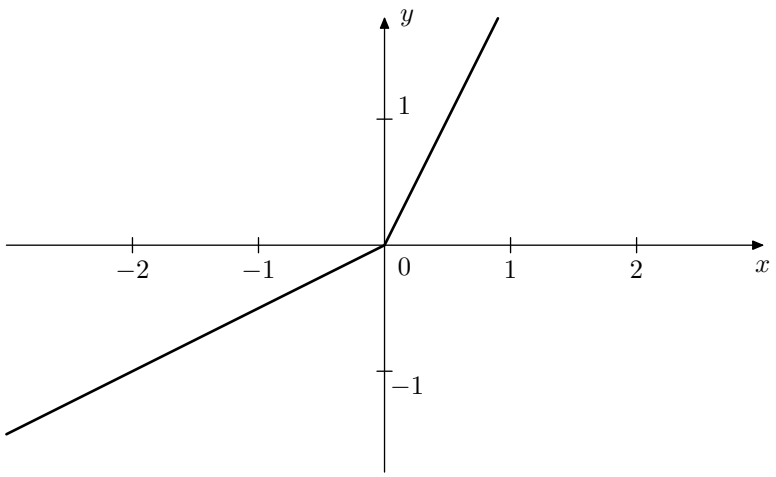


Figure 6.58: "Asymmetric cutoff" nonlinearity.

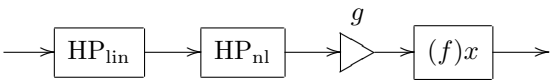
Imagine such filter receives a steady periodic signal with a zero DC offset (meaning that the average value of the signal is zero, or, in other words, there is an "equal amount" of signal above and below zero). And suppose this signal's fundamental frequency is well above the nominal cutoff of the filter. In such case a *linear* lowpass filter would have performed a kind of averaging of the input signal, thereby producing a zero output signal.[31] However in the case of using the nonlinearity (6.32) the positive input values will have "more weight" than the negative ones and the lowpass output will be nonzero.

[30] Clearly, by "amplitude" here we don't mean the momentary value of the signal but rather some kind of average or maximum.

[31] Formally the filter would have produced the DC offset of the signal at the output. The fundamental and all other harmonics, being way above filter's cutoff, would have been filtered out.

It should be inituitively clear that the lowpass output value will increase as the input signal amplitude increases and vice versa (particularly it should be obvious that in the case of the zero amplitude of the input signal the output signal will also be zero). Therefore, qualitatively such lowpass filter works as an envelope follower, the filter cutoff in a way controlling the envelope follower's response time. Respectively, the highpass output will contain the input signal with an added (or subtracted) DC offset, such offset being approximately proportional to the input signal's amplitude. This means that if initially 50% of the signal were above zero and the other 50% below zero, we now have changed this ratio to something like 80% to 20%, and this effect is happening more or less at any amplitude of the input signal.

Thus, asymmetric nonlinear shaping could be produced by the following structure:



where $\mathrm{HP}_{\mathrm{lin}}$ is an initial lowpass, killing the DC offset which might be previously contained in the input signal, $\mathrm{HP}_{\mathrm{nl}}$ is the asymmetric nonlinear highpass, introducing the DC offset into the signal, the signal is then boosted by the gain g, controlling the amount of "drive", and $f(x)$ is a usual symmetric saturator.

The nonlinearity (6.32) has a drawback that it contains a discontinuity in the 1st derivative at $x = 0$. Such discontinuity may add a noticeable amount of new harmonic content into the signal. This effect might be desired at times, but for now we would rather at least reduce it, if not avoiding it altogether, as the filter's main purpose is to introduce the DC offset into the signal. This can be achieved by smoothing the discontinuity. E.g. we could replace (6.32) with a hyperbola going at 45° through the origin, but having a similar to (6.32) asymptotic behavior (Fig. 6.59).

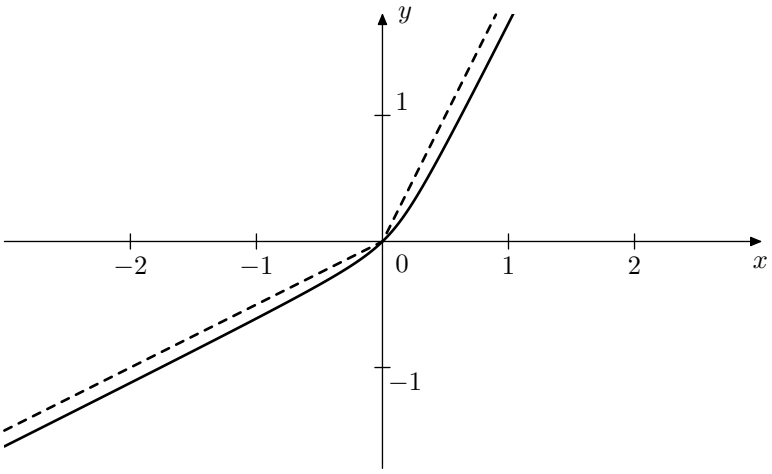


Figure 6.59: Replacing the nonlinearity (6.32) (Fig. 6.58) (dashed line) by a hyperbola.

This kind of nonlinear highpass can occur easily in analog circuits, if nonlinear resistances are involved. E.g. consider Fig. 6.60. The effective resistance

connected in series with the capacitor varies between approximately R_1 and R_2 depending on the polarity of the output voltage. Respectively the cutoff varies between $1/R_1C$ and $1/R_2C$.

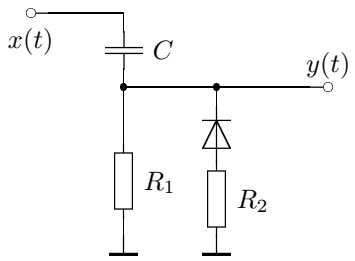


Figure 6.60: Highpass filter with asymmetric cutoff.

6.13 Antialiasing of waveshaping

Aliasing

When a signal goes through a waveshaper, the waveshaping introduces additional partials into the spectrum of the signal. These partials extend into the entire frequency range $\omega \in [0, \infty)$ for almost any waveshaper. We can show that in several steps.

First, let's consider waveshapers of the form $f(x) = x^n$ (where $n > 1$), starting with $f(x) = x^2$. Let $x(t)$ be a periodic signal. Therefore it can be represented as a sum of its harmonics:

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

where N can be finite or inifinity. Note that we are using complex-form Fourier series, therefore, assuming a real $x(t)$, we have an equal number of positive- and negative-frequency partials. Then

$$y(t) = f(x(t)) = (x(t))^2 = \left(\sum_{n=-N}^{N} X_n e^{jn\omega t} \right)^2 =$$

$$= \sum_{n_1,n_2=-N}^{N} X_{n_1} X_{n_2} e^{j(n_1+n_2)\omega t} = \sum_{n=-2N}^{2N} Y_n e^{jn\omega t} \qquad (6.33)$$

Thus, the frequencies of partials of $y(t)$ are all possible sums $n_1\omega + n_2\omega$ of frequencies of partials of $x(t)$.[32] Respectively the frequencies of the partials of $y(t)$ vary between $-2N\omega$ and $2N\omega$. That is the width of the spectrum of $y(t)$ is twice the width of the spectrum of $x(t)$.

[32] Or, if we think in terms of real-form Fourier series, where only positive-frequency partials are present, the frequencies of partials of $y(t)$ are all possible sums and differences $n_1\omega \pm n_2\omega$ of frequencies of partials of $x(t)$.

For $f(x) = x^3$ we obtain

$$y(t) = f(x(t)) = (x(t))^3 = \left(\sum_{n=-N}^{N} X_n e^{jn\omega t} \right)^3 =$$

$$= \sum_{n_1,n_2,n_3=-N}^{N} X_{n_1} X_{n_2} X_{n_3} e^{j(n_1+n_2+n_3)\omega t} = \sum_{n=-3N}^{3N} Y_n e^{jn\omega t}$$

that is the width of the spectrum is tripled. It's not difficult to generalize it to an arbitrary power of x, concluding that $f(x) = x^n$ increases the width of the spectrum of $x(t)$ n times.

It should be clear by now that, if $f(x)$ is a polynomial of order N:

$$f(x) = a_0 + a_1 x + a_2 x^2 + \ldots + a_N x^N$$

the highest-order term x^N will expand the spectrum of x n times, while the lower-order terms will also expand the spectrum of $x(t)$ but not as much, thus $f(x)$ expands the spectrum of $x(t)$ N times.

Now suppose $f(x)$ is a function of a more or less general form, expandable into Taylor series around $x = 0$:

$$f(x) = \sum_{n=0}^{\infty} \frac{f^{(n)}(0)}{n!} x^n = \sum_{n=0}^{\infty} a_n x^n$$

We can consider such $f(x)$ as a polynomial of an infinite order and thus $f(x)$ expands the spectrum of $x(t)$ by an infinite number of times.[33]

Some of the waveshapers that we were previously discussed were constructed as piecewise functions, including e.g. a piecewise polynomial saturator (6.2b). Would the saturator (6.2b), which consists of polynomial segments of order not higher than 2, thereby expand the spectrum of $f(x)$ only 2 times? It turns out that such piecewise function waveshapers also expand the spectrum an infinite number of times. Having discontinuous derivatives themselves (e.g. (6.2b) has a continuous 1st derivative, but three discontinuities of the 2nd derivative), such waveshapers also introduce discontinuous derivatives into their output signal $y(t)$. The presence of discontinuities in a signal's derivative automatically implies an infinite spectrum of the signal.[34]

Therefore all waveshapers which we have been considering until now (as well as most of the ones we could even think of) expand the spectrum of the input signal an infinite number of times. This means that discrete-time waveshaping produces aliasing.

Indeed, suppose we are given a waveshaper $f(x)$ of a general shape, so that it expands the spectrum of its input signal an infinite number times. And

[33] If the Taylor expansion of $f(x)$ has a finite convergence radius, we still can make the same argument about spectrum expansion, at least for the signals $x(t)$ which are small enough to fit into the convergence radius of the Taylor series of $f(x)$. Note that we also could expand $f(x)$ not around $x = 0$ but around some other point $x = x_0$, making the same consideration applicable for signals $x(t)$ centered around $x = x_0$.

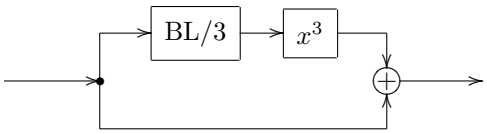
[34] A discontinuity of N-th order derivative generates harmonics rolling off as $1/n^{N+1}$. Thus a discontinuity in a 2nd derivative generates harmonics at $1/n^3$. A discontiuity in the function itself (0th derivative) generates harmonics at $1/n$ (Fourier series of sawtooth and square signals are examples of that).

imagine we are having a sampled signal $x[n]$ and its corresponding continuous bandlimited version $x(t)$. Assuming unit sampling period $T = 1$ we can write $x[n] = x(n)\ \forall n \in \mathbb{Z}$. A direct application of a waveshaper $f(x)$ to discrete-time signal $x[n]$:

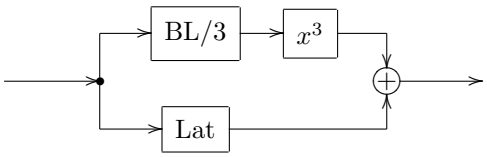
$$y[n] = f(x[n]) \tag{6.34}$$

is fully equivalent to sampling the continuous-time signal $y(t) = f(x(t))$, by simply letting $y[n] = y(n)$. However, since $f(x)$ expands the spectrum of $x(t)$ infinitely, the spectrum of $y(t)$ is not bandlimited and simply letting $y[n] = y(n)$ will result in aliased frequencies contained in $y[n]$.

Trying to use polynomial waveshapers doesn't help much. We could definitely construct polynomial antisaturators, e.g. $f(x) = x^3 + x$, whereas a purely polynomial saturator could be constucted only if we know that the input signal has a limited range, which is a pretty heavy restriction. However even $x^3 + x$ will triple the width of the spectrum, so that we'll need e.g. to bandlimit $x(t)$ to one third of the Nyquist frequency, process it by an $f(x) = x^3$ saturator and add the result to the unprocessed signal:

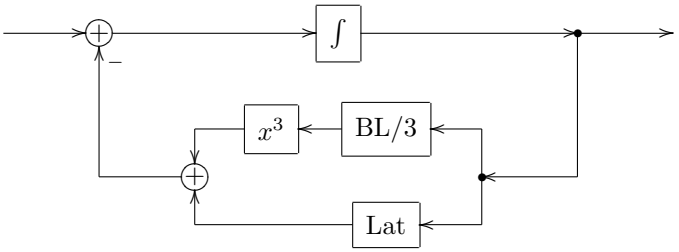


(where BL/3 denotes a filter which bandlimits the signal to 1/3 of the Nyquist frequency, and Lat denotes). Actually bandlimiting will introduce latency into the signal, so we'll need to add the same latency on the lower path



(where Lat denotes a structure which artificially introduces the same latency as introduced by BL/3).

This idea stll doesn't work really well, since antisaturators are normally used in feedback loops. We can't perform the bandlimiting inside the feedback loop, e.g.



because of the introduced latency. Doing it outside the feedback loop is also problematic. For one, we'd need to bandlimit the entire signal $x(t)$, not only

the part of it which goes through the x^3 shaper. This still could be done, though, if the sampling rate is sufficiently high (at least 3×44.1kHz). The other problem is that the signal inside the feedback loop will go infinitely many times through the waveshaper. Therefore bandlimiting of the signal prior to entering the feedback loop to 1/3 of Nyquist frequency won't really prevent the aliasing from happening.[35]

Antialiasing

The antialiasing of waveshapers is a difficult problem, not having a universally good solution at the time of writing this text. The only thing which is more or less guaranteed to work is heavy oversampling.[36] Unfortuntately, oversampling introduces latency, thus, if e.g. a waveshaper is used in a filter feedback loop, we cannot oversample locally just the waveshaper, but at least the entire feedback loop must be oversampled.

There is however an approach[37] which reduces aliasing by a noticeable amount, so that the same quality of sound can be achieved at lower sampling rates than otherwise.[38] Suppose we are having a discrete-time signal $x[n]$ going through a waveshaper $f(x)$. Instead of sending $x[n]$ through the waveshaper in discrete time, thereby producing the discrete time signal

$$y[n] = f(x[n]) \tag{6.35}$$

let's convert $x[n]$ to continuous time by means of linear interpolation. Without loss of generality we will consider the linear interpolating segment going between $x[0]$ and $x[1]$:

$$x(t) = (1-t)x[0] + tx[1] \qquad 0 \le t \le 1 \tag{6.36}$$

(where we assume unit sampling period $T = 1$). Applying the waveshaper $f(x)$ in contnuous time to this segment we obtain

$$y(t) = f(x(t)) = f((1-t)x[0] + tx[1])$$

Now we propose to compute the discrete time sample $y[1]$ as

$$y[1] = \int_0^1 y(\tau)\,\mathrm{d}\tau = \int_0^1 f((1-\tau)x[0] + \tau x[1])\,\mathrm{d}\tau = \frac{F(x[1]) - F(x[0])}{x[1] - x[0]} \tag{6.37}$$

where $F(x)$ is some antiderivative of $f(x)$, that is $F'(x) = f(x)$. Or, more generally

$$y[n] = \int_{n-1}^{n} y(\tau)\,\mathrm{d}\tau = \frac{F(x[n]) - F(x[n-1])}{x[n] - x[n-1]} \tag{6.38}$$

[35] However, it still might reduce the amount of aliasing.

[36] Higher sampling rates lead to smaller relative increments of integrator states (at the same cutoff value in Hz). Thus, at some point higher computation precision will be required. 32 bit floats might happen to become insufficient pretty quickly, but 64 bit floats should still do in a wide range of high sampling rates.

[37] The approach was proposed independently by A.Huovilainen, E.Le Bivic, Dr. J.Parker and possibly others. The application of the approach within zero-delay feedback context has been developed by the author.

[38] Still, 44.1kHz would be usually insufficient and one will need to go to 88.2kHz or even higher.

where $y(t) = f(x(t))$ and where $x(t)$ is a piecewise linear continuous-time function arising out of linear interpolation of $x[n]$. It might seem that the averaging of $y(t)$ on $t \in [n-1, n]$ in (6.38) is a somewhat arbitrary operation. However, it isn't. In fact, such averaging can be considered as one of the simplest possible forms of lowpass filtering the continuous-time signal, aiming to suppress the aliasing frequencies above Nyquist.[39]

The averaging (6.38) does a reasonable job of reducing the aliasing in $y[n]$ compared to (6.35), however it is introducing two problems: latency and ill-conditioning.

Latency

Assuming the transparency of the waveshaper at small signal levels $f(x) \approx x$ we have $F(x) \approx x^2/2$ and (6.38) turns into

$$y[n] = \frac{x^2[n]/2 - x^2[n-1]/2}{x[n] - x[n-1]} = \frac{x[n] + x[n-1]}{2} \tag{6.39}$$

The expression (6.39) describes a discrete time 1-pole lowpass filter with a cutoff at half the Nyquist frequency. Indeed, let's take the lowpass filter in Fig. 3.31. At $g = 1$ (which corresponds to $\omega_c T/2 = 1$, which in turn corresponds to prewarped half Nyquist frequency $\omega_c T/2 = \pi/4$) we have $g/(g+1) = 1/2$ and thus

$$v[n] = \frac{x[n] - s[n]}{2} \tag{6.40a}$$

$$y[n] = v[n] + s[n] = \frac{x[n] + s[n]}{2} \tag{6.40b}$$

$$s[n+1] = y[n] + v[n] = x[n] \tag{6.40c}$$

Combining (6.40b) and (6.40c) we obtain

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

which is the same as (6.39).

The lowpass filtering effect of (6.38) is actually another problem that we didn't mention so far. It arises out of the approximations that the method does when converting from discrete-time signal $x[n]$ to $x(t)$ and back from $y(t)$ to $y[n]$. This problem is however not very noticeable at sampling rates of 88.2kHz and higher. So, let's concentrate on the latency introduced by (6.39).

The averaging in (6.39) can be seen as a mid-way linear interpolation between $x[n]$ and $x[n-1]$ and thus intuitively one could expect that it introduces a half-sample delay. This is indeed the case. Taking $x[n] = e^{j\omega n}$ and assuming $|\omega| \ll 1$, so that the signal's frequency is far below the cutoff of the lowpass filter (6.39), we have

$$y[n] = \frac{e^{j\omega n} + e^{j\omega(n-1)}}{2} = \exp j\omega\left(n - \frac{1}{2}\right) \cdot \frac{e^{j\omega/2} + e^{-j\omega/2}}{2} =$$

[39] Similarly, linear interpolation can be interpreted in terms of continuous-time lowpass filtering which suppresses the aliasing discrete time spectra.

$$= \exp j\omega \left(n - \frac{1}{2}\right) \cdot \cos\frac{\omega}{2} \approx \exp j\omega \left(n - \frac{1}{2}\right)$$

where $\cos(\omega/2) \approx 1$ since $\omega \approx 0$.

It can be shown that the source of this half-sample delay is the averaging of $y(t)$ on $[n-1, n]$ done in (6.38). Taking $y(t) = e^{j\omega t}$ where $|\omega| \ll 1$, we have

$$\int_{n-1}^{n} e^{j\omega\tau}\,\mathrm{d}\tau = \frac{e^{j\omega n} - e^{j\omega(n-1)t}}{j\omega} = \exp j\omega \left(n - \frac{1}{2}\right) \cdot \frac{e^{j\omega/2} - e^{-j\omega/2}}{2j \cdot \omega/2} =$$
$$= \exp j\omega \left(n - \frac{1}{2}\right) \cdot \frac{\sin(\omega/2)}{\omega/2} = \exp j\omega \left(n - \frac{1}{2}\right) \cdot \operatorname{sinc}\frac{\omega}{2} \approx$$
$$\approx \exp j\omega \left(n - \frac{1}{2}\right)$$

(where $\operatorname{sinc} x = \frac{\sin x}{x}$ is the cardinal sine function). Thus (6.39) and (6.38) indeed introduce a delay of half sample. Inside a zero-delay feedback loop this would be a serious problem. In order to develop an idea of how to address this problem, let's look at a few examples.

Waveshaper followed by an integrator

Suppose a waveshaper is immediately preceding an integrator, as shown in Fig. 6.61 (this particularly happens in the OTA-style 1-poles in Figs. 6.35 and 6.36). Normally we recommended to use transposed direct form II integrators however this time we suggest to use a direct form I integrator (Fig. 6.62). Looking at the first half of the direct form I integrator (highlighted by the dashed line in Fig. 6.62) we can notice that it exactly implements the formula (6.39). So, the first part of the integrator implements a half-sample delay too and does so in exactly the same way as the antialiased waveshaper for low-level signals. This therefore leads to an idea to simply drop this part of the integrator, as it is done in Fig. 6.63, since we are getting the same half-delay from the waveshaper already.

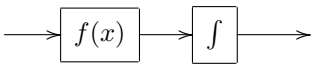


Figure 6.61: Waveshaper immedidately followed by an integrator.

It is interesting to notice that what thereby remains of the integrator is the native integrator contained in Fig. 3.3). Thus, in order to implement an antialiased waveshaper followed by a trapezoidal integrator, simply use a naive integrator instead. This effectively produces trapezoidal integrator, simultaneously "killing" the unwanted latency produced by the antialiased waveshaper.

The solution proposed in Fig. 6.63 works quite well. There is still one subtlety though, which, depending on the circumstances, may be fully academic or not. By "assigning" the functionality of the first part of the integrator to the antialiased waveshaper, we effectively positioned the $\omega_c T$ gain element into the middle of the integrator (Fig. 6.64). This doesn't affect the time-invariant behavior of the integrator, but will introduce some changes if the cutoff ω_c is varying.

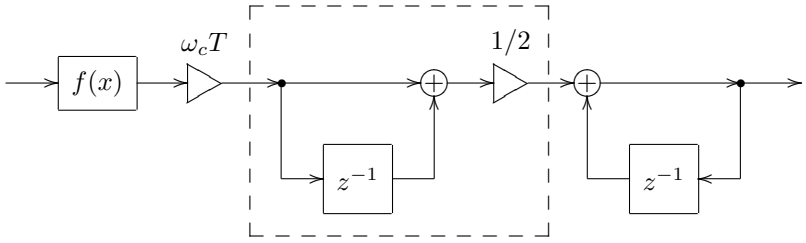


Figure 6.62: Antialiased waveshaper combined with direct form I integrator. The dashed line highlights the part of the integrator which is equivalent to the waveshaper at low signals.

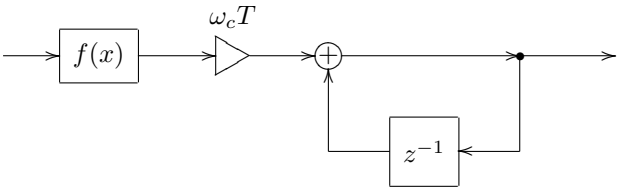


Figure 6.63: Antialiased waveshaper combined with direct form I integrator, the first part of the integrator being dropped, since its implemented by the antialiased waveshaper already.

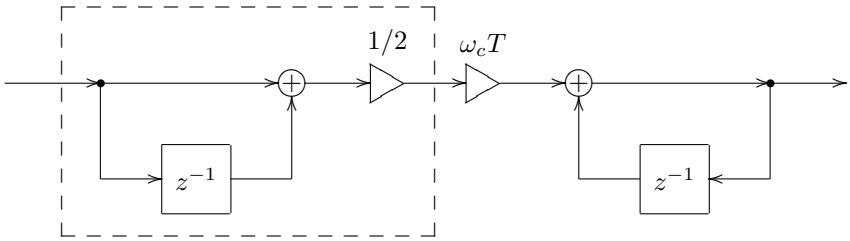


Figure 6.64: Structure in Fig. 6.63 implies positioning the $\omega_c T$ gain element in the middle of the integrator. The dashed line highlights the part which is being replaced by the antialiased waveshaper.

In order to avoid that effect, we would need to somehow include the varying $\omega_c T$ into the averaging implemented by (6.38). A straightforward possibility would be to change (6.37) into

$$u[1] = \int_0^1 \omega_c(\tau) y(\tau)\, \mathrm{d}\tau \tag{6.41}$$

(remember that we assume $T = 1$), where $u(t) = \omega_c(t)y(t) = \omega_c(t)Ty(t)$. Trapezoidal integration assumes (kind of) that the signals are varying linearly in

between the samples, therefore (6.41) can be rewritten as

$$u[1] = \int_0^1 \big((1-\tau)\omega_c[0] + \tau\omega_c[1]\big)y(\tau)\,\mathrm{d}\tau = \\ = \int_0^1 \big((1-\tau)\omega_c[0] + \tau\omega_c[1]\big)f((1-\tau)x[0] + \tau x[1])\,\mathrm{d}\tau \qquad (6.42)$$

Unfortunately, the formula (6.42) is not fully convincing. At $f(x) = x$ we would expect (6.42) to turn into ordinary trapezoidal integration of $\omega_c f(x)$ yielding

$$\frac{\omega_c[0]f(x[0]) + \omega_c[1]f(x[1])}{2}$$

However (6.42) gives in this case

$$\frac{\omega_c[0] + \omega_c[1]}{2} \cdot \frac{f(x[0]) + f(x[1])}{2}$$

Of course, (6.42) can be further artificially amended. Whether one should attempt anything like that, is an open question.

Waveshaper following an integrator

The opposite order of connection of a waveshaper and an integrator looks much better at first sight, since in this case we could use a transposed direct form I integrator (Fig. 6.65), which won't require us to reposition the cutoff gain.[40]

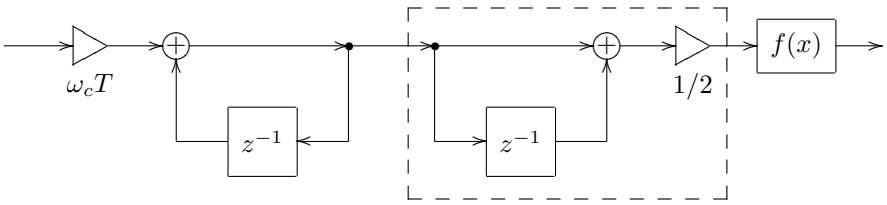


Figure 6.65: Transposed direct form I trapezoidal integrator followed by a waveshaper. The dashed line highlights the part of the integrator which is about to be dropped.

A concern which this approach is raising though, is that, as we have seen, the latency introduced by the waveshaper is caused by the averaging occurring *after* the nonlinearity, whereas in Fig. 6.65 the averaging in the integrator, which we are dropping, is occurring before the nonlinearity. On the other hand, linear interpolation, which is used to construct $x(t)$ from $x[n]$ in (6.36), is also having a lowpass filtering effect similar to the one of the averaging, while it doesn't actually matter, whether we compensate the latency before or after the waveshaper. Therefore Fig. 6.65 may also provide an acceptable solution.

[40] Technically the integrator in Fig. 6.65 is, formally, not exactly a transposed direct form II integrator, as the 1/2 gain element should have been positioned in the middle. However, since this is a constant gain, we can shift it without causing the same concerns as in the case of shifting the potentially varying $\omega_c T$ gain element.

Waveshaper followed by a 1-pole lowpass

Let's now consider the case of the feedback saturator in Fig. 6.6, where the saturator is not exactly followed by an integrator, but by a complete 1-pole? Fig. 6.66 depicts this situation explicitly showing the internal structure of the 1-pole.

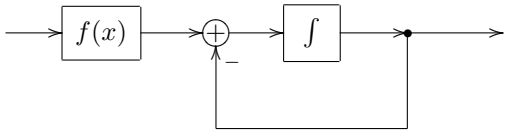


Figure 6.66: Waveshaper followed by a 1-pole lowpass.

Replacing the integrator with its direct form I implementation we obtain the structure in Fig. 6.67. Following the approach of Fig. 6.64, we reposition the $\omega_c T$ element, as shown in Fig. 6.68.

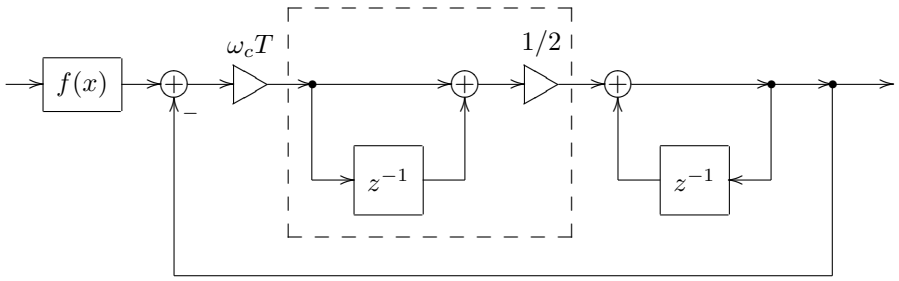


Figure 6.67: Waveshaper followed by a 1-pole lowpass built around a direct form I integrator.

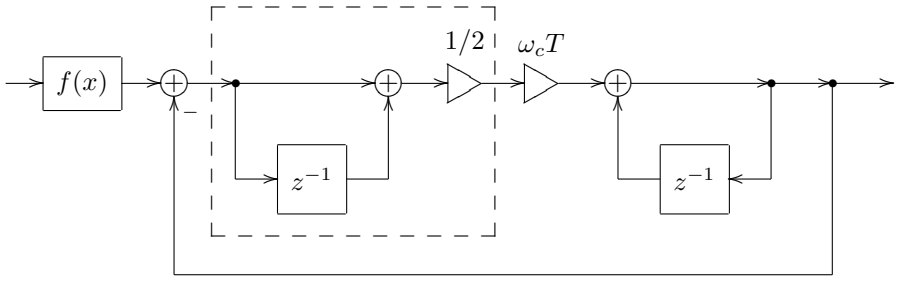


Figure 6.68: Structure from Fig. 6.67 with a changed position of the cutoff gain.

Now we would like to drop the $(1+z^{-1})/2$ part of the direct form I integrator, but only for the signal coming from the waveshaper. The feedback signal of the

1-pole should still come through the full integrator. This can be achieved by injecting the waveshaped signal into a later point of the feedback loop. The resulting structure in Fig. 6.69 thereby compensates the latency introduced by the antialiased waveshaper.

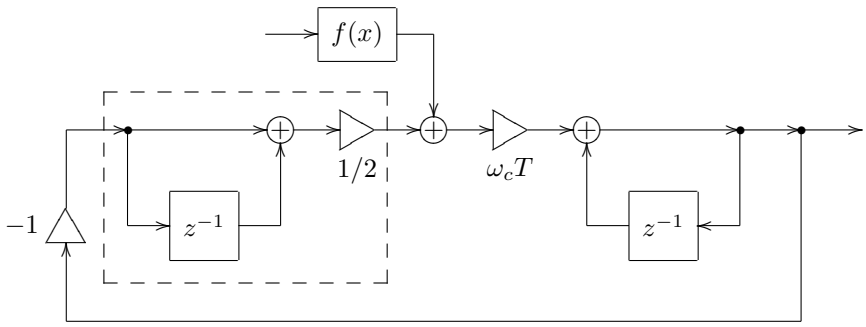


Figure 6.69: Structure from Fig. 6.68 with the waveshaped signal bypassing the first part of the direct form I integrator (thereby compensating the introduced latency).

The structure in Fig. 6.69 can be further simplified as shown in Fig. 6.70, where we "slid" the inverter "−1" all the way through $(1 + z^{-1})/2$ to the injection point of the waveshaped signal. Such change doesn't cause any noticeable effects.[41] Noticing that the two z^{-1} elements in Fig. 6.70 are actually sharing the same input signal, we can combine both into one (Fig. 6.71).

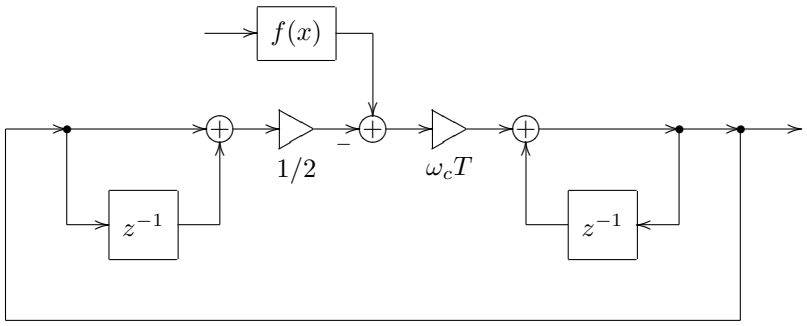


Figure 6.70: Structure from Fig. 6.69 with a changed position of the inverter.

Fig. 6.71 contains a zero-delay feedback loop, which can be resolved. Let's introduce helper variables u, v and s as shown in Fig. 6.71 and let $g = \omega_c T$. Writing the equations implied by the block diagram we have

$$v = g \cdot \left(u - \frac{v + s + s}{2} \right)$$

[41] The internal state stored in the first z^{-1} element is inverted compared to what it used to be, but this is compensated by the new position of the inverter.

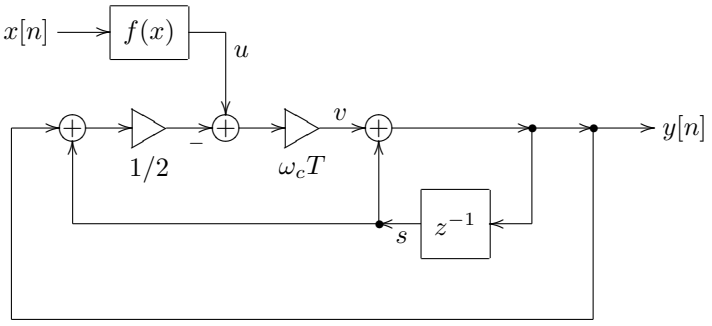


Figure 6.71: Structure from Fig. 6.70 with merged z^{-1} elements.

from where

$$v \cdot (1 + g/2) = g \cdot (u - s)$$

$$v = \frac{g}{1 + g/2} \cdot (u - s)$$

Considering that $y = v + s$ we obtain the structure in Fig. 6.72.

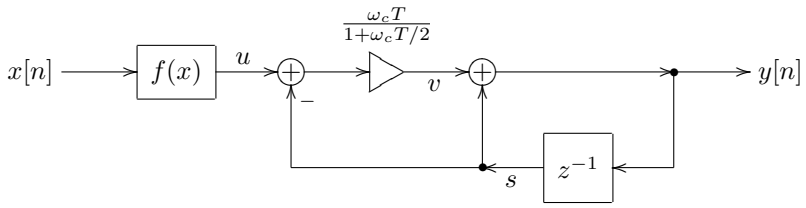


Figure 6.72: Structure from Fig. 6.71 with resolved zero-delay feedback loop, implementing Fig. 6.66 with latency compensation.

Notice that the obtained structure in Fig. 6.72 is pretty much identical to the structure of the naive 1-pole lowpass filter in Fig. 3.5, except that the cutoff gain is not $\omega_c T$ but $\omega_c T/(1+\omega_c T/2)$. Thus, in order to implement an antialiased waveshaper followed by a 1-pole lowpass, we simply use a naive 1-pole lowpass with adjusted cutoff instead, which effectively "kills" the unwanted latency.

In principle we could have tried to avoid the repositioning of the $\omega_c T$ gain element. Attempting to do so, we could have gone from Fig. 6.67 to the structure in Fig. 6.73. However, this solves only one half of the problem, namely fixing the issue in the feedback path, while the issue is still there for the waveshaped signal. The considerations of possibly including the averaging of $\omega_c T$ into the antialiased waveshaper apply, where we are having exactly the same situation as in the case of an integrator following a waveshaper.

1-pole lowpass followed by a waveshaper

In case of a 1-pole lowpass filter followed by a waveshaper (Fig. 6.74) we can use the transposed direct form I integrator, as we did in Fig. 6.65. The respective structure is shown in Fig. 6.75.

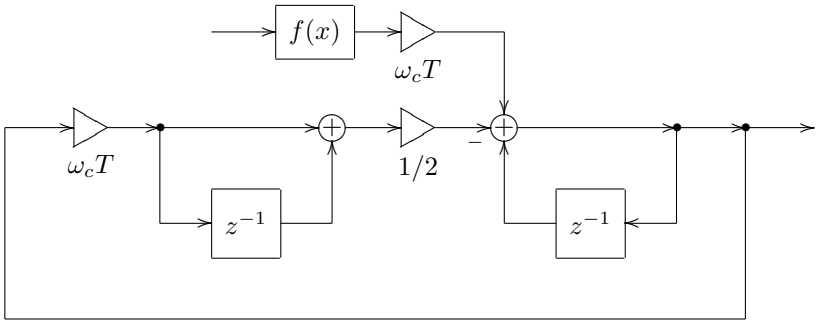


Figure 6.73: Structure from Fig. 6.67 with the waveshaped signal bypassing the first part of the direct form I integrator (thereby compensating the introduced latency) but without repositioning of $\omega_c T$ gain element.

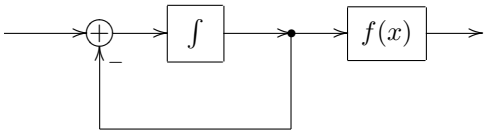


Figure 6.74: 1-pole lowpass followed by a waveshaper.

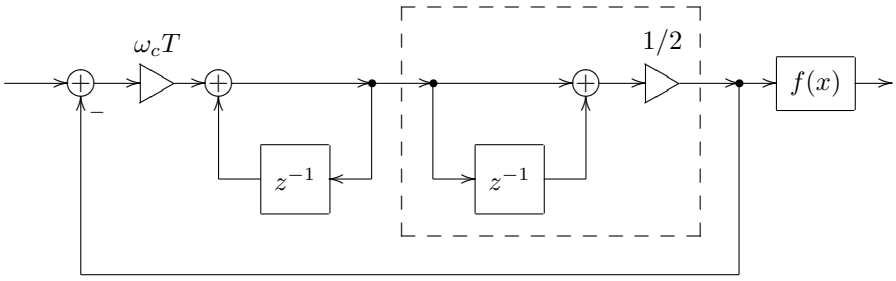


Figure 6.75: 1-pole lowpass built around a transposed direct form I integrator followed by a waveshaper.

In this case we can simply pick up the waveshaper input signal in the middle of the integrator, bypassing the second half (Fig. 6.76). Noticing that the two z^{-1} elements in Fig. 6.76 are picking up the same signal, we could merge them into a single z^{-1} element as shown in Fig. 6.77, thereby producing a direct form II integrator (compare to Fig. 3.9).

In order to resolve the zero-delay feedback loop in Fig. 6.77 we introduce helper variables u, v and s as shown in Fig. 6.77 and we let $g = \omega_c T$. Then,

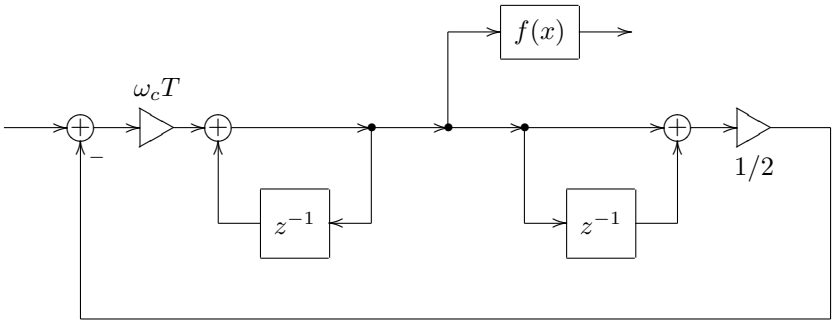


Figure 6.76: Structure from Fig. 6.75 with the waveshaper skipping the second half of the transposed direct form I integrator (thereby compensating the introduced latency).

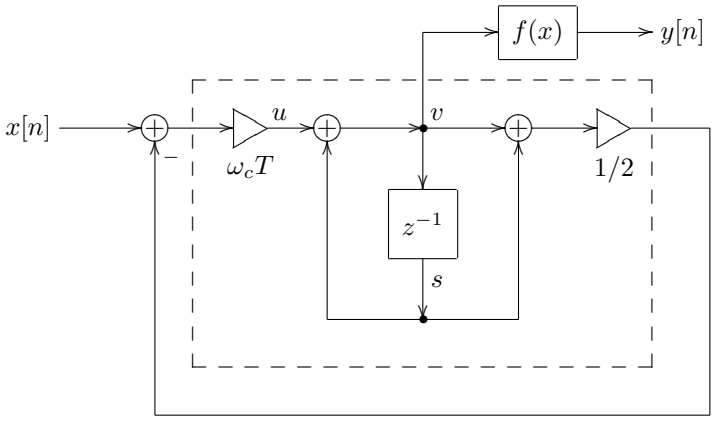


Figure 6.77: Structure from Fig. 6.76 with merged z^{-1} elements. The dashed line highlights the direct form II integrator.

writing the equations implied by the block diagram, we have

$$u = g \cdot \left(x - \frac{u + s + s}{2} \right)$$

from where

$$u \cdot (1 + g/2) = g \cdot (x - s)$$

$$u = \frac{g}{1 + g/2} \cdot (x - s)$$

Considering that $v = u + s$ we obtain the structure in Fig. 6.78.

Notice that the obtained structure in Fig. 6.78 is identical to the structure in Fig. 6.72, except for the the opposite order of the naive 1-pole lowpass and the waveshaper. Thus, in order to implement a 1-pole lowpass followed by an antialiased waveshaper we simply use a naive 1-pole lowpass with adjusted cutoff instead.

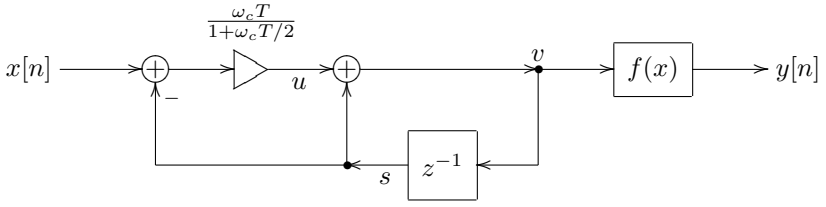


Figure 6.78: Structure from Fig. 6.77 with resolved zero-delay feedback loop, implementing Fig. 6.74 with latency compensation.

Other positions of waveshaper

In Fig. 6.66 we had a waveshaper followed by a lowpass, but imagine it was a highpass instead (Fig. 6.79).[42] In this case, even if we use the tricks similar to the ones we did in the lowpass case, we still won't be able to eliminate the latency on the feedforward path between $x(t)$ and $y(t)$.

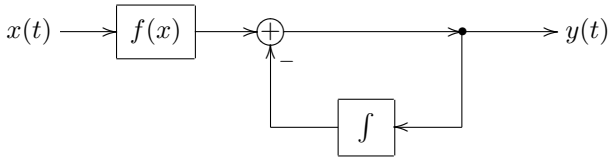
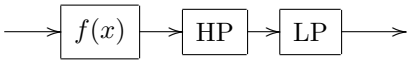
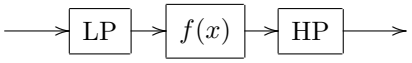


Figure 6.79: Waveshaper followed by a 1-pole highpass.

If there is e.g. a lowpass further after the the highpass:



then we can eliminate the latency by changing the lowpass, exactly as we did before. The highpass filter will work on a signal delayed by half a sample, but this will be compensated in the immediately following lowpass. Similarly, if there is a preceding lowpass:



we could consider compensating the latency by changing that lowpass. The same of course could be done if instead of a lowpass we find an integrator, or a suitable structure containing one.

However it might happen that there is no lowpass or an integrator or any other structure suitable for this purpose, neither after the waveshaper nor before it. In such cases we could artificially insert a 1-pole lowpass immediately before or after the waveshaper (Fig. 6.80), setting the cutoff of this lowpass to a very high value. In this case we could hope that the insertion of the new lowpass would not significantly change the signal, at least not in the audible range, if its cutoff is lying well above.

[42] The highpass in Fig. 6.79 might look different from the one in Fig. 2.9, however it's not difficult to realize that in fact both structures are identical.

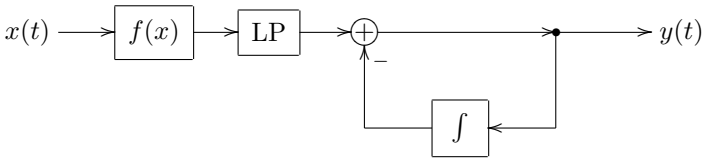


Figure 6.80: Artificially inserted 1-pole lowpass.

One still has to be careful, since such lowpass will introduce noticeable changes into the behavior of the system in the spectral range above the lowpass's cutoff and even, to an extent, below its cutoff. Even though a lowpass generally reduces the amplitude of signals, due to the changes in the phase it could increase the system's resonance, causing the system to turn to selfoscillation earlier than expected.[43] In a nonlinear system the inaudible parts of the spectrum could become audible through the so-called intermodulation distortion.[44] So, it's a good idea to test for the possible artifacts created by the introduction of such lowpass.

Zero-delay feedback equation

The appearance of antialiased waveshapers in a zero-delay feedback loop creates the question of solving the arising zero-delay feedback equations. Fortunately, this doesn't create any new problems, as the inistantaneous response of an antialiased waveshaper can be represented in familiear terms.

Indeed, according to (6.38) the instantaneous response of an antialiased waveshaper is simply another waveshaper:

$$\tilde{f}(x) = \frac{F(x) - F(a)}{x - a} \tag{6.43}$$

where $a = x[n-1]$ is the waveshaper function's parameter, which is having a fixed value at each given time moment n. Thus we obtain the already familiar kind of a zero-delay feedback equation with a waveshaper.

Ill-conditioning

If $x[n] \approx x[n-1]$, the denominator of (6.38) will be close to zero. Rather fortunately this also means that the numerator will be close to zero as well, so that, at least formally, their ratio should produce a finite value. However practically this could mean precision losses in the numeric evaluation of the right-hand side of (6.38) (or division by zero if $x[n] = x[n-1]$).

Since $x[n] \approx x[n-1]$, the value of the interpolated signal $x(t)$ and respectively the value of $y(t) = f(x(t))$ shouldn't change much on $[n-1, n]$ and thus the integral in (6.38) can be well approximated by a value of $y(t)$ somewhere on that

[43] This would be particularly the case in a 4-pole lowpass ladder filter, where the effect is noticeable at ladder filter's cutoff settings comparable or higher than the cutoff of the added lowpass.

[44] Recall that e.g. in (6.33) the output signal of a waveshaper contained all sums and differences of the frequencies of the original signal. Thus if the difference of two frequencies, both lying well above the audible range, falls into the audible range, these originally inaudible partials will create an audible one.

interval.[45] In principle we could take any point on that interval, but intuitively we should expect the midway point to give the best result, and thus we take

$$y[n] = f\left(\frac{x[n] + x[n-1]}{2}\right) \qquad \text{if } x[n] \approx x[n-1] \tag{6.44}$$

Notice that at $f(x) \approx x$ (6.44) turns into (6.39).

The fallback formula (6.44) creates no new problems for the solution of the zero-delay feedback equation, since in instantaneous response terms it looks like another waveshaper

$$\tilde{f}(x) = f\left(\frac{x + a}{2}\right) \tag{6.45}$$

where $a = x[n-1]$. Note, however, that when using iterative approaches to the solution of the zero-delay feedback equation, we potentially may need to switch between (6.43) and (6.45) on each iteration step.

The choice between the normal and the ill-conditioned case formulas should depend on the comparison of estimated precision losses in (6.38) and the error in (6.44). In that regard note, that it might be a good idea to choose the antiderivative $F(x)$ so that $F(0) = 0$. This could improve the precision of numerical computation of (6.38) and (6.43) at low signal levels, as subtraction of two close numbers is the main source of precision losses here. On the other hand, the main source of error in (6.44) and (6.45) is nonlinear behavior of $f(x)$ on the segment lying between $x[n-1]$ and $x[n]$.[46]

SUMMARY

Nonlinear filters can be constructed by introducing waveshapers into block diagrams. Two important types of waveshapers are saturators and antisaturators. Saturators used in resonating feedback loops prevent the signal level from infinite growth. Antisaturators have a similar effect in damping feedback paths.

The discussed types of usage of saturators in filters included feedback loop saturation, transistor ladder-style 1-pole saturation and OTA-style 1-pole saturation. The discussed usage of antisaturators included the diode clipper-style saturation of 1-poles and the usage in the damping path of an SVF.

Waveshapers usually turn zero-delay feedback equations into transcendental ones, which then need to be solved using approximate or numeric methods, although in some cases analytic solution is possible.

Discrete-time waveshaping produces aliasing, which might need to be mitigated using oversampling and/or some more advanced methods.

[45] This is more precisely stated by the mean value theorem.

[46] Note that if $f(x)$ is fully linear on that segment, then (6.44) gives the exact answer. One could also obtain an estimation of the error of (6.44) by expanding $f(x)$ in Taylor series around $x = (x[n] + x[n-1])/2$ and noticing that applying (6.38) just to the first two terms of this expansion gives (6.44) (as the contribution of the first-order term of the series turns out to be zero). Therefore the error of (6.44) is equal to the contribution of the remaining terms of the Taylor series to (6.38).

Chapter 7

State-space form

Starting with this chapter we begin the discussion of subjects of a more theoretical nature, not in the sense that they are not useful for practical purposes, but rather that one can already do a lot without the respective knowledge. Simultaneously the mathematical level of the presented text is generally higher than in the previous chapters. Readers who are not too interested in the respective subjects may consider skipping directly to Chapter 11, where the discussion returns to the previous "practical" level.

Transfer functions fully describe the behavior of linear time-invariant systems, but, as we already have seen, once the system parameters start to vary, we find out that some important information about the system topology is lacking. The state-space form provides a mathematical way to describe a system without losing the essential information about the system's topology.[1] Practically it's not much different from block diagrams, just instead of a graphical representation of a system we represent it by mathematical equations. The state-space form can help to obtain new insights into the way how differential and difference systems work.

7.1 Differential state-space form

The term state-space form simply means that a differential system is written in the form of ordinary differential equations of the first order, where the differentiation is done with respect to time, and the equations have been algebraically resolved in respect to derivatives. E.g. suppose we are interested in a 2-pole allpass based on the state-variable filter (Fig. 4.1). In principle we already have the respective equations in (4.1) but for the sake of demonstration let's reobtain them from the block diagram in Fig. 4.1.

Let $u_1 = y_{\mathrm{BP}}$ denote the output of the first integrator and $u_2 = y_{\mathrm{LP}}$ denote the output of the second integrator. The input of the first integrator is $x - 2Ry_{\mathrm{BP}} - y_{\mathrm{LP}}$, thus

$$u_1 = \int \omega_c \left(x - 2Ru_1 - u_2\right) \mathrm{d}t$$

[1] Except in cases where continuous-time block diagrams contain instantaneously unstable integratorless feedback loops.

The output of the second integrator is simply y_{BP}:

$$u_2 = \int \omega_c u_1 \, \mathrm{d}t$$

According to (4.23), the allpass signal can be obtained as $y = x - 4Ry_{\mathrm{BP}}$:

$$y = x - 4Ru_1$$

Writing all three equations together:

$$\begin{aligned} u_1 &= \int \omega_c \left(x - 2Ru_1 - u_2\right) \, \mathrm{d}t \\ u_2 &= \int \omega_c u_1 \, \mathrm{d}t \\ y &= x - 4Ru_1 \end{aligned}$$

we have obtained the state-space form representation of Fig. 4.1, except that we are having integral rather than differential equations. From the mathematical point of view this is no more than a matter of notation and we can equivalently rewrite the same equations as

$$\begin{aligned} \dot{u}_1 &= \omega_c \left(x - 2Ru_1 - u_2\right) \\ \dot{u}_2 &= \omega_c u_1 \\ y &= x - 4Ru_1 \end{aligned}$$

It is common to write the state-space equations in the matrix form:

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{pmatrix} u_1 \\ u_2 \end{pmatrix} = \begin{pmatrix} -2R\omega_c & -\omega_c \\ \omega_c & 0 \end{pmatrix}\begin{pmatrix} u_1 \\ u_2 \end{pmatrix} + \begin{pmatrix} \omega_c \\ 0 \end{pmatrix} x \tag{7.1a}$$

$$y = \begin{pmatrix} -4R & 0 \end{pmatrix}\begin{pmatrix} u_1 \\ u_2 \end{pmatrix} + x \tag{7.1b}$$

or, by introducing

$$\begin{aligned} A &= \begin{pmatrix} -2R\omega_c & -\omega_c \\ \omega_c & 0 \end{pmatrix} \\ \mathbf{b} &= \begin{pmatrix} \omega_c & 0 \end{pmatrix}^{\mathsf{T}} \\ \mathbf{c}^{\mathsf{T}} &= \begin{pmatrix} -4R & 0 \end{pmatrix} \\ d &= 1 \end{aligned}$$

we rewrite the same in vector notation:

$$\begin{aligned} \dot{\mathbf{u}} &= A\mathbf{u} + \mathbf{b}x \\ y &= \mathbf{c}^{\mathsf{T}}\mathbf{u} + d \cdot x \end{aligned}$$

This is the general *state-space form* for a single-input single-output differential system. We can promote it further to multiple inputs and multiple outputs by promoting x and y to vectors and promoting $\mathbf{b}$, $\mathbf{c}^{\mathsf{T}}$ and d to matrices:

$$\dot{\mathbf{u}} = A\mathbf{u} + B\mathbf{x} \tag{7.2a}$$

$$\mathbf{y} = C\mathbf{u} + D\mathbf{x} \tag{7.2b}$$

E.g. for a single-input multiple-output LP/BP/HP SVF the equation (7.1b) turns into

$$\mathbf{y} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \\ -2R & -1 \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} x$$

The term state-space form originates from the fact that the vector of differential variables $\mathbf{u}$ represents the states of the integrators, or simply the state of the system. Respectively the linear space of vectors $\mathbf{u}$ is referred to as the *state space* of the system.

The state-space form encodes the essential information about the system's topology, namely, which gains precede the integrators and which follow the integrators. Specifically, B is the matrix of gains occuring on the paths from the inputs to the integrators, C is the matrix of gains occurring on the paths from the integrators to the outputs, D is the matrix of gains bypassing the integrators and A is the matrix of gains on the feedback paths, thus they simultaneously precede and follow the integrators.

Integral form

Equations (7.2) can be rewritten in the integral form, which is merely a notational switch:

$$\mathbf{u} = \int (A\mathbf{u} + B\mathbf{x})\ \mathrm{d}t = \mathbf{u}(0) + \int_0^t (A\mathbf{u} + B\mathbf{x})\ \mathrm{d}\tau \tag{7.3a}$$

$$\mathbf{y} = C\mathbf{u} + D\mathbf{x} \tag{7.3b}$$

The integral form also allows to convert the state-space form back to the block diagram form. Each line of (7.3a) corresponds to an integrator, the respective right-hand side describing the integrator's input signal.

Nonlinear state-space form

The right-hand sides of the equations (7.2) actually can be arbitary nonlinear vector functions of vector arguments, in which case we could write the equations as

$$\dot{\mathbf{u}} = F(\mathbf{u}, \mathbf{x})$$
$$\dot{\mathbf{y}} = G(\mathbf{u}, \mathbf{x})$$

The discussion of nonlinear systems has been done in Chapter 6. Most of the ideas discussed in Chapter 6 can be equally applied to the systems expressed as a state-space form, and we won't discuss nonlinear state-space forms further.

7.2 Integratorless feedback

Before we can convert a block diagram (or an equation system, for that matter) into a state-space form we need to resolve integratorless feedback loops, if there are any. Integratorless feedback is a continuous-time version of zero-delay feedback. While zero-delay feedback loops in discrete time systems are

the loops containing no unit delays, integratorless feedback loops in continuous time systems are the loops containing no integrators.

The resolution of integratorless feedback is therefore subject to the same considerations and procedures as the resolution of zero-delay feedback. We are going to demonstrate this using the TSK allpass from Fig. 5.35 as an example.

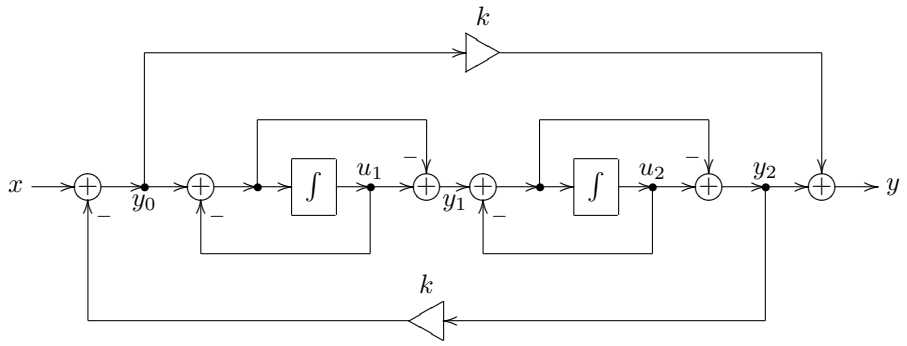


Figure 7.1: Allpass TSK filter from Fig. 5.35 with expanded 1-pole allpass structures.

Expanding the internal structures of the 1-pole allpasses in Fig. 5.35 we obtain the structure in Fig. 7.1. Denoting the 1-pole allpass states as u_1 and u_2, their output signals as y_1 and y_2 and the input of the first 1-pole allpass as y_0 (as shown in Fig. 7.1) we obtain the following equations:

$$\begin{aligned}
\dot{u}_1 &= y_0 - u_1 \\
y_1 &= u_1 - (y_0 - u_1) = 2u_1 - y_0 \\
\dot{u}_2 &= y_1 - u_2 \\
y_2 &= u_2 - (y_1 - u_2) = 2u_2 - y_1 \\
y_0 &= x - ky_2 \\
y &= y_2 + ky_0
\end{aligned}$$

where this time we have assumed $\omega_c = 1$ for simplicity.

Apparently Fig. 7.1 contains an integratorless feedback loop, starting at y_0, going through the highpass path of the first allpass to y_1, then through the highpass path of the second allpass to y_2 and returning via the global feedback path to y_0. This loop contains three inverters and a gain of k, thus the total gain of this integratorless feedback loop is $-k$ and it is not instantaneously unstable provided $k > -1$. Under this assumption we can resolve it algebraically. Selecting just the equations for y_n we have

$$\begin{aligned}
y_1 &= 2u_1 - y_0 \\
y_2 &= 2u_2 - y_1 \\
y_0 &= x - ky_2
\end{aligned}$$

We would like to solve for y_0, therefore we first eliminate y_2 in the third equation:

$$y_0 = x - k(2u_2 - y_1)$$

and then y_1 in the just obtained equation:

$$y_0 = x - k(2u_2 - (2u_1 - y_0)) = x - ky_0 + 2k(u_1 - u_2)$$

$$(1+k)y_0 = x + 2k(u_1 - u_2)$$

and

$$y_0 = \frac{x + 2k(u_1 - u_2)}{1+k}$$

Notice that the denominator corresponds to the instantaneously unstable case occuring for $k < -1$.

Now that we have resolved the integratorless feedback, we need to substitute the resolution result into the remaining equations of the original equation system:

$$\begin{aligned}
\dot{u}_1 &= y_0 - u_1 = \left(\frac{2k}{1+k} - 1\right) u_1 - \frac{2k}{1+k} u_2 + \frac{1}{1+k} x = \frac{(k-1)u_1 - 2ku_2 + x}{1+k} \\
\dot{u}_2 &= y_1 - u_2 = 2u_1 - y_0 - u_2 = \\
&= \left(2 - \frac{2k}{1+k}\right) u_1 - \left(1 - \frac{2k}{1+k}\right) u_2 - \frac{1}{1+k} x = \\
&= \frac{2u_1 + (k-1)u_2 - x}{1+k} \\
y &= y_2 + ky_0 = 2u_2 - y_1 + ky_0 = 2u_2 - (2u_1 - y_0) + ky_0 = \\
&= 2(u_2 - u_1) + (1+k)y_0 = 2(u_2 - u_1) + x + 2k(u_1 - u_2) = \\
&= 2(k-1)u_1 - 2(k-1)u_2 + x
\end{aligned}$$

Or, in the matrix form

$$\dot{\mathbf{u}} = \frac{1}{k+1} \begin{pmatrix} k-1 & -2k \\ 2 & k-1 \end{pmatrix} \mathbf{u} + \frac{1}{k+1} \begin{pmatrix} 1 \\ -1 \end{pmatrix} x \tag{7.4a}$$

$$y = 2(k-1) \cdot \begin{pmatrix} 1 & -1 \end{pmatrix} \mathbf{u} + x \tag{7.4b}$$

7.3 Transfer matrix

If $\mathbf{x}(t) = \mathbf{X}(s)e^{st}$, all other signals in the system have the same exponential form and the system turns into

$$\begin{aligned}
s\mathbf{U}(s)e^{st} &= A\mathbf{U}(s)e^{st} + B\mathbf{X}(s)e^{st} \\
\mathbf{Y}(s)e^{st} &= C\mathbf{U}(s)e^{st} + D\mathbf{X}(s)e^{st}
\end{aligned}$$

or

$$\begin{aligned}
s\mathbf{U}(s) &= A\mathbf{U}(s) + B\mathbf{X}(s) \\
\mathbf{Y}(s) &= C\mathbf{U}(s) + D\mathbf{X}(s)
\end{aligned}$$

The first of the two equations is a linear equation system in a matrix form in respect to the unknown $\mathbf{U}(s)$ and the solution is found from

$$(s - A)\mathbf{U}(s) = B\mathbf{X}(s)$$

(where $s - A$ is a short notation for $sI - A$ where I is identify matrix), and

$$\mathbf{U}(s) = (s - A)^{-1} B \cdot \mathbf{X}(s) \tag{7.5}$$

and thus

$$\mathbf{Y}(s) = C\mathbf{U}(s) + D\mathbf{X}(s) = C(s - A)^{-1} B \cdot \mathbf{X}(s) + D \cdot \mathbf{X}(s)$$

Introducing the matrix

$$H(s) = C(s - A)^{-1} B + D = \frac{C \operatorname{adj}(s - A) B}{\det(s - A)} + D \tag{7.6}$$

we have

$$\mathbf{Y}(s) = H(s)\mathbf{X}(s)$$

Thus $H(s)$ is the *transfer matrix* of the system, its elements being the individual transfer functions corresponding to all possible input-output pairs of the system. In case of a single-input single-output system $H(s)$ reduces to a 1×1 matrix:

$$H(s) = \mathbf{c}^{\mathsf{T}}(s - A)^{-1}\mathbf{b} + d = \frac{\mathbf{c}^{\mathsf{T}} \operatorname{adj}(s - A)\mathbf{b}}{\det(s - A)} + d \tag{7.7}$$

being simply the familiar transfer function.

From the formula (7.6) or (7.7) we can derive why the transfer functions of system built on integrators are nonstrictly proper rational functions. Indeed, the elements of $\operatorname{adj}(s - A)$ are polynomials of s of up to $(N-1)$-th order (where N is the dimension of the state space, that is simply the number of integrators). On the other hand, $\det(s - A)$ is a polynomial of s of N-th order. Therefore, the elements of $(s - A)^{-1}$ are rational functions of s sharing the same N-th order denominator $\det(s - A)$ and having numerators of up to $(N-1)$-th order. Thus, if $D = 0$, the elements of $H(s)$ are strictly proper rational functions.

If $D \neq 0$, (7.6) turns into

$$H(s) = \frac{C \operatorname{adj}(s - A) B}{\det(s - A)} + D = \frac{C \operatorname{adj}(s - A) B + D \det(s - A)}{\det(s - A)}$$

and thus the numerators of the elements of $H(s)$ become polynomials of order N, if the respective element of matrix D is nonzero. Thus, the transfer function becomes nonstrictly proper only if there is a direct (in the sense that it contains no integrators) path from the input to the output.

Note that, since the denominator of the transfer function(s) is $\det(s - A)$, it follows that the roots of the $\det(s - A)$ polynomial are the system poles. At the same time the roots of $\det(s - A) = 0$ are the eigenvalues of A. Thus, eigenvalues of A are the system poles.

7.4 Transposition

Computing the transfer matrix transpose we obtain from (7.6):

$$H^{\mathsf{T}}(s) = \left(C(s - A)^{-1} B + D\right)^{\mathsf{T}} = B^{\mathsf{T}}(s - A^{\mathsf{T}})^{-1} C^{\mathsf{T}} + D^{\mathsf{T}}$$

This looks like a transfer function of another system:

$$\dot{\mathbf{u}}' = A'\mathbf{u}' + B'\mathbf{x}'$$
$$\mathbf{y}' = C'\mathbf{u}' + D'\mathbf{x}'$$

where

$$A' = A^{\mathsf{T}} \quad B' = C^{\mathsf{T}} \quad C' = B^{\mathsf{T}} \quad D' = D^{\mathsf{T}}$$

We will refer to this new system as *transposed system*. The transposition of the state-space form corresponds to the transposition of block diagrams described in Section 2.14. Particularly, we swap the input gains B for the output gains C and vice versa.

So, the transfer function of the transposed system is

$$H'(s) = C'(s - A')^{-1}B' + D' = B^{\mathsf{T}}(s - A^{\mathsf{T}})^{-1}C^{\mathsf{T}} + D^{\mathsf{T}} = H^{\mathsf{T}}(s)$$

where

$$\mathbf{Y}'(s) = H'(s)\mathbf{X}'(s) = H^{\mathsf{T}}(s)\mathbf{X}'(s)$$

or, in component form

$$Y'_n(s) = H'_{nm}(s)X'_m(s) = H_{mn}(s)X'_m(s)$$

while for the original system we have

$$Y_m(s) = H_{mn}(s)X_n(s)$$

But the input/output pair x'_m, y'_n, is the transposed system corresponds to the input/output pair x_n, y_m of the original system and the transfer function for each pair is H_{mn}. Thus, transposition preserves the transfer function relationships between the respective input/output pairs.

7.5 Basis changes

In the process of further analysis of state-space forms it will be highly useful to be able to change the basis of the state-space. Since a basis change is equivalent to a linear transformation of the linear space, let $\mathbf{u}' = T\mathbf{u}$ denote such transformation (where T is some nonsingular matrix). Remember that what we are doing is changing the basis of the space, the transformation T is just a way to notate the respective change of coordinates! Then $\mathbf{u} = T^{-1}\mathbf{u}'$ and we can rewrite (7.2a) in terms of $\mathbf{u}'$:

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(T^{-1}\mathbf{u}'\right) = AT^{-1}\mathbf{u}' + B\mathbf{x} \tag{7.8}$$
$$T^{-1}\dot{\mathbf{u}}' = AT^{-1}\mathbf{u}' + B\mathbf{x}$$
$$\dot{\mathbf{u}}' = TAT^{-1}\mathbf{u}' + TB\mathbf{x}$$

Respectively, (7.2b) in terms of $\mathbf{u}'$ turns into

$$\mathbf{y} = C\mathbf{u} + D\mathbf{x} = CT^{-1}\mathbf{u}' + D\mathbf{x}$$

Introducing

$$A' = TAT^{-1} \quad B' = TB \quad C' = CT^{-1} \tag{7.9}$$

we obtain

$$\dot{\mathbf{u}}' = A'\mathbf{u}' + B'\mathbf{x} \tag{7.10a}$$
$$\mathbf{y} = C'\mathbf{u}' + D\mathbf{x} \tag{7.10b}$$

which has exactly the same form as (7.2). That is we have obtained a new state-space representation of the system in the new basis. Note that thereby we didn't change the basis of the spaces of the input signals $\mathbf{x}$ or the output signals $\mathbf{y}$, but solely the basis of the state signals $\mathbf{u}$. Thus, the basis change is a purely internal operation and doesn't affect the components of the vectors $\mathbf{x}$ and $\mathbf{y}$. Respectively, the transfer matrix is not affected either, which can be explicitly shown by computing the transfer matrix in the new basis:

$$\begin{aligned} H'(s) &= C'(s - A')^{-1}B' + D = CT^{-1}(s - TAT^{-1})^{-1}TB + D = \\ &= CT^{-1}(TsT^{-1} - TAT^{-1})^{-1}TB + D = \\ &= CT^{-1}(T(s - A)T^{-1})^{-1}TB + D = \\ &= CT^{-1}T(s - A)^{-1}T^{-1}TB + D = \\ &= C(s - A)^{-1}B + D = H(s) \end{aligned}$$

7.6 Matrix exponential

Another tool which we will need is the concept of the *matrix exponential*. We define the matrix exponential by writing the Taylor series for an ordinary exponential:

$$e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots$$

and replacing x with a matrix:

$$e^X = 1 + X + \frac{X^2}{2!} + \frac{X^3}{3!} + \dots \tag{7.11}$$

The properties of the matrix exponential are similar to the ones of the ordinary exponential, except that typically the commutativity of the involved matrices is required. Particularly, the following properties are derived from (7.11) in a straightforward manner, under the assumption $XY = YX$ and $XX' = X'X$:

$$e^X Y = Y e^X \qquad e^{X+Y} = e^X e^Y = e^Y e^X \qquad \frac{\mathrm{d}}{\mathrm{d}t} e^{X(t)} = e^X X' = X' e^X$$

The value of e^X is particularly easy to compute if X is diagonal:

$$X = \begin{pmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ & & & \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & \cdots & \lambda_N \end{pmatrix}$$

In this case formula (7.11) turns into N parallel Taylor series for e^{λ_n} and we simply have

$$e^X = \begin{pmatrix} e^{\lambda_1} & 0 & \cdots & 0 \\ 0 & e^{\lambda_2} & \cdots & 0 \\ & & & \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & \cdots & e^{\lambda_N} \end{pmatrix}$$

If X is not diagonal, but diagonalizable by a similarity transformation TXT^{-1}, the value e^X can be computed by noticing that matrix exponential commutes with similarity trasformation:

$$e^{TXT^{-1}} = Te^X T^{-1} \tag{7.12}$$

which allows to express e^X via $e^{TXT^{-1}}$. The formula (7.12) is obtainable from (7.11) in a straightforward manner as well, where we also notice that (7.12) holds for any T and X, they don't have to commute.

If X is not diagonalizable, then Jordan normal form can be used instead. We are going to address this case slightly later.

7.7 Transient response

The differential state-space equation (7.2a) can be solved in the same fashion as we solved the differential equations for the 1-pole in Section 2.15. Indeed, the difference between (7.2a) and the Jordan 1-pole (2.21) is that the former has matrix form. Also in (7.2a) the input signal is additionally multiplied by the matrix B, but that doesn't change the picture essentially.

Repeating the same steps as in in Section 2.15, we multiply both sides of (7.2a) by the matrix exponential e^{-At}:

$$e^{-At}\dot{\mathbf{u}} = e^{-At}A\mathbf{u} + e^{-At}B\mathbf{x}$$

or

$$e^{-At}\dot{\mathbf{u}} - e^{-At}A\mathbf{u} = e^{-At}B\mathbf{x}$$

Noticing that

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(e^{-At}\mathbf{u}\right) = e^{-At}\dot{\mathbf{u}} - e^{-At}A\mathbf{u}$$

we rewrite the state-space differential equation further as

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(e^{-At}\mathbf{u}\right) = e^{-At}B\mathbf{x}$$

Integrating with respect to time from 0 to t:

$$e^{-At}\mathbf{u} - \mathbf{u}(0) = \int_0^t e^{-A\tau}B\mathbf{x}\,\mathrm{d}t$$

$$\mathbf{u} = e^{At}\mathbf{u}(0) + e^{At}\int_0^t e^{-A\tau}B\mathbf{x}\,\mathrm{d}t = e^{At}\mathbf{u}(0) + \int_0^t e^{A(t-\tau)}B\mathbf{x}\,\mathrm{d}t \tag{7.13}$$

The formula (7.13) is directly analogous to (2.22). Further, assuming complex exponential $\mathbf{x}(t) = \mathbf{X}(s)e^{st}$ (note that all elements of $\mathbf{x}$ share the same exponential e^{st}, just with different amplitudes) we continue as:

$$\begin{aligned}\mathbf{u} &= e^{At}\mathbf{u}(0) + e^{At}\int_0^t e^{-A\tau}B\mathbf{X}(s)e^{s\tau}\,\mathrm{d}t = \\ &= e^{At}\mathbf{u}(0) + e^{At}\int_0^t e^{(s-A)\tau}\,\mathrm{d}t \cdot B\mathbf{X}(s) = \end{aligned}$$

$$= e^{At}\mathbf{u}(0) + e^{At}(s-A)^{-1}e^{(s-A)\tau}\Big|_{\tau=0}^{t} \cdot B\mathbf{X}(s) =$$
$$= e^{At}\mathbf{u}(0) + e^{At}(s-A)^{-1}\left(e^{(s-A)t} - 1\right)B\mathbf{X}(s) =$$
$$= e^{At}\left(\mathbf{u}(0) - (s-A)^{-1}B\mathbf{X}(s)\right) + (s-A)^{-1}B\mathbf{X}(s)e^{st}$$

Comparing to the transfer matrix for $\mathbf{u}(t)$ defined by (7.5) we introduce the steady-state response

$$\mathbf{u}_s(t) = (s-A)^{-1}B \cdot \mathbf{X}(s)e^{st}$$

and therefore

$$\mathbf{u}(t) = e^{At}\left(\mathbf{u}(0) - \mathbf{u}_s(0)\right) + \mathbf{u}_s(t) = \mathbf{u}_t(t) + \mathbf{u}_s(t) \tag{7.14}$$

where $\mathbf{u}_t(t)$ is the transient response.

Note that we have just explicitly obtained the fact (previously shown only for the system orders $N \leq 2$) that, given a complex exponential input $\mathbf{X}(s)e^{st}$, the elements of the steady-state response $\mathbf{u}_s$ will be the same complex exponentials e^{st}, just with different amplitudes. An immediately following conclusion is that the steady-state signals $\mathbf{y}$, being linear combinations of $\mathbf{u}$ and $\mathbf{x}$, are also the same complex exponentials e^{st}. In fact, any other steady-state signal in the system, being a linear combination of $\mathbf{u}$ and $\mathbf{x}$, is the same complex exponential e^{st}.

In a fully analogous to the 1-pole case way we can show that (7.14) also holds for

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

in which case

$$\mathbf{u}_s(t) = \int_{\sigma-j\infty}^{\sigma+j\infty} (s-A)^{-1}B\mathbf{X}(s)e^{st}\,\frac{\mathrm{d}s}{2\pi j}$$

Substituting (7.14) into (7.2b) we obtain

$$\mathbf{y}(t) = Ce^{At}\left(\mathbf{u}(0) - \mathbf{u}_s(0)\right) + C\mathbf{u}_s(t) + D\mathbf{x}(t) =$$
$$= e^{At}\left((C\mathbf{u}(0) + D\mathbf{x}(0)) - (C\mathbf{u}_s(0) + D\mathbf{x}(0))\right) + C\mathbf{u}_s(t) + D\mathbf{x}(t) =$$
$$= e^{At}\left(\mathbf{y}(0) - \mathbf{y}_s(0)\right) + \mathbf{y}_s(t) = \mathbf{y}_t(t) + \mathbf{y}_s(t)$$

where

$$\mathbf{y}_t(t) = C\mathbf{u}_t(t) = e^{At}\left(\mathbf{y}(0) - \mathbf{y}_s(0)\right)$$

and

$$\mathbf{y}_s(t) = C\mathbf{u}_s(t) + D\mathbf{x}(t) = C\int_{\sigma-j\infty}^{\sigma+j\infty} (s-A)^{-1}B\mathbf{X}(s)e^{st}\,\frac{\mathrm{d}s}{2\pi j} + D\mathbf{x}(t) =$$
$$= \int_{\sigma-j\infty}^{\sigma+j\infty} \left(C(s-A)^{-1}B + D\right)\mathbf{X}(s)e^{st}\,\frac{\mathrm{d}s}{2\pi j} =$$
$$= \int_{\sigma-j\infty}^{\sigma+j\infty} H(s)\mathbf{X}(s)e^{st}\,\frac{\mathrm{d}s}{2\pi j} \tag{7.15}$$

The latter confirms the fact that $\mathbf{y}_s$ is the steady-state response.

7.8 Diagonal form

We have seen that the transient response part of the signals in the system consists of linear combinations of elements of the matrix e^{At}. The elements of e^{At} can be easily found if A is diagonalized by a similarity transformation. However, instead of diagonalizing the matrix A taken in isolation, it will be more instructive to consider this as diagonalization of the state-space system itself.

According to (7.2a), the matrix A is an operator converting vectors from the state space into vectors in the same space. This, diagonalization of A can be achieved by a specific choice of the state space basis, where the basis vectors must be the eigenvectors of A. After the change of basis we are having exactly the same system, just expressed in different coordinates. In these coordinates the matrix A becomes diagonal and its diagonal elements are eigenvalues of A (which are basis-independent). Now recall that eigenvalues of A are the same as the system poles. Therefore, a sufficient condition for the state-space system to be diagonalizable is that all of its poles are distinct.[2]

Thus, in a diagonalizing basis the elements of A are simply the system poles:

$$A = \begin{pmatrix} p_1 & 0 & \cdots & 0 \\ 0 & p_2 & \cdots & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & \cdots & p_N \end{pmatrix}$$

and the system falls apart into a set of parallel Jordan 1-poles:

$$\dot{u}_n = p_n u_n + \mathbf{b}_n^{\mathsf{T}} \cdot \mathbf{x} \tag{7.16a}$$

$$\mathbf{y} = C\mathbf{u} + D\mathbf{x} = \sum_n \mathbf{c}_n u_n + D\mathbf{x} \tag{7.16b}$$

where $\mathbf{b}_n^{\mathsf{T}}$ are the rows of matrix B (respectively $\mathbf{b}_n^{\mathsf{T}} \cdot \mathbf{x}$ are the input signals of the Jordan 1-poles), and $\mathbf{c}_n$ are the columns of matrix C.

Stability

We already know that the transient response $u_{tn}(t)$ of a 1-pole is an exponent $K_n e^{p_n t}$ (where K_n is the exponent's amplitude). Respectively, the transient response part of $\mathbf{y}$ in (7.16b) is a linear combination of transient responses of the Jordan 1-poles:

$$\mathbf{y}_t = C\mathbf{u}_t = \sum_n \mathbf{c}_n u_{tn} = \sum_n \mathbf{c}_n K_n e^{p_n t}$$

That is, the elements of $\mathbf{y}_t$ are linear combinations of exponents $e^{p_n t}$.

Now recall that $\mathbf{y}$ is independent on the choice of basis and so must be its separation into steady-state and transient response parts. Note that this is in agreement with the fact that according to (7.15) the steady-state response depends only on the transfer matrix and thus is independent of the basis changes. This means that the fact that the elements of $\mathbf{y}_t$ are linear combinations of

[2] A little bit later we will establish the fact that a system where some poles coincide is most likely not diagonalizable.

exponents $e^{p_n t}$ is also independent of basis choice. Respectively $\mathbf{y}_t \to 0$ if and only if $\operatorname{Re} p_n < 0 \ \forall n$. Thus we have obtained the explanation of the stability criterion for linear filters which we introduced in Section 2.9 and have partially shown for lower-order systems.[3]

Transfer matrix

Computing the transfer matrix for the diagonal form we notice that

$$(s - A)^{-1} = \begin{pmatrix} \dfrac{1}{s - p_1} & 0 & \cdots & 0 \\ 0 & \dfrac{1}{s - p_2} & \cdots & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & \cdots & \dfrac{1}{s - p_N} \end{pmatrix} \tag{7.17}$$

that is we have transfer functions of the Jordan 1-poles on the main diagonal. Respectively the main term of the transfer matrix $C(s - A)^{-1}B$ is just a linear combination of the Jordan 1-pole transfer functions. Apparently the common denominator of the terms of this linear combination is

$$\prod_{n=1}^{N} (s - p_n)$$

which is simultaneously the common denominator of the transfer matrix elements.

It can be instructive to explicitly write out the elements of the transfer matrix $H(s)$ in the diagonal case:

$$H_{nm}(s) = \sum_{k=1}^{N} c_{nk} \frac{1}{s - p_k} b_{km} + d_{nm} = \sum_{k=1}^{N} \frac{c_{nk} b_{km}}{s - p_k} + d_{nm} \tag{7.18}$$

that is, we are having a partial fraction expansion of the rational function $H_{nm}(s)$ into fractions of 1st order. Thus, if a transfer matrix is given in advance, there is not much freedom in respect to the choice of the elements of B and C. The poles p_k are prescribed by the common denominator of the transfer matrix and the values of the products $c_{nk}b_{km}$ and of d_{nm} are prescribed by the specific functions $H_{nm}(s)$ occurring in the respective elements of the transfer matrix.

In the single-input single-output case the transfer matrix has 1×1 dimensions, while C has $1 \times N$ and B has $N \times 1$ dimensions respectively. Thus there is only one equation (7.18) and we have N freedom degrees in respect to the choice of c_{nk} and b_{km} giving the required values of the products $c_{nk}b_{km}$. Each such degree can be associated with a variable α_k, where we replace b_{km} with $\alpha_k b_{km}$ and c_{nk} with c_{nk}/α_k. Apparently such replacement doesn't affect the value of the product $c_{nk}b_{km}$. One can also realize that α_k simply scale the levels

[3] Of course, exactly the same results would have been obtained if we simply computed the explicit form of the matrix exponential e^{At} for the diagonal matrix At. However then we would have missed the interpretation of the diagonalizing basis as the basis in which the system can be seen simply as a set of parallel 1-poles.

of the signals u_k, which corresponds to different choices of the lengths of the basis vectors.

Now if we add one more output signal, thereby the dimensions of C becoming $2 \times N$, we can notice that we still have exactly the same N degrees of freedom. If we attempt to change any of b_{km} we need to compensate this in both of c_{nk} for the same k by dividing these c_{nk} by α_k, the latter being the ratio of the new and old values of b_{km}. Respectively, if we change any of c_{nk} in one of the two rows of C, this immediately requires the compensating change of b_{km}, which in turn requires that the same change occurs not just in one but in both rows of C. Adding more rows to C and/or more columns to B we see that the available freedom degrees are still the same and correspond to the freedom of choice of the basis vector lengths.

Thus, aside from the free choice of the basis vector lengths (and of their ordering) the transfer matrix uniquely defines the diagonal form of the state-space system. Respectively, for a non-diagonal form, if the matrix A is given, then the transformation T to the diagonal form is uniquely defined (up to the lengths and the ordering of the basis vectors), and, since the transfer function uniquely defines the matrices B', C' and D' of the diagonal form, the matrices $B = T^{-1}B'$, $C = C'T$ and $D = D'$ are also uniquely defined.

Steady-state response

Apparently, there is the usual freedom in regards to the choice of the steady-state response arising out of evaluating the inverse Laplace transform of $H(s)\mathbf{X}(s)$ to the left or to the right of the poles of $H(s)$. The change of the steady-state response (7.15) depending on the choice of the inverse Laplace transform's integration path in (7.15) to the left or to the right (or in between) the poles of $H(s)$ poses no fundamentally new questions compared to the previous discussion in the analysis of 1- and 2-pole transient responses and results simply in the changes of the amplitudes of transient response partials.

Diagonalization in case of coinciding poles

Even if two or more poles of the system coincide, it still might be diagonalizable, if the eigenvectors corresponding to these poles are distinct. It might seem that this is the most probable situation, after all, what are the changes of two vectors coinciding, or at least being collinear? Without trying to engage ourselves into an analysis of the respective probabilities, we are going to look at this fact from a different angle.

Namely, given a diagonal state space form with some of the eigenvalues coinciding, we are going to have identical entires in the matrix $(s - A)^{-1}$, as one can easily see from (7.17). This means that the order of the common denominator of the elements of $(s - A)^{-1}$ will be less than N and respectively the order of the denominator of the transfer matrix $H(s)$ will also be less than N. This means that the effective order of the system is less than N and the system is degenerate.

Thus, a non-degenerate system with coinciding poles cannot be diagonalized. In such cases we will have to use the Jordan normal form, which we discuss a bit later.

7.9 Real diagonal form

Given a state-space system we could decide to implement it in a diagonal form by first performing a diagonalizing change of basis and then implementing the obtained diagonal state space form. However, if the system has complex poles, the underlying Jordan 1-poles of the system will become complex too, respectively generating complex signals u_n. So, while the system has real input and real output, internally it would need to deal with complex signals. Of course, in a digital world using complex signals internally in a system shouldn't be a big problem. But, for one, this is simply unusual and complicates the implementation structure. More importantly, operations on complex numbers are at least twice as expensible as the same operations on real numbers. We therefore wish to convert a diagonal form containing complex poles to a purely real system, while retaining as much of the diagonalization as possible.

Since the system itself and the matrix A in the original basis are real, the complex poles need to come in conjugate pairs. Without loss of generality we can order the poles in such a way that complex-conjugate pairs come first, followed by purely real poles: p_1, p_1^*, p_3, p_3^*, ..., p_N (where $p_2 = p_1^*$, $p_4 = p_3^*$, etc.) We will refer to the complex poles p_1, p_3, ... as the odd poles and to p_1^*, p_3^*, ... as the even poles. When referring to odd/even poles we will mean only the essentially complex poles, the purely real poles being excluded. Since the poles p_n are eigenvalues of A, we will be referring to even/odd eigenvalues and respectively to even/odd eigenvectors.

Let $\mathbf{v}_1$ be the eigenvector corresponding to p_1, that is $A\mathbf{v}_1 = p_1\mathbf{v}_1$. Then, since A has purely real coefficients, $A\overline{\mathbf{v}_1} = \overline{A\mathbf{v}_1} = \overline{p_1\mathbf{v}_1} = p_1^*\overline{\mathbf{v}_1}$, (where $\overline{\mathbf{v}}$ denotes conjugation of vector's components). Thus $\overline{\mathbf{v}_1}$ is the eigenvector corresponding to p_1^*. Obviously, the same applies to any other even/odd eigenvector. Therefore we can choose a set of eigenvectors such that even eigenvectors are component conjugates of odd eigenvectors: $\mathbf{v}_1$, $\overline{\mathbf{v}_1}$, $\mathbf{v}_3$, $\overline{\mathbf{v}_3}$, ..., $\mathbf{v}_N$.

If $\mathbf{u}' = T\mathbf{u}$ is the diagonalizing transformation of the system, the new basis must consist of eigenvectors of A. Respectively, since $\mathbf{u} = T^{-1}\mathbf{u}'$, the columns of T^{-1} must consist of the new basis vectors, that is of the eigenvectors of A (or, more precisely, consist of coordinates of these eigenvectors in the original basis). We will choose

$$T^{-1} = \begin{pmatrix} \mathbf{v}_1 & \overline{\mathbf{v}_1} & \mathbf{v}_3 & \overline{\mathbf{v}_3} & \dots & \mathbf{v}_N \end{pmatrix}$$

This means that applying component conjugation to T^{-1} swaps the even and the odd columns of T^{-1}, which can be expressed as

$$\overline{T^{-1}} = T^{-1}S$$

where

$$S = \begin{pmatrix} 0 & 1 & 0 & 0 & \cdots & 0 \\ 1 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 1 & \cdots & 0 \\ 0 & 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & 0 & \cdots & 1 \end{pmatrix}$$

is the "swapping matrix". Note that elements of S are purely real and that $S^{-1} = S$.

Since

$$\overline{T} \cdot \overline{T^{-1}} = \overline{TT^{-1}} = 1^* = 1$$

component conjugation and matrix inversion commute:

$$\overline{T}^{\,-1} = \overline{T^{-1}} = T^{-1}S$$

Reciprocating the leftmost and the rightmost expressions we have

$$\overline{T} = \left(T^{-1}S\right)^{-1} = S^{-1}T = ST$$

That is, component conjugation of T swaps its even and odd rows. Or, put in a slightly different way, the even/odd rows of T are component conjugates of each other, and so are the even/odd columns of T^{-1}.

Let's now concentrate on the first conjugate pair of poles. Taking the diagonalized form equations (7.16) we extract those specifically concerning the first two poles:

$$\dot{u}'_1 = p_1 u'_1 + \mathbf{b}'^{\mathsf{T}}_1 \cdot \mathbf{x} \tag{7.19a}$$

$$\dot{u}'_2 = p^*_1 u'_2 + \mathbf{b}'^{\mathsf{T}}_2 \cdot \mathbf{x} \tag{7.19b}$$

$$\mathbf{y} = \mathbf{c}'_1 u'_1 + \mathbf{c}'_2 u'_2 + \sum_{n=3}^{N} \mathbf{c}'_n u'_n + D\mathbf{x} \tag{7.19c}$$

(where we need to employ the prime notation (7.10) for the diagonalized form, since we explicitly used the diagonalizing transformation $\mathbf{u}' = T\mathbf{u}$, thus the non-primed state $\mathbf{u}$ referring to the non-diagonalized form).

Using (7.9) and recalling that the first two rows of T are component conjugates of each other, we must conclude that so are the first two rows of B', that is $\mathbf{b}'^{\mathsf{T}}_2 = \overline{\mathbf{b}'^{\mathsf{T}}_1}$. Recalling that the first two colums of T^{-1} are component conjugates of each other, we conclude that $\mathbf{c}'_2 = \overline{\mathbf{c}'_1}$. On the other hand, writing out the first two rows of (7.13) in the diagonal case we have

$$u'_1(t) = e^{p_1 t} u'_1(0) + \int_0^t e^{p_1(t-\tau)} \mathbf{b}'^{\mathsf{T}}_1 \mathbf{x}\, \mathrm{d}t$$

$$u'_2(t) = e^{p^*_1 t} u'_2(0) + \int_0^t e^{p^*_1(t-\tau)} \overline{\mathbf{b}'^{\mathsf{T}}_1} \mathbf{x}\, \mathrm{d}t$$

Except for the initial state term, the right-hand side of the second equation is a complex conjugate of the right-hand side of the first one. Regarding the initial state term, practically seen, we would have the following situations

- the initial state would be either zero, in which case $u'_2(t) = u'^*_1(t)$,
- or it would be a result of some previous signal processing by the system, where previously to that processing the initial state would be zero, in which case $u'_2(0) = u'^*_1(0)$ and respectively $u'_2(t) = u'^*_1(t)$.

Therefore, we can simply require that $u'_2(0) = u'^*_1(0)$, and thus the output signals $u'_1(t)$ and $u'_2(t)$ of the first two Jordan 1-poles are mutually conjugate. Respectively, the contribution of $u'_1(t)$ and $u'_2(t)$ into $\mathbf{y}$ in (7.19c), being equal to $\mathbf{c}'_1 u'_1 + \mathbf{c}'_2 u'_2$, turns out to be a sum of two conjugate values and is therefore

purely real. Obviously, the same applies to all other complex conjugate pole pairs.

Thus, even equations of (7.16a) do not contribute any new information about the system and we could drop them, simply computing even state signals as conjugates of odd state signals: $u'_2 = u'^*_1$, $u'_4 = u'^*_3$, etc. At the same time we could rewrite the odd equations of (7.16a) explicitly using real and imaginary parts of the signals u':

$$\frac{\mathrm{d}}{\mathrm{d}t}\operatorname{Re} u'_n = \operatorname{Re} p_n \operatorname{Re} u'_n - \operatorname{Im} p_n \operatorname{Im} u'_n + \left(\operatorname{Re} \mathbf{b}'^{\mathsf{T}}_n\right) \cdot \mathbf{x} \tag{7.20a}$$

$$\frac{\mathrm{d}}{\mathrm{d}t}\operatorname{Im} u'_n = \operatorname{Im} p_n \operatorname{Re} u'_n + \operatorname{Re} p_n \operatorname{Im} u'_n + \left(\operatorname{Im} \mathbf{b}'^{\mathsf{T}}_n\right) \cdot \mathbf{x} \tag{7.20b}$$

Therefore we can introduce the new state variables, taking purely real values:

$$\left.\begin{aligned} u''_n &= \operatorname{Re} u'_n = \frac{u'_n + u'_{n+1}}{2} \\ u''_{n+1} &= \operatorname{Im} u'_n = \frac{u'_n - u'_{n+1}}{2j} \end{aligned}\right\} \qquad \text{for odd } p_n \tag{7.21a}$$

and

$$u''_n = u'_n \qquad \text{for purely real } p_n \tag{7.21b}$$

Then (7.20) turn into

$$\dot{u}''_n = \operatorname{Re} p_n \cdot u''_n - \operatorname{Im} p_n \cdot u''_{n+1} + \left(\operatorname{Re} \mathbf{b}'^{\mathsf{T}}_n\right) \cdot \mathbf{x} \tag{7.22a}$$

$$\dot{u}''_{n+1} = \operatorname{Im} p_n \cdot u''_n + \operatorname{Re} p_n \cdot u''_{n+1} + \left(\operatorname{Im} \mathbf{b}'^{\mathsf{T}}_n\right) \cdot \mathbf{x} \tag{7.22b}$$

and the respective terms in (7.19c) turn into

$$\begin{aligned} \mathbf{c}'_n u'_n + \mathbf{c}'_{n+1} u'_{n+1} = \mathbf{c}'_n u'_n + \mathbf{c}'^*_n u'^*_n = \mathbf{c}'_n u'_n + (\mathbf{c}'_n u'_n)^* = 2 \operatorname{Re}(c'_n u'_n) = \\ = 2 \operatorname{Re} c'_n \cdot u''_n - 2 \operatorname{Im} c'_n \cdot u''_{n+1} \end{aligned}$$

Thus we have obtained a purely real system

$$\dot{\mathbf{u}}'' = \begin{pmatrix} \operatorname{Re} p_1 & -\operatorname{Im} p_1 & 0 & 0 & \cdots & 0 \\ \operatorname{Im} p_1 & \operatorname{Re} p_1 & 0 & 0 & \cdots & 0 \\ 0 & 0 & \operatorname{Re} p_3 & -\operatorname{Im} p_3 & \cdots & 0 \\ 0 & 0 & \operatorname{Im} p_3 & \operatorname{Re} p_3 & \cdots & 0 \\ 0 & 0 & 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & 0 & \cdots & p_N \end{pmatrix} \mathbf{u}'' + \begin{pmatrix} \operatorname{Re} \mathbf{b}'^{\mathsf{T}}_1 \\ \operatorname{Im} \mathbf{b}'^{\mathsf{T}}_1 \\ \operatorname{Re} \mathbf{b}'^{\mathsf{T}}_3 \\ \operatorname{Im} \mathbf{b}'^{\mathsf{T}}_3 \\ \vdots \\ \mathbf{b}'^{\mathsf{T}}_N \end{pmatrix} \mathbf{x} \tag{7.23a}$$

$$\mathbf{y} = \begin{pmatrix} 2\operatorname{Re} c'_1 & -2\operatorname{Im} c'_n & 2\operatorname{Re} c'_3 & -2\operatorname{Im} c'_3 & \cdots & c'_N \end{pmatrix} \mathbf{u}'' + D\mathbf{x} \tag{7.23b}$$

We will refer to (7.23) as the *real diagonal form*. It represents the system as a set of parallel 2-poles (7.22) (and optionally additional parallel 1-poles if some of the system poles are real).

Note that the substitutions (7.21) are expressible as another linear transfor-

mation $\mathbf{u}'' = T'\mathbf{u}'$ where

$$T' = \begin{pmatrix} \frac{1}{2} & \frac{1}{2} & 0 & 0 & \cdots & 0 \\ \frac{1}{2j} & -\frac{1}{2j} & 0 & 0 & \cdots & 0 \\ 0 & 0 & \frac{1}{2} & \frac{1}{2} & \cdots & 0 \\ 0 & 0 & \frac{1}{2j} & -\frac{1}{2j} & \cdots & 0 \\ 0 & 0 & 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & 0 & \cdots & 1 \end{pmatrix}$$

Therefore the real diagonal form of the system is related to the original form by a change of basis, where the respective transformation matrix is $T'\,T$.

Jordan 2-poles

The 2-poles (7.22) in the real diagonal form are fully analogous to the 1-poles occuring in the diagonal form. They will also occur in the real Jordan normal form. For that reason we will refer to them as *Jordan 2-poles*. They are also sometimes (especially in their discrete-time counterpart form) referred to as *coupled-form resonators*.

The key feature of the Jordan 2-pole topology is that in the absence of the input signal, the system state is spiralling in a circle of an exponentially decaying (or growing) radius. Indeed, recalling that equations (7.22) are simply separate equations for the real and imaginary components of a complex signal u'_n, we can return to using the equation (7.19a), which by letting $\mathbf{x} = 0$ and turns into

$$\dot{u} = pu$$

where we also dropped the indices and the prime notation for simplicity. Respectively

$$\frac{\mathrm{d}}{\mathrm{d}t} \log u = \frac{\dot{u}}{u} = p = \operatorname{Re} p + j \operatorname{Im} p \tag{7.24}$$

On the other hand

$$\log u = \ln |u| + j \arg u$$

therefore

$$\frac{\mathrm{d}}{\mathrm{d}t} \log u = \frac{\mathrm{d}}{\mathrm{d}t} \ln |u| + j \frac{\mathrm{d}}{\mathrm{d}t} \arg u \tag{7.25}$$

Equating the right-hand sides of (7.24) and (7.25), we obtain

$$\frac{\mathrm{d}}{\mathrm{d}t} \ln |u| + j \frac{\mathrm{d}}{\mathrm{d}t} \arg u = \operatorname{Re} p + j \operatorname{Im} p$$

or

$$\frac{\mathrm{d}}{\mathrm{d}t} \ln |u| = \operatorname{Re} p$$
$$\frac{\mathrm{d}}{\mathrm{d}t} \arg u = \operatorname{Im} p$$

from where

$$\ln |u(t)| = \ln |u(0)| + \operatorname{Re} p \cdot t$$

$$\arg u(t) = \arg u(0) + \operatorname{Im} p \cdot t$$

or

$$\begin{aligned} |u(t)| &= |u(0)| \cdot e^{t \operatorname{Re} p} \\ \arg u(t) &= \arg u(0) + \operatorname{Im} p \cdot t \end{aligned}$$

Thus the complex value $u(t)$ is rotating around the origin with the angular speed $\operatorname{Im} p$, it's distance from the origin changing as $e^{t \operatorname{Re} p}$, thereby moving in a decaying spiral if $\operatorname{Re} p < 0$, an expanding spiral if $\operatorname{Re} p > 0$, or a circle if $\operatorname{Re} p = 0$. Recalling that the state components of (7.22) are simply the real and imaginary parts of u in the above equations, we conclude that the state of (7.22) in the absence of the input signal is moving in the same spiral trajectory.

Notably, the separation of u into real and imaginary parts works only if the pole is complex.[4]

Transfer matrix

In order to obtain the transfer matrix of the real diagonal form we could first obtain the transfer matrices of the individual 2-poles (7.22). Concentrating on a single 2-pole, we write (7.22) as

$$\begin{aligned} \dot{u}_1 &= \operatorname{Re} p \cdot u_1 - \operatorname{Im} p \cdot u_2 + x_1 \\ \dot{u}_2 &= \operatorname{Im} p \cdot u_1 + \operatorname{Re} p \cdot u_2 + x_2 \end{aligned}$$

where we ignored the input mixing coefficients B (in principle we can understand this form in the sense that the input signals are picked up past the mixing coefficients B, or as a particular case of B being identity matrix). We could explicitly compute the matrix $(s-A)^{-1}$ for the above system, or we could derive it "manually", which is what we're going to do.

Given $x_1 = X_1(s)e^{st}$, $x_2 = X_2(s)e^{st}$ we have

$$\begin{aligned} sU_1(s)e^{st} &= \operatorname{Re} p \cdot U_1(s)e^{st} - \operatorname{Im} p \cdot U_2(s)e^{st} + X_1(s)e^{st} \\ sU_2(s)e^{st} &= \operatorname{Im} p \cdot U_1(s)e^{st} + \operatorname{Re} p \cdot U_2(s)e^{st} + X_2(s)e^{st} \end{aligned}$$

Respectively

$$\begin{aligned} (s - \operatorname{Re} p)U_1(s) + \operatorname{Im} p \cdot U_2(s) &= X_1(s) \\ -\operatorname{Im} p \cdot U_1(s) + (s - \operatorname{Re} p)U_2(s) &= X_2(s) \end{aligned}$$

Attempting to eliminate $U_1(s)$, we multiply each equation by a different factor:

$$\begin{aligned} (s - \operatorname{Re} p)\operatorname{Im} p \cdot U_1(s) + (\operatorname{Im} p)^2 \cdot U_2(s) &= \operatorname{Im} p \cdot X_1(s) \\ -(s - \operatorname{Re} p)\operatorname{Im} p \cdot U_1(s) + (s - \operatorname{Re} p)^2 U_2(s) &= (s - \operatorname{Re} p)X_2(s) \end{aligned}$$

and add both equations together:

$$\left((s - \operatorname{Re} p)^2 + (\operatorname{Im} p)^2\right) U_2(s) = \operatorname{Im} p \cdot X_1(s) + (s - \operatorname{Re} p)X_2(s)$$

[4] This is strongly related to the appearance of Jordan normal form at the moment when two complex conjugate poles coincide on the real axis.

Respectively attempting to eliminate $U_2(s)$, we multiply each equation by a different factor:

$$\begin{aligned}(s-\operatorname{Re}p)^2U_1(s)\ &+(s-\operatorname{Re}p)\operatorname{Im}p\cdot U_2(s)=(s-\operatorname{Re}p)X_1(s)\\ -(\operatorname{Im}p)^2\cdot U_1(s)&+(s-\operatorname{Re}p)\operatorname{Im}p\cdot U_2(s)=\operatorname{Im}p\cdot X_2(s)\end{aligned}$$

and subtract the second equation from the first one:

$$\big((s-\operatorname{Re}p)^2+(\operatorname{Im}p)^2\big)\,U_1(s)=(s-\operatorname{Re}p)X_1(s)-\operatorname{Im}p\cdot X_2(s)$$

Thus

$$\begin{aligned}\begin{pmatrix}U_1(s)\\U_2(s)\end{pmatrix}&=\frac{1}{(s-\operatorname{Re}p)^2+(\operatorname{Im}p)^2}\begin{pmatrix}s-\operatorname{Re}p & -\operatorname{Im}p\\ \operatorname{Im}p & s-\operatorname{Re}p\end{pmatrix}\begin{pmatrix}X_1(s)\\X_2(s)\end{pmatrix}=\\&=\frac{1}{s^2-2\operatorname{Re}p\cdot s+|p|^2}\begin{pmatrix}s-\operatorname{Re}p & -\operatorname{Im}p\\ \operatorname{Im}p & s-\operatorname{Re}p\end{pmatrix}\begin{pmatrix}X_1(s)\\X_2(s)\end{pmatrix}\end{aligned}$$

and, since for this system the matrix B is identity matrix,

$$(s-A)^{-1}=\frac{1}{s^2-2\operatorname{Re}p\cdot s+|p|^2}\begin{pmatrix}s-\operatorname{Re}p & -\operatorname{Im}p\\ \operatorname{Im}p & s-\operatorname{Re}p\end{pmatrix}\tag{7.26}$$

Note that the denominator is the standard 2-pole filter's transfer function denominator, written in terms of the pole. Indeed, the complex conjugate poles p and p^* of two complex Jordan 1-poles were combined into a Jordan 2-pole by means of a linear combination. Respectively the Jordan 2-pole has exactly the same poles.

Generalizing the result obtained in (7.26) to systems of arbitrary order, containing multiple parallel 2-poles, we conclude that the main diagonal of $(s-A)^{-1}$ contains the matrices of the form

$$G(s)=\frac{1}{s^2-2\operatorname{Re}p\cdot s+|p|^2}\begin{pmatrix}s-\operatorname{Re}p & -\operatorname{Im}p\\ \operatorname{Im}p & s-\operatorname{Re}p\end{pmatrix}\tag{7.27}$$

similarly to how the transfer functions $1/(s-p_n)$ of the Jordan 1-poles are occurring on the main diagonal of $(s-A)^{-1}$ in (7.17). Thus $(s-A)^{-1}$ has the form

$$(s-A)^{-1}=\begin{pmatrix}G_1(s) & 0 & \cdots & 0\\ 0 & G_2(s) & \cdots & 0\\ 0 & 0 & \ddots & 0\\ 0 & 0 & \cdots & \dfrac{1}{s-p_N}\end{pmatrix}$$

where $G(s)$ have the form (7.27).

Similarly to what we did in the diagonal case, in the real diagonal case we also would like to explicitly write out the elements of the transfer matrix $H(s)$. For the sake of notation simplicity we will write them out for the case of a 2×2 matrix A. First, let's notice that

$$\begin{pmatrix}\mathbf{c}_1 & \mathbf{c}_2\end{pmatrix}\begin{pmatrix}\rho_{11} & \rho_{12}\\ \rho_{21} & \rho_{22}\end{pmatrix}\begin{pmatrix}\mathbf{b}_1^{\mathsf{T}}\\ \mathbf{b}_2^{\mathsf{T}}\end{pmatrix}=\begin{pmatrix}\mathbf{c}_1 & \mathbf{c}_2\end{pmatrix}\begin{pmatrix}\rho_{11}\mathbf{b}_1^{\mathsf{T}}+\rho_{12}\mathbf{b}_2^{\mathsf{T}}\\ \rho_{21}\mathbf{b}_1^{\mathsf{T}}+\rho_{22}\mathbf{b}_2^{\mathsf{T}}\end{pmatrix}=$$

$$= \left(\mathbf{c}_1\rho_{11}\mathbf{b}_1^\mathsf{T} + \mathbf{c}_1\rho_{12}\mathbf{b}_2^\mathsf{T} + \mathbf{c}_2\rho_{21}\mathbf{b}_1^\mathsf{T} + \mathbf{c}_2\rho_{22}\mathbf{b}_2^\mathsf{T}\right) =$$
$$= \left(\sum_{k,l=1}^{2} \rho_{kl}\mathbf{c}_k\mathbf{b}_l^\mathsf{T}\right)$$

where ρ_{nm} are the elements of $(s-A)^{-1}$ and where $\mathbf{c}_n\mathbf{b}_m^\mathsf{T}$ denotes the outer product of the n-th column of C by the m-th row of B. Then, for a 2×2 real diagonal system we obtain:

$$\begin{aligned} H_{nm}(s) &= \sum_{k,l=1}^{2} \rho_{kl}c_{nk}b_{lm} + d_{nm} = \\ &= \frac{(c_{n1}b_{1m} + c_{n2}b_{2m})(s - \operatorname{Re} p) + (c_{n2}b_{1m} - c_{n1}b_{2m})\operatorname{Im} p}{s^2 - 2\operatorname{Re} p \cdot s + |p|^2} + d_{nm} = \\ &= \frac{\alpha_{nm}s + \beta_{nm}}{s^2 - 2\operatorname{Re} p \cdot s + |p|^2} + d_{nm} \end{aligned}$$

where α_{nm} and β_{nm} are obtained by summing the respective products of the elements of b and c. Respectively, for higher-order systems we have

$$H_{nm}(s) = \sum_{\operatorname{Im} p_k > 0} \frac{\alpha_{nmk}s + \beta_{nmk}}{s^2 - 2\operatorname{Re} p_k \cdot s + |p_k|^2} + \sum_{\operatorname{Im} p_k = 0} \frac{c_{nk}b_{km}}{s - p_k} + d_{nm} \qquad (7.28)$$

Since real diagonal form is nothing more than a linear transformation of the diagonal form, there are the same freedom degrees in respect to the choice of the coefficients of B and C matrices, corresponding to choosing the basis vectors of different lengths.

7.10 Jordan normal form

We have shown that if a non-degenerate system has coinciding poles, it is not diagonalizable. The generalization of the diagonalization idea, which also works in this case, is *Jordan normal form*. The process of diagonalization implies that there is a similarity transformation of the matrix which brings the matrix into a diagonal form. Such transformation might not exist. However, there is always a similarity transformation bringing the matrix into the Jordan normal form.

The building element of a matrix in the Jordan normal form is a *Jordan cell*. A Jordan cell is a matrix having the form

$$J_n = \begin{pmatrix} p_n & 0 & 0 & \cdots & 0 & 0 \\ 1 & p_n & 0 & \cdots & 0 & 0 \\ 0 & 1 & p_n & \cdots & 0 & 0 \\ 0 & 0 & \ddots & \ddots & \vdots & 0 \\ 0 & 0 & 0 & \ddots & p_n & 0 \\ 0 & 0 & 0 & \cdots & 1 & p_n \end{pmatrix} \qquad (7.29)$$

That is it contains one and the same eigenvalue p_n all over its main diagonal, and it contains 1's on the subdiagonal right below its main diagonal, all other

elements being equal to zero.[5] Respectively, a matrix in the Jordan normal form consists of Jordan cells on its main diagonal:

$$A = \begin{pmatrix} J_1 & 0 & \cdots & 0 \\ 0 & J_2 & \cdots & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & \cdots & J_M \end{pmatrix}$$

(where M is the number of different Jordan cells), all other entries in the matrix being equal to zero.

Apparently the sizes of all Jordan cells should sum up to the dimension of the matrix A. The total number of times an eigenvalue appears on the main diagonal of A is equal to the multiplicity of the eigenvalue. Typically there would be a single Jordan cell corresponding to a given eigenvalue. Thus, if an eigenvalue has a multiplicity of 5, typically there would be a single Jordan cell of size 5×5 containing that eigenvalue. It is also possible that there are several Jordan cells corresponding to the same eigenvalue, e.g. given an eigenvalue of a multiplicity of 5, there could be a 2×2 and a 3×3 Jordan cell containing that eigenvalue. If there are several Jordan cells for a given eigenvalue, the respective state-space system is degenerate, fully similar to the case of repeated poles in the diagonalized case.

It is easy to notice that, compared to the diagonal form, Jordan cells appear on the main diagonal instead of eigenvalues. A Jordan cell may have a 1×1 size, in which case it is identical to an eigenvalue appearing on the main diagonal. If all Jordan cells have 1×1 size Jordan normal form turns into diagonal form.

Similarly to diagonal form being unique up to the order of eigenvalues, the Jordan normal form is unique up to the order of Jordan cells. That is, the number and the sizes of Jordan cells corresponding to a given pole is a property of the original matrix A. The process of finding the similarity transformation converting a matrix into Jordan normal form is not much different from the diagonalization process: we need to find a basis in which the matrix takes Jordan normal form, which immediately implies a set of equations for such basis vectors. More details can be found outside of this book.

Jordan chains

It's not difficult to realize that a Jordan cell corresponds to a series of Jordan 1-poles, which we introduced in Section 2.15 under the name of a *Jordan chain*. So, now we should be able to understand the reason for that name.

Indeed, suppose A is in Jordnal normal form and suppose there is a Jordan cell of size N_1 located at the top of the main diagonal of A. Then, writing out the first N_1 rows of (7.2) we have

$$\begin{aligned} \dot{u}_1 &= p_1 u_1 + \mathbf{b}_1^{\mathsf{T}} \cdot \mathbf{x} \\ \dot{u}_2 &= p_1 u_2 + \left(u_1 + \mathbf{b}_2^{\mathsf{T}} \cdot \mathbf{x}\right) \\ \dot{u}_3 &= p_1 u_3 + \left(u_2 + \mathbf{b}_3^{\mathsf{T}} \cdot \mathbf{x}\right) \\ &\cdots \end{aligned}$$

[5] Some texts place 1's above the main diagonal. This is simply a matter of convention. One can convert from one version to the other by simply reindexing the basis vectors.

$$\dot{u}_{N_1} = p_1 u_{N_1} + \left(u_{N_1-1} + \mathbf{b}_{N_1}^{\mathsf{T}} \cdot \mathbf{x}\right)$$

Note that except for the first line, the input signal of the respective 1-pole contains the output of the previous 1-pole. In Fig. 2.24 we had a single-input single-output Jordan chain, now we are having a multi-input multi-output one (Fig. 7.2).

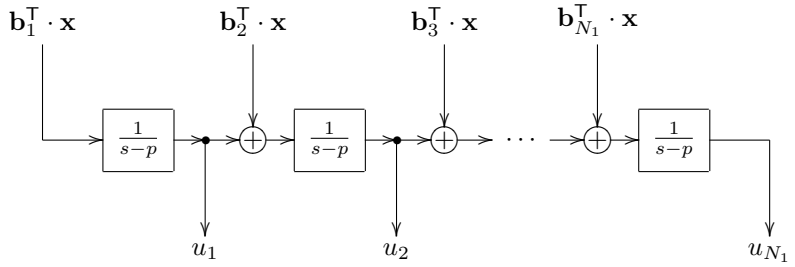


Figure 7.2: Multi-input multi-output Jordan chain

Transfer matrix

In the diagonal case the transfer matrix had a diagonal form (7.17) corresponding to the fact that the diagonal form is just a set of parallel Jordan 1-poles. Now we need to replace these 1-poles with Jordan chains. Thus, instead of single values $1/(s-p_n)$ on the main diagonal, the transfer matrix will have submatrices of the size of respective Jordan cells. From Fig. 7.2 it's not difficult to realize that a transfer submatrix corresponding to a Jordan cell of the form (7.29) will have the form

$$\begin{pmatrix}
\frac{1}{s-p_n} & 0 & 0 & \cdots & 0 & 0 \\
\frac{1}{(s-p_n)^2} & \frac{1}{s-p_n} & 0 & \cdots & 0 & 0 \\
\frac{1}{(s-p_n)^3} & \frac{1}{(s-p_n)^2} & \frac{1}{s-p_n} & \cdots & 0 & 0 \\
\vdots & \vdots & \ddots & \ddots & \vdots & 0 \\
\frac{1}{(s-p_n)^{N_1-1}} & \frac{1}{(s-p_n)^{N_1-2}} & \frac{1}{(s-p_n)^{N_1-3}} & \ddots & \frac{1}{s-p_n} & 0 \\
\frac{1}{(s-p_n)^{N_1}} & \frac{1}{(s-p_n)^{N_1-1}} & \frac{1}{(s-p_n)^{N_1-2}} & \cdots & \frac{1}{(s-p_n)^2} & \frac{1}{s-p_n}
\end{pmatrix}$$

Transient response

According to (7.13), the elements of the matrix e^{At} are the exponent terms in $\mathbf{u}(t)$ which have the amplitudes $u_n(0)$. Apparently, being a part of the transient response, these terms do not explicitly depend on the system input signal and thus are the same in the single-input single-output and multiple-input multiple-output cases. Comparing to the explicit expression (2.25) for the output signal of a single-input single-output Jordan chain, we realize the following.

The elements of e^{At} are $t^\nu e^{p_n t}/\nu!$. These elements are organized into submatrices of e^{At} corresponding to Jordan cells of A. Each such submatrix has the following form:[6]

$$\begin{pmatrix} 1 & 0 & 0 & \cdots & 0 & 0 \\ t & 1 & 0 & \cdots & 0 & 0 \\ \frac{t^2}{2} & t & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots & 0 \\ \frac{t^{N_1-2}}{(N_1-2)!} & \frac{t^{N_1-3}}{(N_1-3)!} & \frac{t^{N_1-4}}{(N_1-4)!} & \ddots & 1 & 0 \\ \frac{t^{N_1-1}}{(N_1-1)!} & \frac{t^{N_1-2}}{(N_1-2)!} & \frac{t^{N_1-3}}{(N_1-3)!} & \cdots & t & 1 \end{pmatrix} \cdot e^{p_n t}$$

This confirms that the stability criterion $\operatorname{Re} p_n < 0 \ \forall n$ stays the same even if the system is not diagonalizable.

Real Jordan normal form

If the system has pairs of mutually conjugate poles, the Jordan cells for these poles will also come in conjugate pairs. Following the same steps as for diagonal form, we can introduce new state variables for the real and imaginary parts of complex state signals. Respectively, we each pair of conjugate Jordan cells will be converted to a purely real cell of double size. We will refer to such cells as *real Jordan cells*.

In order to understand how a real Jordan cell looks like, we can recall the interpretation of Jordan cells as Jordan chains (Fig. 7.2). Let's imagine that the signals passing through this chain are complex. This can be equivalently represented as passing real and imaginary parts of these signals separately. Respectively, an element of a real Jordan chain must simply forward the real and imaginary parts of its output signal to the real and imaginary inputs of the next element. E.g. for a pair of conjugate 2nd-order Jordan cells

$$\begin{pmatrix} p & 0 & 0 & 0 \\ 1 & p & 0 & 0 \\ 0 & 0 & p^* & 0 \\ 0 & 0 & 1 & p^* \end{pmatrix}$$

the corresponding real Jordan cell would be

$$\begin{pmatrix} \operatorname{Re} p & -\operatorname{Im} p & 0 & 0 \\ \operatorname{Im} p & \operatorname{Re} p & 0 & 0 \\ 1 & 0 & \operatorname{Re} p & -\operatorname{Im} p \\ 0 & 1 & \operatorname{Im} p & \operatorname{Re} p \end{pmatrix}$$

7.11 Ill-conditioning of diagonal form

Suppose we are having a system where all poles are distinct, which is therefore diagonalizable. And suppose, as a matter of a thought experiment, we begin

[6] The explicit form of an exponent of a Jordan normal form matrix can also be obtained directly from (7.11), but that approach is more involved and we won't do it here.

to modify the system parameters in a continuous way, simultaneously keeping track of the diagonal form of this system. We also keep track of the similarity transformation matrix T defined by $\mathbf{u}' = T\mathbf{u}$, where $\mathbf{u}$ is the original state and $\mathbf{u}'$ is the "diagonalized" state. Note, that by this experiment we don't mean that we are varying the system parameters in respect to time, rather we consider it as looking at different systems with different parameter values.

Suppose, we modify the system parameters in such a way, that some poles of the system get close to each other and finally coincide. Assuming the system order doesn't degenerate, at this point we should switch from a diagonal matrix A' to a Jordan normal form matrix A'. The difference between these two matices is clearly non-zero, thus there is a sudden jump in the components of matrix A' at the moment of the switching. Respectively, there is a jump in the components of T as well. We wish to analyse more closely, what's happening in this case.

If two eigenvalues of a matrix become close then the respective eigenvectors might either also get close to each other or not. If they don't, the eigenspace retains the full dimension as the poles coincide, respectively the system is diagonalizable and the system order degenerates. Thus, if the order of the system doesn't degenerate, the eigenvectors corresponding to closely located eigenvalues must get close to each other too. Note that by saying that the eigenvectors are getting close to each other we mean that they are becoming almost collinear. Apparently, eigenvectors simply having different lengths but the same (or the opposite) directions don't count as different eigenvectors.

Let's pick a pair of such eigenvectors which are getting close to each other. Without loss of generality we may denote these two eigenvectors as $\mathbf{v}_1$ and $\mathbf{v}_2$. In order to simplify the discussion, we will first assume that both eigenvectors are normalized: $|\mathbf{v}_1| = |\mathbf{v}_2| = 1$ (where here and further the lengths will be defined in terms of the original basis, that is we are treating the original basis as an orthonormal one). Again, without loss of generality we may assume that $\mathbf{v}_1$ and $\mathbf{v}_2$ are pointing in (almost) the same direction.

Suppose we have a state vector $\mathbf{u}$ lying fully in the two-dimensional subspace spanned by $\mathbf{v}_1$ and $\mathbf{v}_2$. Therefore its coordinate expansion in the diagonalizing basis is a linear combination of $\mathbf{v}_1$ and $\mathbf{v}_2$, the other coordinates being zeros:

$$\mathbf{u} = \alpha_1\mathbf{v}_1 + \alpha_2\mathbf{v}_2$$

We are going to show that α_1 and α_2 are not well defined.

Let's introduce two other unit-length vectors into the same two-dimensional subspace:

$$\mathbf{v}_+ = \frac{\mathbf{v}_1 + \mathbf{v}_2}{|\mathbf{v}_1 + \mathbf{v}_2|}$$
$$\mathbf{v}_- = \frac{\mathbf{v}_1 - \mathbf{v}_2}{|\mathbf{v}_1 - \mathbf{v}_2|}$$

Apparently, $\mathbf{v}_+$ and $\mathbf{v}_-$ are orthogonal to each other and we could expand $\mathbf{u}$ in terms of $\mathbf{v}_+$ and $\mathbf{v}_-$:

$$\mathbf{u} = \alpha_+\mathbf{v}_+ + \alpha_-\mathbf{v}_-$$

such expansion being well-defined, since the basis $\mathbf{v}_+$, $\mathbf{v}_-$ is orthonormal.

Now we wish to express α_1 and α_2 via α_+ and α_-:

$$\mathbf{u} = \alpha_+\mathbf{v}_+ + \alpha_-\mathbf{v}_- = \alpha_+\frac{\mathbf{v}_1 + \mathbf{v}_2}{|\mathbf{v}_1 + \mathbf{v}_2|} + \alpha_-\frac{\mathbf{v}_1 - \mathbf{v}_2}{|\mathbf{v}_1 - \mathbf{v}_2|} =$$

$$= \left(\frac{\alpha_+}{|\mathbf{v}_1 + \mathbf{v}_2|} + \frac{\alpha_-}{|\mathbf{v}_1 - \mathbf{v}_2|} \right) \mathbf{v}_1 + \left(\frac{\alpha_+}{|\mathbf{v}_1 + \mathbf{v}_2|} - \frac{\alpha_-}{|\mathbf{v}_1 - \mathbf{v}_2|} \right) \mathbf{v}_2$$

from where

$$\alpha_1 = \frac{\alpha_+}{|\mathbf{v}_1 + \mathbf{v}_2|} + \frac{\alpha_-}{|\mathbf{v}_1 - \mathbf{v}_2|}$$
$$\alpha_2 = \frac{\alpha_+}{|\mathbf{v}_1 + \mathbf{v}_2|} - \frac{\alpha_-}{|\mathbf{v}_1 - \mathbf{v}_2|}$$

Since α_+ and α_- are coordinates in an orthonormal basis, both α_+ and α_- are taking values of comparable orders of magnitude, bounded by the length of the vector $\mathbf{u}$. On the other hand, since $|\mathbf{v}_1 - \mathbf{v}_2| \approx 0$, the values of α_1 and α_2 will get extremely large, unless α_- is very small.

Now consider a conversion from the basis $\mathbf{v}_1$, $\mathbf{v}_2$ to a more "decent" basis, e.g. to $\mathbf{v}_+$, $\mathbf{v}_-$. Expressing $\mathbf{v}_1$, $\mathbf{v}_2$ via $\mathbf{v}_+$, $\mathbf{v}_-$, we have

$$\mathbf{v}_1 = \beta_+ \mathbf{v}_+ + \beta_- \mathbf{v}_-$$
$$\mathbf{v}_2 = \beta_+ \mathbf{v}_+ - \beta_- \mathbf{v}_-$$

where $\beta_+ \approx 1$ and $\beta_- \approx 0$. Therefore

$$\mathbf{u} = \alpha_1 \mathbf{v}_1 + \alpha_2 \mathbf{v}_2 = \alpha_1 (\beta_+ \mathbf{v}_+ + \beta_- \mathbf{v}_-) + \alpha_2 (\beta_+ \mathbf{v}_+ - \beta_- \mathbf{v}_-) =$$
$$= (\alpha_1 + \alpha_2)\beta_+ \cdot \mathbf{v}_+ + (\alpha_1 - \alpha_2)\beta_- \cdot \mathbf{v}_- = \alpha_+ \mathbf{v}_+ + \alpha_- \mathbf{v}_-$$

As we have noted, usually α_1 and α_2 are having very large magnitudes, while $\alpha_+^2 + \alpha_-^2 \leq |\mathbf{u}|$. This means that usually α_1 and α_2 are having opposite signs, in order to have $|(\alpha_1 + \alpha_2)\beta_+| < 1$, since $\beta_+ \approx 1$. Respectively their difference $\alpha_1 - \alpha_2$ is usually having a very large magnitude which is being compensated by the multiplication by $\beta_- = 0$.

Thus, the problematic equation is

$$\alpha_+ = (\alpha_1 + \alpha_2)\beta_+ \approx \alpha_1 + \alpha_2$$

where we add two very large numbers of opposite sign in order to obtain a value of α_+ of a reasonable magnitude. Such computations are associated with large numeric precision losses. Choosing different lengths for $\mathbf{v}_1$ and $\mathbf{v}_2$ will not change the picture, we still will need to obtain α_+ as the sum of the same opposite values of a much larger magnitude.

A conversion from the basis $\mathbf{v}_1$, $\mathbf{v}_2$ to a "decent" basis other than $\mathbf{v}_+$, $\mathbf{v}_-$ can be viewed as converting first to $\mathbf{v}_+$, $\mathbf{v}_-$ and then to the desired basis. Apparently, converting from one "decent" basis to another "decent" one neither introduces new precision-related issues, nor removes the already existing ones.

Now realize, that essentially we have just been analysing the precision issues arising in the transformations from the original to the diagonalizing basis and back. It's just that we have restricted the analysis to a particular subspace of the state space, but the transformation which we have been analysing was a diagonalizing transformation of the entire space. We have therefore determined that there are range and precision issues arising in the diagonalizing transformation when two eigenvectors become close to each other. We have also found out that this situation always occurs in non-degenerate cases of poles getting

close to each other. Thus, diagonal form becomes ill-conditioned if the poles are located close to each other, the effects of ill-conditioning being huge precision losses and the values possibly going out of range. Jordan cells of size larger than 1 are nothing more than a limiting case of this ill-conditioned situation, where a different choice of basis avoids the precision issues.

The reader may also recall at this point the ill-conditioning in the analysis of the transient response of the 2-pole filters, which occurs at $R \approx 1$, when both poles of the system coincide on the real axis. That was exactly the same effect as the one which we analysed in this section.

7.12 Time-varying case

Until now we have been assuming that the system coefficients are not changing. If the system coefficients are varying with time, then quite a few of the previously derived statements do not hold anymore. This also causes problems with some of the techniques. The fact that the transfer function doesn't apply in the time-varying case should be well-known by now, however the other issues arising out of parameter variation are not that obvious. Let's look through them one by one.

Basis change

If the matrix A is varying with time, we might need T to vary with time as well, e.g. if T is a matrix of the diagonalizing transformation. However, if T is not constant anymore, the transformations of (7.8) get a more complicated form, since instead of

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(T^{-1}\mathbf{u}'\right) = T^{-1}\dot{\mathbf{u}}'$$

we are having

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(T^{-1}\mathbf{u}'\right) = T^{-1}\dot{\mathbf{u}}' + \frac{\mathrm{d}}{\mathrm{d}t}T^{-1}\cdot\mathbf{u}'$$

Thus (7.8) transforms as

$$T^{-1}\dot{\mathbf{u}}' + \frac{\mathrm{d}}{\mathrm{d}t}T^{-1}\cdot\mathbf{u}' = AT^{-1}\mathbf{u}' + B\mathbf{x}$$

respectively yielding

$$T^{-1}\dot{\mathbf{u}}' = \left(AT^{-1} - \frac{\mathrm{d}}{\mathrm{d}t}T^{-1}\right)\mathbf{u}' + B\mathbf{x}$$

and

$$\dot{\mathbf{u}}' = \left(TAT^{-1} - T\frac{\mathrm{d}}{\mathrm{d}t}T^{-1}\right)\mathbf{u}' + TB\mathbf{x}$$

Thus the first of the equations (7.9) is changed into

$$A' = TAT^{-1} - T\frac{\mathrm{d}}{\mathrm{d}t}T^{-1} \tag{7.30}$$

The extra term in (7.30) is the main reason why different topologies have different time-varying behavior. If two systems are to share the same transfer

function, they need to share the poles. In this case the matrices A and A' have the same diagonal or Jordan normal form (unless the system order is degenerate) and are therefore related by a similarity transformation. Given that B, C and B', C' are related via the same transformation matrix according to (7.9), the difference between the two systems will be purely the one of a different state-space basis, and we would expect a fully identical behavior of both. However, in order to have identical time-varying behavior, the matrices A and A' would need to be related via (7.30) rather than via a similarity transformation. In fact (7.30) cannot hold, unless at least one of the matrices A and A' depends not only on some externally controlled parameters (such as cutoff and resonance), but also on their derivatives, which is a highly untypical control scenario.

Transient response

In the derivation of the transient response in Section 7.7 we have been using the fact that

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(e^{-At}\mathbf{u}\right) = e^{-At}\dot{\mathbf{u}} - e^{-At}A\mathbf{u}$$

However if A is not constant then the above needs to be written as

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(e^{-At}\mathbf{u}\right) = e^{-At}\dot{\mathbf{u}} - \left(\frac{\mathrm{d}}{\mathrm{d}t}e^{-At}\right)\cdot\mathbf{u}$$

We might want to rewrite the derivative of e^{-At} as

$$\left(\frac{\mathrm{d}}{\mathrm{d}t}e^{-At}\right) = e^{-At}\frac{\mathrm{d}}{\mathrm{d}t}(-At) = e^{-At}\cdot\left(-A - t\frac{\mathrm{d}}{\mathrm{d}t}A\right)$$

but actually we cannot do that, since we don't know whether the derivative of $-At$ will commute with At. Thus, our derivation of the transient response stops right there.[7]

Diagonal form

Given that we are using a diagonal form as a replacement for another non-diagonal system, we already know that such replacement changes the time-varying behavior of the system due to the extra term in (7.30).

A more serious problem occurs in this situation if we want to go through parameter ranges where the system poles get close or equal to each other. Such situation is unavoidable if we want a pair of mutually conjugate complex poles of a real system to smoothly change into real poles, since such poles would need to become equal on the real axis before they can go further apart. As we have found out, the diagonal form doesn't support the case of coinciding poles in a continuous manner, since switching from poles to Jordan cells on the main diagonal is a non-continuous transformation of the state space.

[7] Notably, the same was the case for our transient response derivations for 1- and 2-pole cases, where we were assuming the fixed values of system parameters. Except for the 1-pole case, where the only available freedom degree in the 1×1 matrix A could be represented as the cutoff, leading to an equivalent representation of the modulation via time-warping.

Cutoff modulation

If all cutoff gains are identical and precede the integrators, it is convenient to factor them out of matrices A and B:

$$\dot{\mathbf{u}} = \omega_c \cdot (A\mathbf{u} + B\mathbf{x}) \tag{7.31a}$$
$$\mathbf{y} = C\mathbf{u} + D\mathbf{x} \tag{7.31b}$$

If the cutoff is varying with time, we could explicitly reflect this in the first equation, where we can also let B (but not A) vary with time:

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{u}(t) = \omega_c(t) \cdot (A\mathbf{u}(t) + B(t)\mathbf{x}(t))$$

Introducing $\mathrm{d}\tau = \omega_c(t)\mathrm{d}t$ we have

$$\frac{\mathrm{d}}{\mathrm{d}\tau}\mathbf{u}(t(\tau)) = A\mathbf{u}(t(\tau)) + B(t(\tau))\mathbf{x}(t(\tau))$$

or

$$\frac{\mathrm{d}}{\mathrm{d}\tau}\tilde{\mathbf{u}}(\tau) = A\tilde{\mathbf{u}}(\tau) + \tilde{\mathbf{x}}(\tau) \tag{7.32}$$

where

$$\tilde{\mathbf{u}}(\tau) = \mathbf{u}(t(\tau)) \qquad \tilde{\mathbf{x}}(\tau) = B(t(\tau))\mathbf{x}(t(\tau))$$

Thus, as we have already shown in Section 2.16, cutoff modulation is expressible as a warping of the time axis, provided cutoff is uniformly positive

$$\tau(t) = \int \omega_c(t)\mathrm{d}t \qquad \text{where } \omega_c(t) \geq \omega_0 > 0$$

where the time-warped system defined by (7.32) is time-invariant.

Note that cutoff modulation in (7.31) is a transformation of A which changes its eigenvalues but not its eigenvectors. Thus, if we diagonalize the system by a basis change, the new basis can stay unchanged, and there will not be the extra term in (7.30). Respectively, the diagonalized system will stay fully equivalent to the original one, even though the cutoff is being modulated. Apparently the diagonalized system also can be written in the factored-out-cutoff form (7.31).

Equivalence of systems under cutoff modulation

It's not difficult to realize that the equivalence under the condition of cutoff modulation in (7.31) holds not only between the original system and its diagonalized version, but between any two systems related by a basis change, since the cutoff modulation is not affecting the transformation between the two systems. Suppose we are having two systems sharing the same transfer function. In such case they have an equivalent behavior in the time-invariant case, but we wish to have it equivalent in the time-varying case too. More specifically, we would like make the second system have the time-varying behavior of the first one.

Since the transfer function is the same, both systems share the same diagonal form up to the ordering and the lengths of the basis vectors. The transformations between both systems and the shared diagonal form are cutoff-independent and therefore the systems are equivalent.

Equivalence under other modulations

We have already shown that two systems sharing the same transfer function are equivalent under the cutoff modulation (7.31). We often would wish to also analyse for the equivalence under modulation of other parameters. Generally this will not be the case, but the state-space form techniques may allow us to find out more details about the specific differences between the systems. In order to demonstrate some of the analysis possibilities, we are going to analyse the TSK allpass (Fig. 7.1), which we have been converting to the state-space form in Section 7.2.

Taking (7.4) let's replace the feedback amount k with damping R. From (5.17) we are having

$$\begin{aligned}\frac{2k}{k+1} &= 1 + \frac{k-1}{k+1} = 1 - R \\ \frac{1}{1+k} &= \frac{1}{1+\frac{1-R}{1+R}} = \frac{1+R}{1+R+1-R} = \frac{R+1}{2} \\ k - 1 &= \frac{1-R}{1+R} - 1 = \frac{1-R-1-R}{1+R} = -\frac{2R}{1+R}\end{aligned}$$

and thus (7.4) turns into

$$\dot{u}_1 = -Ru_1 + (R-1)u_2 + \frac{R+1}{2}x \tag{7.33a}$$

$$\dot{u}_2 = (R+1)u_1 - Ru_2 - \frac{R+1}{2}x \tag{7.33b}$$

$$y = -\frac{4R}{R+1}u_1 + \frac{4R}{R+1}u_2 + x \tag{7.33c}$$

Looking at the output mixing coefficients we notice a strong similarity to (4.23) where we subtract the bandpass signal (which, as we should remember, is obtained directly from one of the state variables of an SVF) from the input, the bandpass signal being multiplied by $4R$. On the other hand for the TSK allpass we have just obtained (7.33c):

$$y = -\frac{4R}{R+1}u_1 + \frac{4R}{R+1}u_2 + x = x - \frac{4R}{R+1}(u_1 - u_2)$$

This motivates to attempt an introduction of new state variables, where one of the variables will be a difference of u_1 and u_2. We expect this variable to behave somewhat like an SVF bandpass signal.

Attempting to turn $4R/(R+1)\cdot(u_1-u_2)$ into exactly $4Ru_1'$ (which is what we would have had for an SVF) might be not the best idea, since the transformation would be dependent on R, and it would be difficult to assess possible implications of such dependency. Instead we want something which is proportional to u_1-u_2, but the transformation should be independent of R. This is achieved by e.g.

$$\begin{aligned}u_1 &= u_2' + u_1' \\ u_2 &= u_2' - u_1'\end{aligned}$$

which implies $u_1' = (u_1 - u_2)/2$. Applying this transformation to (7.33) we have

$$\dot{u}_2' + \dot{u}_1' = -R(u_2' + u_1') + (R-1)(u_2' - u_1') + \frac{R+1}{2}x =$$

$$
\begin{aligned}
&= (1-2R)u_1' - u_2' + \frac{R+1}{x} \\
\dot{u}_2' - \dot{u}_1' &= (R+1)(u_2' + u_1') - R(u_2' - u_1') - \frac{R+1}{2}x = \\
&= (1+2R)u_1' + u_2' - \frac{R+1}{2}x \\
y &= -\frac{4R}{R+1}(u_2' + u_1') + \frac{4R}{R+1}(u_2' - u_1') + x = -\frac{8R}{R+1}u_1' + x
\end{aligned}
$$

from where

$$
\begin{aligned}
2\dot{u}_1' &= -4Ru_1' - 2u_2' + (R+1)x \\
2\dot{u}_2' &= 2u_1' \\
y &= -\frac{8R}{R+1}u_1' + x
\end{aligned}
$$

or

$$
\begin{aligned}
\dot{u}_1' &= -2Ru_1' - u_2' + \frac{R+1}{2}x \\
\dot{u}_2' &= u_1' \\
y &= -\frac{8R}{R+1}u_1' + x
\end{aligned}
$$

Now this looks very much like an SVF allpass, except that the input signal has been multiplied by $(R+1)/2$ and the bandpass signal is respectively multiplied by $8R/(R+1)$ instead of multiplying by $4R$ (Fig. 7.3). Note that the product of pre- and post-gains is still $4R$, exactly what we would normally use to build an SVF allpass. Thus, the only difference between the SVF allpass and the TSK allpass is the distribution of the pre- and post-bandpass gains.

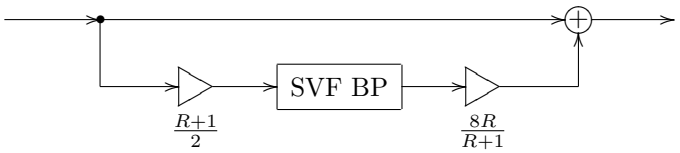


Figure 7.3: An equivalent representation of the allpass TSK filter from Fig. 5.35 using an SVF bandpass.

We could also cancel the denominator 2 of the pre-gain with the numerator of the post-gain (Fig. 7.4). Since 2 is a constant, "sliding" it through the SVF bandpass system effectively just rescales the internal state of the SVF by a factor of 2 (without introducing any new time-varying effects), but this rescaling is then compensated in the post-gain. Thus the system in Fig. 7.4 is fully equivalent to the one in Fig. 7.3.

7.13 Discrete-time case

Discrete-time block diagrams can be converted to the discrete-time version of the state-space form, which is also referred to as the *difference state-space form.*

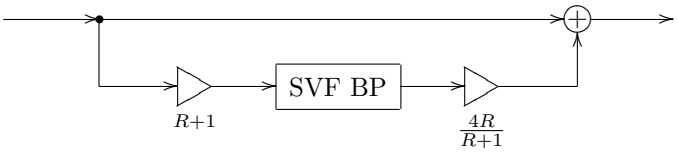


Figure 7.4: An equivalent modification of Fig. 7.3.

The main principles are the same, except that instead of $A\mathbf{u}+B\mathbf{x}$ delivering the input signals of the integrators, it delivers the input signals of the unit delays. The same values will occur at the outputs of the unit delays one sample later, thus the first state-space equation takes the form

$$\mathbf{u}[n+1] = A\mathbf{u}[n] + B\mathbf{x}[n]$$

The second equation is the same as in the continuous-time case:

$$\mathbf{y}[n] = C\mathbf{u}[n] + D\mathbf{x}[n]$$

Writing both equations together we obtain the discrete-time state-space form:

$$\mathbf{u}[n+1] = A\mathbf{u}[n] + B\mathbf{x}[n] \tag{7.34a}$$
$$\mathbf{y}[n] = C\mathbf{u}[n] + D\mathbf{x}[n] \tag{7.34b}$$

Transfer matrix

Substituting the complex exponential signal $\mathbf{x}[n] = \mathbf{X}(z)z^n$ into (7.34) we obtain

$$\mathbf{U}(z)z^{n+1} = A\mathbf{U}(z)z^n + B\mathbf{X}(z)z^n$$
$$\mathbf{Y}(z)z^n = C\mathbf{U}(z)z^n + D\mathbf{X}(z)z^n$$

from where

$$z\mathbf{U}(z) = A\mathbf{U}(z) + B\mathbf{X}(z)$$
$$\mathbf{Y}(z) = C\mathbf{U}(z) + D\mathbf{X}(z)$$

From the first of the equations we have

$$(z-A)\mathbf{U}(z) = B\mathbf{X}(z)$$

$$\mathbf{U}(z) = (z-A)^{-1}B\mathbf{X}(z) \tag{7.35}$$

Substituting this into the second equation we have

$$\mathbf{Y}(z) = C(z-A)^{-1}B\mathbf{X}(z) + D\mathbf{X}(z)$$

and thus

$$\mathbf{Y}(z) = H(z)\mathbf{X}(z)$$

where

$$H(z) = C(z-A)^{-1}B + D = \frac{C\,\mathrm{adj}(z-A)B}{\det(z-A)} + D$$

therefore the eigenvalues of A are the system poles.

Transient response

Substituting the complex exponential input $\mathbf{x}[n] = \mathbf{X}(z)z^n$ into (7.34a) we can rewrite (7.34a) as

$$\mathbf{u}[n+1] = A\mathbf{u}[n] + B\mathbf{X}(z)z^n$$

or as

$$\mathbf{u}[n] = A\mathbf{u}[n-1] + B\mathbf{X}(z)z^{n-1} = A\mathbf{u}[n-1] + \mathbf{q}z^n \tag{7.36}$$

where

$$\mathbf{q} = B\mathbf{X}(z)z^{-1}$$

Recursively substituting (7.36) into itself at progressively decreasing values of n we obtain

$$\begin{aligned}
\mathbf{u}[n] &= A\mathbf{u}[n-1] + \mathbf{q}z^n = \\
&= A\left(A\mathbf{u}[n-2] + \mathbf{q}z^{n-1}\right) + \mathbf{q}z^n = \\
&= A^2\mathbf{u}[n-2] + \left(Az^{-1} + 1\right)\mathbf{q}z^n = \\
&= A^2\left(A\mathbf{u}[n-3] + \mathbf{q}z^{n-2}\right) + \left(Az^{-1} + 1\right)\mathbf{q}z^n = \\
&= A^3\mathbf{u}[n-3] + \left(\left(Az^{-1}\right)^2 + Az^{-1} + 1\right)\mathbf{q}z^n = \\
&\dots \\
&= A^n\mathbf{u}[0] + \left(\left(Az^{-1}\right)^{n-1} + \left(Az^{-1}\right)^{n-2} + \dots + Az^{-1} + 1\right)\mathbf{q}z^n = \\
&= A^n\mathbf{u}[0] + \left(1 - \left(Az^{-1}\right)^n\right)\left(1 - Az^{-1}\right)^{-1}\mathbf{q}z^n = \\
&= A^n\mathbf{u}[0] + \left(z^n - A^n\right)(z - A)^{-1}\mathbf{q}z = \\
&= A^n\mathbf{u}[0] + \left(z^n - A^n\right)(z - A)^{-1}B\mathbf{X}(z) = \\
&= (z - A)^{-1}B\mathbf{X}(z)z^n + A^n\left(\mathbf{u}[0] - (z - A)^{-1}B\mathbf{X}(z)\right) = \\
&= \mathbf{u}_s[n] + A^n\left(\mathbf{u}[0] - \mathbf{u}_s[0]\right)
\end{aligned}$$

where

$$\mathbf{u}_s[n] = (z - A)^{-1}B\mathbf{X}(z)z^n = (z - A)^{-1}B\mathbf{x}[n]$$

is the steady-state response (compare to the transfer matrix for $\mathbf{u}$ in (7.35)), respectively

$$\mathbf{u}_t[n] = A^n\left(\mathbf{u}[0] - \mathbf{u}_s[0]\right) \tag{7.37}$$

The generalization to arbirary signals $\mathbf{x}[n]$ is done in the same way as in the continuous-time case. The steady-state and transient responses for $\mathbf{y}$ are trivially obtained from those for $\mathbf{u}$.

Stability

Considering the transient response in (7.37), we could diagonalize the system by a change of basis. If diagonalization is successful, then it's obvious than A^n decays to zero if and only if $|p_n| < 1\ \forall n$ and grows to infinity if $\exists p_n\colon |p_n| > 1$. Since neither the system poles nore the decaying of the transient response to zero depend on the basis choice, we have thereby established the criterion of stability of discrete time systems.

The non-diagonalizable case can be handled by using Jordan normal form, where the discrete-time Jordan 1-poles of the Jordan chains will be stable if and only if $|p_n| < 1\ \forall n$.

7.14 Trapezoidal integration

Writing (7.31) in an integral form we have

$$\mathbf{u} = \int \omega_c \left(A\mathbf{u} + B\mathbf{x}\right) \mathrm{d}t \tag{7.38a}$$

$$\mathbf{y} = C\mathbf{u} + D\mathbf{x} \tag{7.38b}$$

On the other hand, expressing direct form I trapezoidal integration Fig. 3.8 in equation form we have

$$y[n] = y[n-1] + \frac{x[n-1] + x[n]}{2} T \tag{7.39}$$

Applying (7.39) to the integral in (7.38a) we obtain

$$\mathbf{u}[n] = \mathbf{u}[n-1] + \omega_c \frac{A(\mathbf{u}[n] + \mathbf{u}[n-1]) + B(\mathbf{x}[n] + \mathbf{x}[n-1])}{2} T$$

from where

$$\left(1 - \frac{\omega_c T}{2} A\right) \mathbf{u}[n] = \left(1 + \frac{\omega_c T}{2} A\right) \mathbf{u}[n-1] + \frac{\omega_c T}{2} B\big(\mathbf{x}[n] + \mathbf{x}[n-1]\big)$$

and

$$\mathbf{u}[n] = \left(1 - \frac{\omega_c T}{2} A\right)^{-1} \left(\left(1 + \frac{\omega_c T}{2} A\right) \mathbf{u}[n-1] + \frac{\omega_c T}{2} B\big(\mathbf{x}[n] + \mathbf{x}[n-1]\big)\right) \tag{7.40}$$

Equation (7.40) is the resolved zero-delay feedback equation for the state-space form (7.31) (or, equivalently (7.38)). Since we have used direct form I intergators, it needs additional state variables for the storage of the previous input values, which we could have spared if direct form II or transposed direct form II integration was used.

Let's apply trasposed direct form II integration (3.3) to the integral in (7.38a). Apparently, we have a notation clash, since in (3.3) the variable u is an internal variable of the integrator. Notating this internal variable as $\mathbf{v}$ and notating the input signals of the integrators as $2\mathbf{w}$, and also not forgetting to introduce a non-unit sampling period T, we obtain from (3.3) a set of equations:

$$\begin{aligned}
\mathbf{u}[n] &= \mathbf{v}[n-1] + \mathbf{w}[n] && \text{obtained from (3.3a)} \\
\mathbf{v}[n] &= \mathbf{u}[n] + \mathbf{w}[n] && \text{obtained from (3.3b)} \\
\mathbf{w}[n] &= \frac{\omega_c T}{2} \left(A\mathbf{u}[n] + B\mathbf{x}[n]\right) && \text{obtained from (7.38a)}
\end{aligned}$$

Solving for $\mathbf{w}[n]$ we have

$$\mathbf{w}[n] = \frac{\omega_c T}{2} \big(A(\mathbf{w}[n] + \mathbf{v}[n-1]) + B\mathbf{x}[n]\big)$$

where $\mathbf{v}[n-1]$ are the previous states of the integrators, and respectively

$$\left(1 - \frac{\omega_c T}{2} A\right) \mathbf{w}[n] = \frac{\omega_c T}{2} \big(A\mathbf{v}[n-1] + B\mathbf{x}[n]\big)$$

and

$$\mathbf{w}[n] = \left(1 - \frac{\omega_c T}{2} A\right)^{-1} \frac{\omega_c T}{2} \big(A\mathbf{v}[n-1] + B\mathbf{x}[n]\big) \tag{7.41}$$

Equation (7.41) is another variant of the resolved zero-delay feedback equation (7.40), this time written for transposed direct form II form. The benefit, compared to (7.40), is that we only need to store the previous states of the integrators $\mathbf{v}[n-1]$.

Since $M^{-1} = \operatorname{adj} M / \det M$, the denominator of both equations (7.40) and (7.41) is $\det(1 - \omega_c T/2 \cdot A)$. Since $\det(\lambda - M) = 0$ is the eigenvalue equation, the denominator turns to zero when 1 becomes an eigenvalue of $\omega_c T/2 \cdot A$, or respectively when $2/T$ becomes an eigenvalue of $\omega_c A$. Thus, we have a limitation

$$\omega_c \cdot \max_{p_n \in \mathbb{R}} \{p_n\} < 2/T \tag{7.42}$$

under which the system doesn't get instantaneously unstable. Apparently $\omega_c p_n$ are simply the poles of the system, thus (7.42) simply states that the real poles of the system must be located to the left of $2/T$.[8]

SUMMARY

The state-space form essentially means writing the system as a differential (or difference, in the discrete-time case) equation system in a matrix form. Thereby we have a compact abstract representation of the system, which, differently from to the transfer function, doesn't lose essential information about the time-varying behavior. A particularly useful way to approach the state-space form analysis is by diagonalizing the matrix, which essentially separates the effects of different poles of the system from each other.

[8] Of course if there are complex poles sufficiently close to the real semiaxis $[2/T, +\infty)$, the performance of trapezoidal integration is also questionable.

Chapter 8

Raising the filter order

As the order of the filter grows, there are more and more different choices of the transfer function. Particularly, there is more than one way to introduce the resonance into a transfer function of order higher than 2. Some of the most interesting options were already discussed in the previous chapters.

We have also introduced the state-space form as a general representation for differential systems. However, being so general, the state-space form leaves lots of open questions in regards to the choice of topology and the user-facing parameters.

In this chapter we are going to discuss a number of standard topologies which can be used to construct a system of any given order and also a number of ways to map commonly used user-facing parameters, such as cutoff and resonsance, to the internal parameters of such systems. Note, however, that these structures and techniques are useful only occasionally, for rather specific purposes.

8.1 Generalized SVF

The idea of the ladder filter could have been generalized from a 4-pole to other numbers of poles, even though there are problems arising at pole counts other than 4. Could we somehow attempt to generalize the SVF?

The most natural way to generalize the SVF is probably to treat it as the so-called *controllable canonical form*, (Fig. 8.1) which is the analog counterpart of direct form II (Fig. 3.33). Apparently, the main difference between Fig. 3.33 and Fig. 8.1 is simply that all unit delays are replaced by integrators. The other differerence, namely the inverted feedback is merely a matter of convention, resulting in opposite signs of the coefficients a_n compared to what they would have been in the absence of the feedback inversion. We chose the convention with the inverted feedback mainly because it's more similar to the 2-pole SVF structure in Fig. 4.1.

The controllable canonical form allows to implement an arbitrary transfer function of N-th order (the requirement that the transfer function is a non-strictly proper rational function being implicitly understood). Indeed, it's not

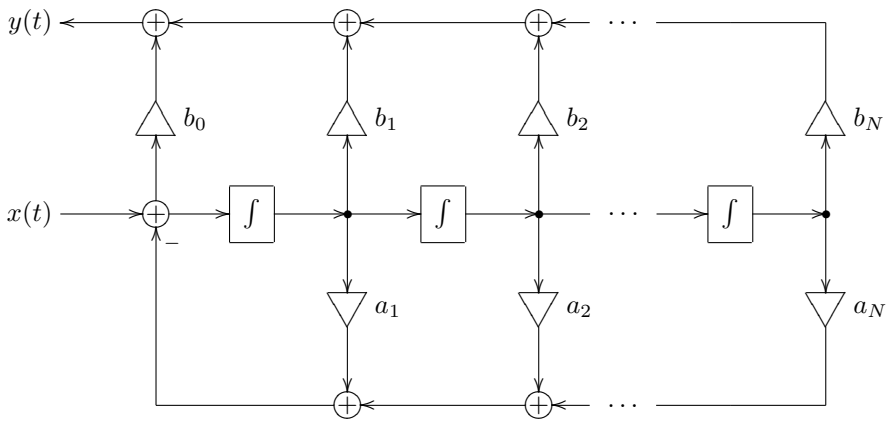


Figure 8.1: Generalized SVF (controllable canonical form).

difficult to figure out that the tranfer function of the system in Fig. 8.1 is

$$H(s) = \frac{\sum_{n=0}^{N} b_n s^{-n}}{1 + \sum_{n=1}^{N} a_n s^{-n}} = \frac{\sum_{n=0}^{N} b_n s^{N-n}}{1 + \sum_{n=1}^{N} a_n s^{N-n}} = \frac{\sum_{n=0}^{N} b_{N-n} s^{n}}{s^N + \sum_{n=0}^{N-1} a_{N-n} s^{n}}$$

Thus a_n and b_n are simply the denominator and numerator coefficients of the transfer function.

Normally Fig. 8.1 assumes unit-cutoff integrators, because the a_n and b_n coefficients provide enough freedom to implement any transfer function of the given order. However, in music DSP applications cutoff control is a common feature, therefore we could also allow the integrators to take identical non-unit cutoffs. Further letting $N = 2$, $a_2 = 1$ and $a_1 = 2R$ we obtain an SVF with b_n serving as modal mixing coefficients for HP, BP and LP outputs. On the other hand, at $N = 1$, $a_1 = 1$ we obtain the 1-pole filter we discussed in the beginning of this book.

Generally, letting a_N have a fixed value is a good way to remove the redundancy introduced into the system control by the embedded cutoffs of the integrators. It is not difficult to realize that

$$a_N = \prod(-p_n)$$

where p_n are the positions of the system poles when $\omega_c = 1$. Notably, this also can support the case of real poles of opposite signs, which cannot be implemented by a classical 2-pole SVF due to a_N being fixed to 1.

Unfortunately, there is no clear answer to what the coefficients a_n should be for $N > 2$. The simplicity of the 2-pole case was due to the fact that the denominator of a 2-pole transfer function essentially has only 2 degrees of freedom (corresponding to a_1 and a_2), one degree being taken by the cutoff, and we are being left with the remaining degree which just happens to correspond to

the resonance. With the 1-pole there was only one freedom degree, being taken by the cutoff. At $N > 2$ there are too many different options of how to map the freedom degrees to filter control parameters and there is no definite answer to that, although some of the options will be discussed later in this chapter.

Notably, with the numerator coefficients b_n there is a bit more clarity, as there are certain general considerations applying more or less for any choice of a_n. E.g. if the numerator is equal to a_N, we get some kind of an N-th order lowpass, since $H(0) = 1$ and $H(s) \sim a_N/s^N$ for $s \to \infty$. For the s^N numerator we have $H(\infty) = 1$ and $H(s) \sim s^N/a_N$ for $s \to 0$, corresponding to some kind of an N-th order highpass. For an even N and an $a_N^{1/2} s^{N/2}$ numerator we get $H(s) \sim s^{N/2}/a_N^{1/2}$ for $s \to 0$ and $H(s) \sim a_N^{1/2}/s^{N/2}$ for $s \to \infty$, corresponding to some kind of a bandpass. This however defines only the asymptotic behavior at 0 and ∞, the amplitude response shape in the middle can be pretty much arbitrary, being defined by the denominator.

By transposing the controllable canonical form one obtains the so-called *observable canonical form*. We are not going to address it in detail, as most of the discussion of the controllable canonical form above applies to the observable canonical form as well.

8.2 Serial cascade representation

Another structure which allows implementing arbitrary transfer functions is the serial cascade. It is probably the one most commonly used. The benefit compared to the generalized SVF is that in the serial cascade representation we are using only 1- and 2-pole filters and we can choose commonly known and well-studied structures to implement those. The benefit compared to the parallel implementation (discussed later in this chapter) is that the serial cascade form doesn't get ill-conditioned when system poles get close to each other.

Cascade decomposition

Given an arbitrary N-th order real transfer function, let's write it in the multiplicative form:

$$H(s) = g \cdot \frac{\prod_{n=1}^{N_z} (s - z_n)}{\prod_{n=1}^{N_p} (s - p_n)} \tag{8.1}$$

where $N_z \le N_p$, since $H(s)$ must be nonstrictly proper. Since $H(s)$ has real coefficients, all complex poles of $H(s)$ will come in conjugate pairs, and the same can be said about the zeros.

Now we are going to write each pair of conjugate poles as a purely real 2nd-order factor in the denominator:

$$(s - p)(s - p^*) = s^2 - s \cdot 2 \operatorname{Re} p + |p|^2$$

and we are going to write each pair of conjugate zeros as a purely real 2nd-order factor in the numerator:

$$(s - z)(s - z^*) = s^2 - s \cdot 2\,\mathrm{Re}\, z + |z|^2$$

Further, if necessary, we *can* combine any two real poles into a 2nd-order factor in the denominator:

$$(s - p_1)(s - p_2) = s^2 - (p_1 + p_2) \cdot s + p_1 p_2$$

and we can combine any two real zeros into a 2nd-order factor in the numerator:

$$(s - z_1)(s - z_2) = s^2 - (z_1 + z_2) \cdot s + z_1 z_2$$

Thus we can distribute all conjugate pair of poles and zeros into 2nd-order *real* rational factors of the form

$$\frac{s^2 + as + b}{s^2 + cs + d}$$

unless we do not have enough zeros, in which case there will be one or more 2nd-order real rational factors of the form

$$\frac{s + b}{s^2 + cs + d} \qquad \text{and/or} \qquad \frac{1}{s^2 + cs + d}$$

The remaining pairs of real poles and zeros can be combined into 1st-order real rational factors of the form

$$\frac{s + a}{s + b} \qquad \text{and/or} \qquad \frac{1}{s + b}$$

or they can be also combined into 2nd-order real rational factors, e.g.:

$$\frac{s + a_1}{s + b_1} \cdot \frac{s + a_2}{s + b_2} = \frac{s^2 + (a_1 + a_2)s + a_1 a_2}{s^2 + (b_1 + b_2)s + b_1 b_2}$$

Thus the entire transfer function is represented as a product of purely real 2nd- and 1st-order factors:

$$H(s) = g \cdot \prod_{n=1}^{N_2} H_{2n}(s) \cdot \prod_{n=1}^{N_1} H_{1n}(s) \tag{8.2}$$

where $H_{2n}(s)$ and $H_{1n}(s)$ are the 2nd- and 1st-order factors respectively. The gain coefficient g, if desired, can be factored into the numerator of one or several of the factors $H_{2n}(s)$ and $H_{1n}(s)$, so that the product expression gets a simpler form:

$$H(s) = \prod_{n=1}^{N_2} H_{2n}(s) \cdot \prod_{n=1}^{N_1} H_{1n}(s) \tag{8.3}$$

Now recall that 1-pole multimode can implement any stable real 1st-order transfer function and SVF can implement any stable real 2nd-order transfer function. This means that we can implement pretty much any $H(s)$ as a serial chain of SVFs[1] and 1st-order multimodes.[2] We will refer to the process of representing $H(s)$ is a cascade form as *cascade decomposition* of $H(s)$.

[1] Of course a multimode TSK, a multimode SKF, or any other 2nd-order filter with sufficient freedom in transfer function parameters would do instead of an SVF.

[2] Apparently $H(s)$ can be implemented by 1-poles and SVFs if its factors can be implemented by 1-poles and SVFs. Those which can not, can be implemented by generalized SVFs.

Cutoff control

The denominator $1 + s/\omega_c$ of a 1-pole filter is controlled by a single parameter, which is the filter cutoff. The denominator $1+2Rs/\omega_c+(s/\omega_c)^2$ of a 2-pole filter is controlled by cutoff and damping. Thus each of the 2- and 1-poles in (8.3) has a cutoff, defined by the positions of the respective poles. Writing explicitly these cutoff parameters in (8.3) we obtain

$$H(s) = \prod_{n=1}^{N_2} \bar{H}_{2n}(s/\omega_{2n}) \cdot \prod_{n=1}^{N_1} \bar{H}_{1n}(s/\omega_{1n})$$

where $\bar{H}_{2n}$ and $\bar{H}_{1n}$ are unit-cutoff versions of the same 2- and 1-poles and ω_{2n} and ω_{1n} are the respective cutoffs.

Suppose the above $H(s)$ defines a unit-cutoff filter. Then non-unit cutoff for $H(s)$ is achieved by

$$H(s/\omega_c) = \prod_{n=1}^{N_2} \bar{H}_{2n}(s/\omega_c\omega_{2n}) \cdot \prod_{n=1}^{N_1} \bar{H}_{1n}(s/\omega_c\omega_{1n}) \tag{8.4}$$

which means that the cutoffs of the underlying 2- and 1-poles are simply multiplied by ω_c and we have $\omega_c\omega_{2n}$ and $\omega_c\omega_{1n}$ as the 2- and 1-pole cutoffs.

One should remember, that it is important to apply one and the same prewarping for all filters in the cascade, as discussed in Section 3.8. E.g. we could choose to prewarp (8.4) at $\omega = \omega_c$, which means that we prewarp only ω_c (rather than individually prewarping the 2- and 1-pole cutoffs $\omega_c\omega_{2n}$ and $\omega_c\omega_{1n}$), thereby obtaining its prewarped version $\tilde{\omega}_c$, and then simply substitute $\tilde{\omega}_c$ for ω_c in (8.4):

$$H(s/\tilde{\omega}_c) = \prod_{n=1}^{N_2} \bar{H}_{2n}(s/\tilde{\omega}_c\omega_{2n}) \cdot \prod_{n=1}^{N_1} \bar{H}_{1n}(s/\tilde{\omega}_c\omega_{1n})$$

Thus, the 2- and 1-pole cutoffs become $\tilde{\omega}_c\omega_{2n}$ and $\tilde{\omega}_c\omega_{1n}$ respectively.

Cascaded model of a ladder filter

As an example of the just introduced technique we are going to implement the transfer function of a 4-pole lowpass ladder filter by a serial chain of two SVFs. A 4-pole lowpass ladder filter has no zeros and two conjugate pairs of poles for $k > 0$. By considering two coinciding poles on a real axis also as mutually conjugate, we can assume $k \geq 0$.

Since there are no zeros, we simply need a 2-pole lowpass SVF for each conjugate pair of poles. Let p_1, p_1^*, p_2, p_2^* be the poles of the ladder filter. According to (5.2)

$$p_{1,2} = -1 + \frac{\pm 1 + j}{\sqrt{2}} k^{1/4} \tag{8.5}$$

By (4.13), the cutoffs of the 2-pole lowpasses $\omega_{1,2} = |p_{1,2}|$ and $R = -\operatorname{Re} p_{1,2}/|p_{1,2}|$. Respectively the transfer function of the ladder filter can be represented as

$$H(s) = g \frac{1}{\left(\frac{s}{\omega_1}\right)^2 + 2R_1\frac{s}{\omega_1} + 1} \cdot \frac{1}{\left(\frac{s}{\omega_2}\right)^2 + 2R_2\frac{s}{\omega_2} + 1} \tag{8.6}$$

The unknown gain coefficient g can be found by evaluating (5.1) at $s = 0$, obtaining the condition $H(0) = 1/(1 + k)$. Evaluating (8.6) at $s = 0$ yields $H(0) = g$. Therefore

$$g = \frac{1}{1+k}$$

This gives us a cascade of 2-poles implementing a unit-cutoff ladder filter. Extending (8.6) to arbitrary cutoffs is respectively done by

$$H(s) = \frac{1}{1+k} \cdot \frac{1}{\left(\frac{s}{\omega_c\omega_1}\right)^2 + 2R_1\frac{s}{\omega_c\omega_1} + 1} \cdot \frac{1}{\left(\frac{s}{\omega_c\omega_2}\right)^2 + 2R_2\frac{s}{\omega_c\omega_2} + 1}$$

Cascaded multimode

The cascade decomposition can be also used to provide modal outputs, sharing the same transfer function denominator. In order to demonstrate this we will consider a serial connection of two SVFs.[3]

The transfer function of such structure can have almost any desired 4th order stable denominator.[4] We would like to construct modal outputs for such connection, so that by mixing those modal signals we should be able to obtain arbitrary numerators. This should allow us to share this chain of SVFs for generation of two or more signals which share the same transfer function's denominator.

We have several options of connecting two SVFs in series, depending on which of the modal outputs of the first SVF is connected to the second SVF's input. The most symmetric option seems to be picking up the bandpass output (Fig. 8.2).

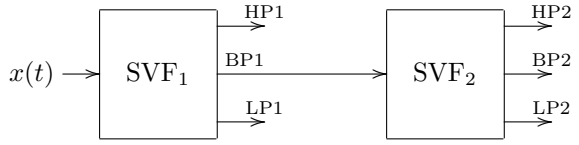


Figure 8.2: A multimode cascade of two SVFs.

Now let

$$D_1(s) = s^2 + 2R_1\omega_1 s + \omega_1^2$$
$$D_2(s) = s^2 + 2R_2\omega_2 s + \omega_2^2$$

be the denominators of the transfer functions of the two SVFs and let $D(s) = D_1(s)D_2(s)$ be their product. Writing out the transfer functions for the signals at the SVF outputs (in respect to the input signal $x(t)$ in Fig. 8.2) we obtain

$$H_{\mathrm{LP1}}(s) = \frac{\omega_1^2}{D_1(s)} = \frac{\omega_1^2 D_2(s)}{D(s)}$$

[3]The idea to specifically address this is the book arose from a discussion with Andrew Simper.

[4]Denominators not achievable by classical SVFs can be achieved by using generalized 2nd-order SVFs.

$$H_{\mathrm{BP1}}(s) = \frac{\omega_1 s}{D_1(s)} = \frac{\omega_1 s D_2(s)}{D(s)}$$
$$H_{\mathrm{HP1}}(s) = \frac{s^2}{D_1(s)} = \frac{s^2 D_2(s)}{D(s)}$$
$$H_{\mathrm{LP2}}(s) = \frac{\omega_2^2}{D_2(s)} \cdot H_{\mathrm{BP1}}(s) = \frac{\omega_2^2 \omega_1 s}{D(s)}$$
$$H_{\mathrm{BP2}}(s) = \frac{\omega_2 s}{D_2(s)} \cdot H_{\mathrm{BP1}}(s) = \frac{\omega_2 \omega_1 s^2}{D(s)}$$
$$H_{\mathrm{HP2}}(s) = \frac{s^2}{D_2(s)} \cdot H_{\mathrm{BP1}}(s) = \frac{\omega_1 s^3}{D(s)}$$

Or, since we have the common denominator $D(s)$ everywhere, we could concentrate just on the numerators:

$$N_{\mathrm{LP1}}(s) = \omega_1^2 D_2(s)$$
$$N_{\mathrm{BP1}}(s) = \omega_1 s D_2(s)$$
$$N_{\mathrm{HP1}}(s) = s^2 D_2(s)$$
$$N_{\mathrm{LP2}}(s) = \omega_2^2 \omega_1 s$$
$$N_{\mathrm{BP2}}(s) = \omega_2 \omega_1 s^2$$
$$N_{\mathrm{HP2}}(s) = \omega_1 s^3$$

Noticing from Fig. 8.2 that BP1 can be obtained as LP2 + $2R_2$BP2 + HP2 anyway, we can drop the respective numerator from the list and try to arrange the remaining ones in the order of the descending polynomial order:

$$N_{\mathrm{HP1}}(s) = s^2 D_2(s) = s^4 + 2R_2\omega_2 s^3 + \omega_2^2 s^2$$
$$N_{\mathrm{HP2}}(s) = \omega_1 s^3$$
$$N_{\mathrm{BP2}}(s) = \omega_2 \omega_1 s^2$$
$$N_{\mathrm{LP2}}(s) = \omega_2^2 \omega_1 s$$
$$N_{\mathrm{LP1}}(s) = \omega_1^2 D_2(s) = \omega_1^2 s^2 + 2R_2\omega_1^2\omega_2 s + \omega_1^2\omega_2^2$$

The last line doesn't really fit, and the first one looks more complicated than the next three, but we can fix that by replacing the first and the last lines by linear combinations:

$$N_{\mathrm{HP1}}(s) - 2R_2\frac{\omega_2}{\omega_1}N_{\mathrm{HP2}}(s) - \frac{\omega_2}{\omega_1}N_{\mathrm{BP2}}(s) = s^4$$
$$N_{\mathrm{HP2}}(s) = \omega_1 s^3$$
$$N_{\mathrm{BP2}}(s) = \omega_2 \omega_1 s^2$$
$$N_{\mathrm{LP2}}(s) = \omega_2^2 \omega_1 s$$
$$N_{\mathrm{LP1}}(s) - \frac{\omega_1}{\omega_2}N_{\mathrm{BP2}}(s) - 2R_2\frac{\omega_1}{\omega_2}N_{\mathrm{LP2}}(s) = \omega_1^2\omega_2^2$$

Thus we can obtain all powers of s from linear combinations of LP1, HP1, LP2, BP2 and HP2, thereby being able to construct arbitrary polynomials of orders up to 4 for the numerator.

Notably, instead of connecting the bandpass output of the first SVF to the input of the second SVF, as it has been shown in Fig. 8.2, we could have connected the lowpass or the highpass output. This would have resulted in somewhat different math, but essentially gives the same modal mixture options.

8.3 Parallel representation

Real poles

Given a transfer function which has only real poles which are all distinct, we could expand it into a sum of 1st-order partial fractions. Each such 1st-order fraction corresponds to a 1-pole and we could implement the transfer function as a sum of 1-poles. Essentially this is identical to the diagonal state-space form, which, provided all system poles are real and sufficiently distinct (so that no ill-conditioning occurs), is just a set of parallel Jordan 1-poles.

In the case of a single-input single-output system, which we are currently considering, the transfer function of such diagonal system, given by (7.18), has the form

$$H(s) = \sum_{n=1}^{N} \frac{c_n b_n}{s - p_n} + d \tag{8.7}$$

where b_n and c_n are the input and output gains respectively. Given a particular nonstrictly rational $H(s)$, the partial fraction expansion (8.7) uniquely defines d and the products $c_n b_n$. The respective freedom of choice of c_n and b_n can be resolved by letting $b_n = 1\ \forall n$ and thus we control the numerator of the transfer function by the output mixing coefficients c_n (Fig. 8.3).[5]

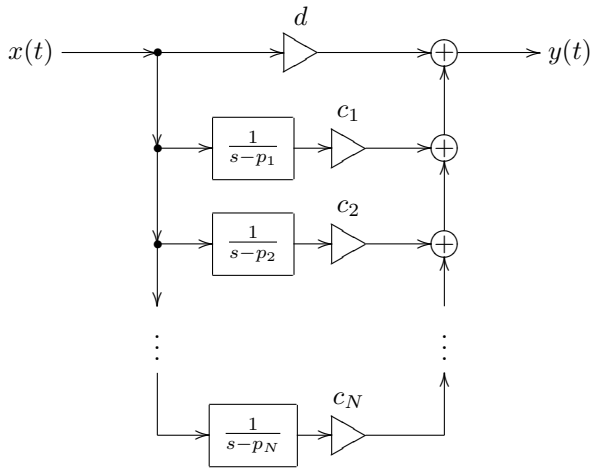


Figure 8.3: Implementation by parallel Jordan 1-poles.

We could also replace Jordan 1-poles by ordinary 1-pole lowpasses, where we need to divide the mixing coefficients by the respective cutoffs ω_{cn} (Fig. 8.4).

[5]Of course, we could instead let $c_n = 1$ and control the transfer function numerator by the input gains b_n, or distribute the control between b_n and c_n.

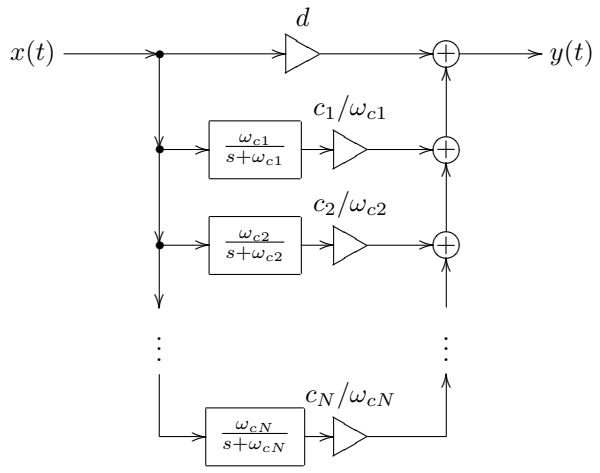


Figure 8.4: Implementation by parallel 1-pole lowpasses.

The global cutoff control of the entire filter in Fig. 8.3 or Fig. 8.4 is achieved in the same way as with serial cascades. Obviously, the usual consideration of common prewarping of the 1-pole components applies here as well.

Complex poles

If system poles are complex we need to use the real diagonal form, which replaces the complex Jordan 1-poles with Jordan 2-poles. For a single-input single-output system, equation (7.28) takes the form

$$H(s) = \sum_{\operatorname{Im} p_n > 0} \frac{\alpha_n s + \beta_n}{s^2 - 2\operatorname{Re} p_n \cdot s + |p_n|^2} + \sum_{\operatorname{Im} p_n = 0} \frac{c_n b_n}{s - p_n} + d \qquad (8.8)$$

We could obtain the explicit expressions for α_n and β_n from the derivation of (7.28), but it would be more practical to simply obtain their values from the partial fraction expansion of $H(s)$. That is, given $H(s)$, we find α_n and β_n (as well as, of course, $c_n b_n$ and d) from (8.8). We also should remember that, according to the freedom of choice of the state space basis vectors lengths, we could choose any non-zero input gains vector, e.g. $\begin{pmatrix}1 & 0\end{pmatrix}^{\mathsf{T}}$ which means that we are using only the "real part" input of the Jordan 2-pole.[6] According to (7.26), the contribution of such Jordan 2-pole to $H(s)$ will be

$$\frac{1}{s^2 - 2\operatorname{Re} p_n \cdot s + |p_n|^2} \begin{pmatrix} c_n & c_{n+1} \end{pmatrix} \begin{pmatrix} s - \operatorname{Re} p_n & -\operatorname{Im} p_n \\ \operatorname{Im} p_n & s - \operatorname{Re} p_n \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} =$$
$$= \frac{c_n(s - \operatorname{Re} p_n) + c_{n+1} \operatorname{Im} p_n}{s^2 - 2\operatorname{Re} p_n \cdot s + |p_n|^2} = \frac{c_n s + (c_{n+1} \operatorname{Im} p_n - c_n \operatorname{Re} p_n)}{s^2 - 2\operatorname{Re} p_n \cdot s + |p_n|^2}$$

Thus

$$\alpha_n = c_n$$

[6] The dual approach would be to let the output mixing vector $\begin{pmatrix}1 & 0\end{pmatrix}$, in which case we control the transfer function's numerator by the input gains.

$$\beta_n = c_{n+1} \operatorname{Im} p_n - c_n \operatorname{Re} p_n$$

from where

$$c_n = \alpha_n$$
$$c_{n+1} = \frac{\beta_n + \alpha_n \operatorname{Re} p_n}{\operatorname{Im} p_n}$$

Thus, having found α_n and β_n, we can find c_n and c_{n+1}. The respective structure is shown in Fig. 8.5. Notice that as $\operatorname{Im} p_n$ becomes smaller, c_{n+1} becomes larger. This is the ill-conditioning effect of the diagonal form discussed in Section 7.11.

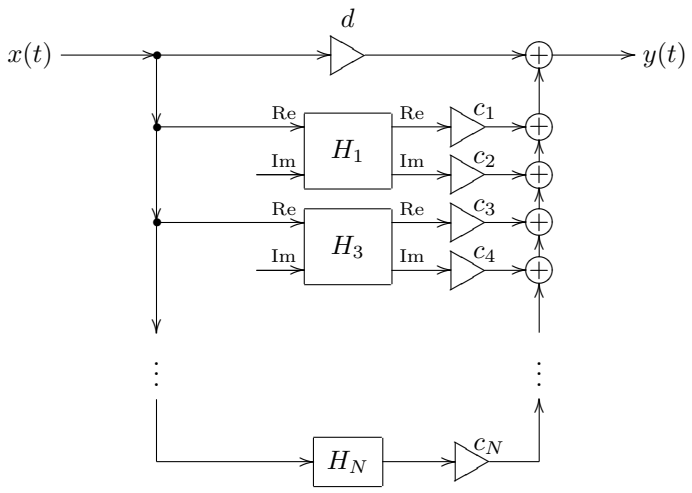


Figure 8.5: Implementation by parallel Jordan 2- and 1-poles. Disconnected imaginary part inputs are receiving zero signals.

Similarly to how we could replace Jordan 1-poles with ordinary 1-pole lowpasses, we could replace Jordan 2-poles by some other 2-poles, e.g. by SVFs. Finding the output mixing coefficients becomes simpler, since, apparently, the coefficients α_n and β_n in (8.8) now simply correspond to SVF bandpass and lowpass output gains (properly scaled by the cutoff). Fig. 8.6 illustrates.

Another benefit of an SVF is that it doesn't have a problem at the point where its poles coincide and also can support the case of real poles, meaning that we could convert arbitrary pairs of parallel 1-poles into an SVF. The same apparently could be done by an SKF/TSK. There would still be a problem though, if poles of different parallel 2-poles coincide, resulting in the already known ill-conditioning effect.

Regarding the cutoff control of the entire system, there is no difference from the parallel 1-poles case.

Coinciding poles

Generally, anything with repeated or close to each other poles cannot be implemented in a parallel form and needs some non-parallel implementation (SVF,

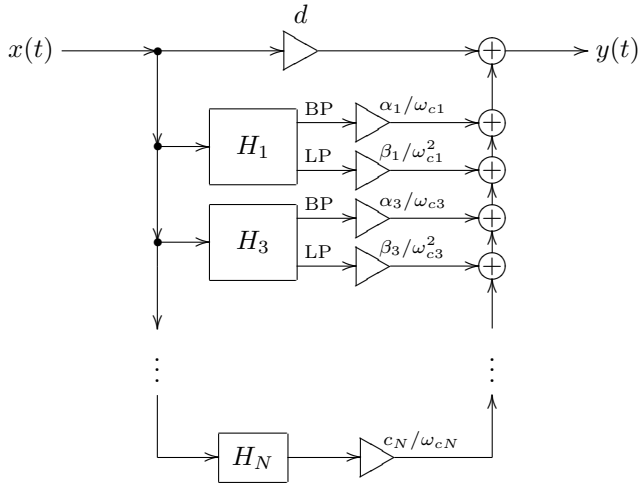


Figure 8.6: Implementation by parallel SVFs and 1-pole lowpasses.

a chain of SVFs, Jordan chain, etc.) However the implementation could still be partially parallel, where the poles may be repeated within each block, but different parallel blocks shouldn't have poles at the same locations.

8.4 Cascading of identical filters

So we have learned a number of different ways to implement higher-order transfer functions, of which cascaded form is said to be usually the best option, however, how do we construct these transfer functions in the first place? E.g. how do we generalize a resonating 2-pole transfer function to a 4-th or 8-th order? Or how do we generalize a 1-st order lowpass to a 5-th or 8-th order?

One possible way which could immediately occur to us is to stack several identical filters together. Note that, given a filter with the transfer function $G(s)$ and another one with the transfer function $H(s) = G^N(s)$ and looking at their decibel-scale amplitude responses, we notice that the latter is simply the former multiplied by N, that is the amplitude response becomes scaled N times vertically (obviously, the same scaling is happening to the phase response). Particularly this means that the rolloff slope of the filter becomes N times steeper.

Therefore in order to generalize a 1-st order lowpass $1/(1+s)$ to the N-th order we could simply connect N such lowpasses in series:

$$H(s) = \left(\frac{1}{1+s}\right)^N$$

resulting in the amplitude response curve in Fig. 8.7. It looks as if the cutoff of $H(s) = G^N(s)$ is is too low. In principle we could address this by shifting the filter cutoff, so that $|G^N(j)| = 1/\sqrt{2}$. In order to do so we solve the equation

$$\frac{1}{|1+j\omega|^N} = \frac{1}{\sqrt{2}}$$

obtaining the frequency which should be treated as the cutoff point of each of the chain's elements:

$$\omega = \sqrt{2^{1/N} - 1}$$

so the transfer function becomes

$$H(s) = \left(\frac{1}{1 + s\sqrt{2^{1/N} - 1}} \right)^N \tag{8.9}$$

This looks a bit better (Fig. 8.8) and can be taken as a possible option.

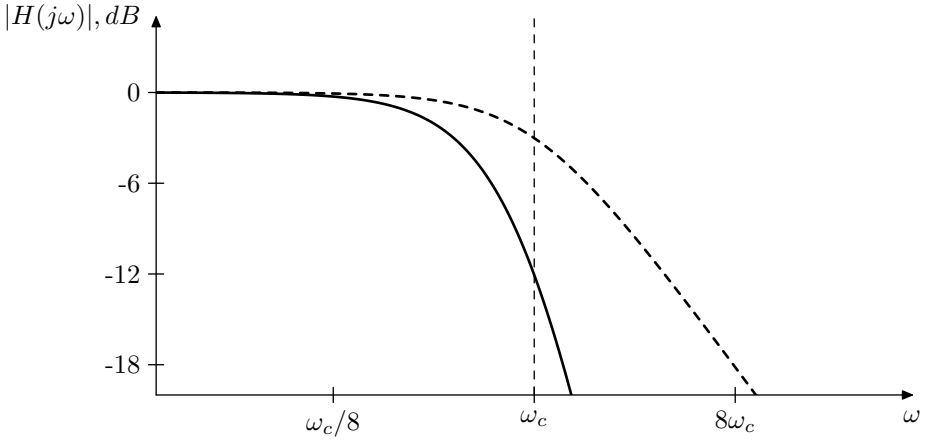


Figure 8.7: Amplitude response of a 1-pole lowpass filter (dashed) vs. amplitude response of a serial chain of 4 identical 1-pole lowpass filters (solid).

.

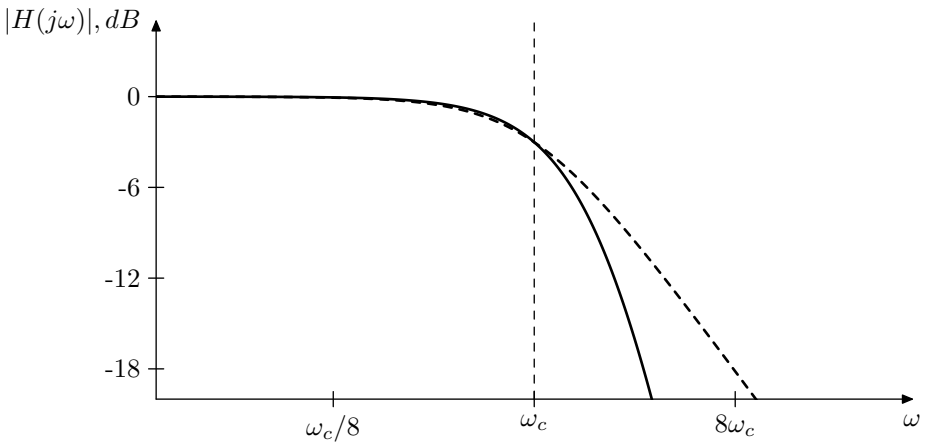


Figure 8.8: Amplitude response of a 1-pole lowpass filter (dashed) vs. amplitude response of a serial chain of 4 identical 1-pole lowpass filters with adjusted cutoff (solid).

.

In the same way we could generalize a resonating 2-nd order lowpass $1/(1+2Rs+s^2)$ to the $2N$-th order by connecting N of such lowpasses together

$$H(s) = \left(\frac{1}{1+2Rs+s^2}\right)^N$$

However in this case the situation is somewhat worse than with 1-poles. First we notice that the resonance peak becomes much higher at the same damping (Fig. 8.9). At first sight it doesn't look like a big problem, we could simply use smaller values of the damping. However if we compare the ampiltude response curves of a 2-pole vs. N stacked 2-poles with the damping adjusted to produce the same peak height,[7] we notice that due to the now smaller damping value the resonance peak of the 2-pole chain is much wider than the peak of a single 2-pole (Fig. 8.10), all in all not a very desirable scenario.

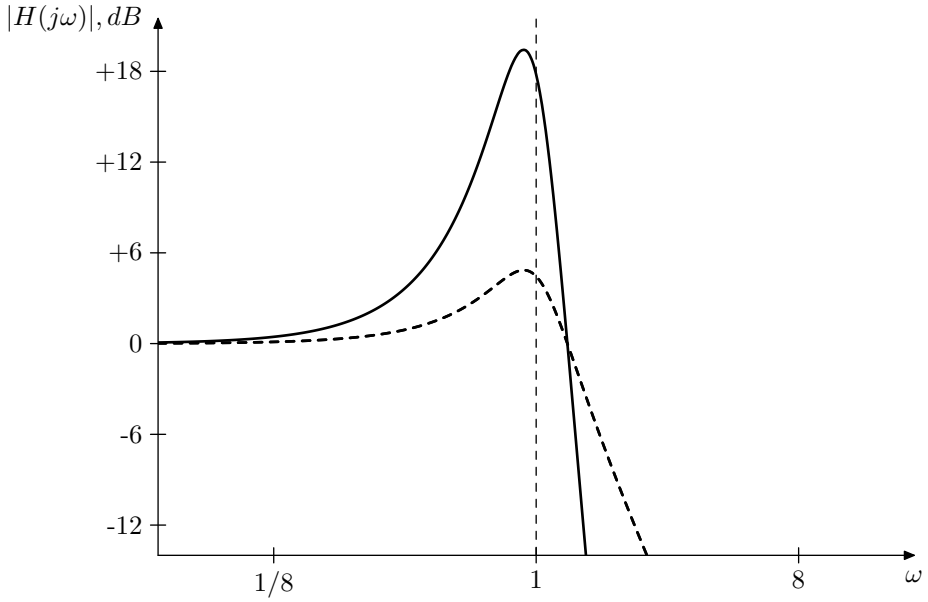


Figure 8.9: Amplitude response of a 2-pole filter (dashed) vs. amplitude response of a serial chain of 4 identical 2-pole filters (solid).

.

8.5 Butterworth transformation

We have seen that cascading N identical filters is one possible way to obtain higher-order filters, which effectively scales the decibel-scale amplitude response and the phase response of the filter N times vertically, respectively making the filter rolloff N times steeper.

Another way to make the rolloff N times steeper would be finding a transformation which shrinks the amplitude response in the logarithmic frequency

[7] We can do this using formulas (4.7) and (4.8).

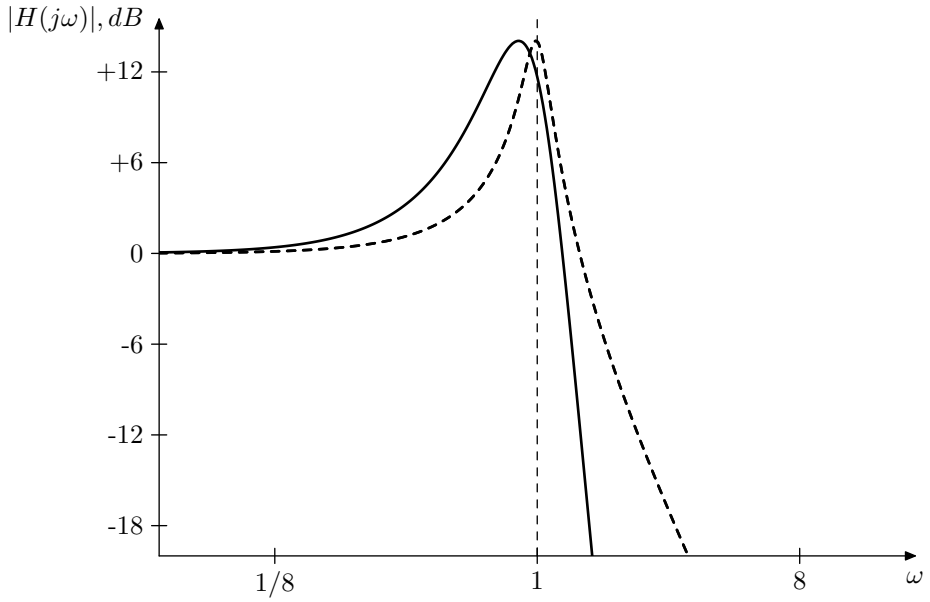


Figure 8.10: Amplitude response of a 2-pole filter (dashed) vs. amplitude response of a serial chain of 4 identical 2-pole filters with adjusted damping (solid).

.

scale N times:

$$\log\omega \leftarrow N\log\omega \qquad \omega \geq 0 \qquad N = 2,3,4,\ldots$$

(where we don't care about $\omega < 0$ because for real filters $|H(j\omega)| = |H(-j\omega)|$, and where $\log 0 = -\infty$). Or equivalently

$$\omega \leftarrow \omega^N \qquad \omega \geq 0 \tag{8.10}$$

The readers may recall the LP to HP transformation $s \leftarrow 1/s$ which flips the responses in the logarithmic frequency axis. One could try to draw an analogy and attempt substitutions of the form $s \leftarrow s^N$ or $s \leftarrow as^N$ ($a \in \mathbb{C}$, $|a| = 1$), however it's not difficult to convince oneself that such substitutions do not work. Nevertheless, the basic direction is mostly right. Just instead of of performing an argument substitution on the transfer function, we will directly apply (8.10) to the amplitude response $|H(j\omega)|$. That is we will be looking for such $H'(s)$ that

$$|H'(j\omega)| = |H(j\omega^N)| \qquad \omega \geq 0 \tag{8.11}$$

We will refer to the transformation of $H(s)$ into $H'(s)$ defined by (8.11) as *Butterworth transformation*.[8] The integer N will be respectively referred to

[8]The term *Butterworth transformation* has been coined by the author and is originating from the fact that this transformation, when applied to 1-pole filters, generates Butterworth filters. At the time of the writing the author is not aware of this concept being described elsewhere in the literature and would be thankful for any pointers to the commonly used terminology, if any exists.

as the *order* of the Butterworth transformation. We will denote Butterworth transformation as

$$H'(s) = \mathcal{B}\left[H(s)\right]$$

or, if we want to explicitly specify the order

$$H'(s) = \mathcal{B}_N\left[H(s)\right]$$

where $H'(s)$ denotes the new transfer function obtained as the result of the transformation.[9]

Without having developed the transformation details yet, we can already establish several properties of this transformation, which follow from (8.11):

- the transformation doesn't change a constant function:

$$\mathcal{B}\left[a\right] = a \tag{8.12a}$$

- a constant gain can be simply factored out of the transformation:

$$\mathcal{B}\left[g \cdot H(s)\right] = g \cdot \mathcal{B}\left[H(s)\right] \tag{8.12b}$$

- a change of the cutoff is shrunk N times in the logarithmic scale after the transformation:

$$\mathcal{B}_N\left[H(s/a)\right] = \mathcal{B}_N\left[H(s)\right]\Big|_{s \leftarrow s/a^{1/N}} \tag{8.12c}$$

- the transformation commutes with LP to HP substitution

$$\mathcal{B}_N\left[H(1/s)\right] = \mathcal{B}_N\left[H(s)\right]\Big|_{s \leftarrow 1/s} \tag{8.12d}$$

- the transformation distributes over multiplication:

$$\mathcal{B}\left[H_1(s)H_2(s)\right] = \mathcal{B}\left[H_1(s)\right] \cdot \mathcal{B}\left[H_2(s)\right] \tag{8.12e}$$

- the transformation distributes over division:

$$\mathcal{B}\left[H_1(s)/H_2(s)\right] = \mathcal{B}\left[H_1(s)\right] / \mathcal{B}\left[H_2(s)\right] \tag{8.12f}$$

- Butterworth transformations can be chained:

$$\mathcal{B}_N\left[\mathcal{B}_M\left[H(s)\right]\right] = \mathcal{B}_{N \cdot M}\left[H(s)\right] \tag{8.12g}$$

[9] Of course, (8.11) doesn't uniquely define $H'(s)$. E.g. if $H'(s)$ satisfies (8.11), then so does $-H'(s)$. In that sense Butterworth transformation is not uniquely defined. However during the development of the Butterworth transformation we will suggest some default choices which will work most of the time. Assuming these default choices, the Butterworth transformation becomes uniquely defined.

Since (8.11) doesn't uniquely define the transformation result, the above properties have to be understood in the sense that the right-hand side can be taken as one possible result of the transformation in the left-hand side. However the amplitude responses of the transformation results are uniquely defined and in those terms the above properties can be understood as usual equalities. E.g. the property (8.12g) can be understood as

$$|\mathcal{B}_N\left[\mathcal{B}_M\left[H(s)\right]\right]| = |\mathcal{B}_{N\cdot M}\left[H(s)\right]| \qquad \forall s = j\omega,\ \omega \in \mathbb{R}$$

Instead of developing Butterworth transformation immediately for arbitrary order filters we are going to first find a way to apply it to 1-pole filters and then to 2-pole filters. At that point we will be able to simply use the property (8.12e) to apply Butterworth transformation to arbitrary-order filters by representing these arbitrary order filters as cascades of 1-st and 2-nd order filters.

8.6 Butterworth filters of the 1st kind

As we just mentioned, first we will develop a way to apply Butterworth transformation to 1-pole filters, in which case we will more specifically refer to this transformation as *Butterworth transformation of the 1st kind*. The results of Butterworth transformation of the 1st kind coincide with filters commonly known as *Butterworth filters*. However in this book later we will generalize the idea of Butterworth filters to include the results of Butterworth transformation of filters of orders higher than 1. In order to be able to tell between different kinds of Butterworth filters, we are going to more specifically refer to the filters obtained by Butterworth transformation of 1-pole filters as *Butterworth filters of the 1st kind*.

Considering that a 1-pole transfer function is essentially a ratio of two 1st-order polynomials

$$H(s) = \frac{P_1(s)}{P_2(s)}$$

and that the amplitude response of $H(s)$ can be written as a ratio of formal amplitude responses of these polynomials:

$$|H(j\omega)| = \frac{|P_1(j\omega)|}{|P_2(j\omega)|}$$

it is sufficient to develop the transformation for 1st-order polynomials. The transformation of $H(s)$ can be then trivially obtained as:

$$H'(s) = \mathcal{B}\left[H(s)\right] = \frac{\mathcal{B}\left[P_1(s)\right]}{\mathcal{B}\left[P_2(s)\right]} = \frac{P_1'(s)}{P_2'(s)}$$

where $P_1'(s)$ and $P_2'(s)$ are transformed polynomials $P_1(s)$ and $P_2(s)$.

Transformation of polynomial $P(s) = s + 1$

We begin by obtaining the Butterworth transformation of the polynomial $P(s) = s + 1$. Its formal amplitude response is

$$|P(j\omega)| = \sqrt{1 + \omega^2}$$

and we wish to find $P'(s) = \mathcal{B}[P(s)]$ such that

$$|P'(j\omega)| = |P(j\omega^N)| = \sqrt{1+\omega^{2N}}$$

In order to get rid of the square root we can deal with squared amplitude response instead

$$|P(j\omega)|^2 = 1+\omega^2$$
$$|P'(j\omega)|^2 = 1+\omega^{2N}$$

Now we would like to somehow obtain $P'(s)$ from the latter equation.

In order to do so, let's notice that

$$|P(j\omega)|^2 = 1+\omega^2 = 1-(j\omega)^2 = (1+j\omega)(1-j\omega) = P(j\omega)P(-j\omega) = Q(j\omega)$$

where $Q(s) = P(s)P(-s)$, so the roots of $Q(s)$ consist of the root of $P(s)$ at $s=-1$ and of its origin-symmetric image at $s=1$, the latter being the root of $P(-s)$. This motivates to introduce $Q'(s)$ such that

$$Q'(j\omega) = 1+\omega^{2N}$$

and then try to factor it into $P'(s)P'(-s)$ in such a way that

$$|P'(j\omega)|^2 = P'(j\omega)P'(-j\omega) = Q'(j\omega)$$

In order to find the possible ways to factor $Q(s)$ into $P'(s)P'(-s)$ let us find the roots of $Q(s)$. Instead of solving $Q'(s)=0$ for s let's solve $Q'(j\omega)=0$ for ω, where we formally let ω take complex values. The solutions in terms of s are related to the solutions in terms of ω through $s=j\omega$.

Solving $Q'(j\omega) = 1+\omega^{2N} = 0$ for ω we obtain

$$\omega = (-1)^{1/2N} = e^{j\alpha} \qquad \alpha = \pi\frac{2n+1}{2N} = \pi\frac{\frac{1}{2}+n}{N} \qquad n = 0,\dots,2N-1 \quad (8.13)$$

The solutions are illustrated in Figs. 8.11 and 8.12 where the complex plane can be alternatively interpreted in terms of s or in terms of ω (note the labelling of the axes), thus these figures simultaneously illustrate the solutions in terms of ω or in terms of s. Thus the $2N$ roots of $Q'(s)$ are equally spaced on a unit circle with an angular step of π/N. If N is odd there will be roots at $s=\pm 1$ otherwise there are no real roots.

Another possible way to look at the solutions of $Q'(j\omega)=0$ is to rewrite the equation $1+\omega^{2N}=0$ as

$$1+\omega^{2N} = \omega^{2N} - j^2 = (\omega^N + j)(\omega^N - j) = 0$$

In this case the roots obtained from the equation $\omega^N - j = 0$ will be interleaved with the roots obtained from the equation $\omega^N + j = 0$ (Figs. 8.11 and 8.12 illustrate). Sometimes therefore such roots are referred to as even and odd roots respectively, since they occur respectively at even and odd n in (8.13). This distinction usually can be ignored, but occasionally becomes important.

Having found the roots of $Q'(s)$ how do we split them into the roots of $P'(s)$ and the roots of $P'(-s)$? Obviously we cannot do this splitting in an arbitrary way, since there are several special properties which need to be satisfied.

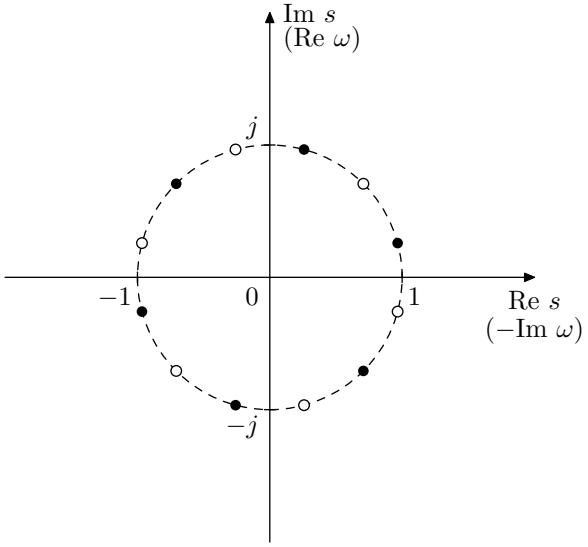


Figure 8.11: Roots of $Q'(s)$ for the Butterworth transformation of the 1st kind of an even order ($N = 6$). White and black dots correspond to even and odd roots.

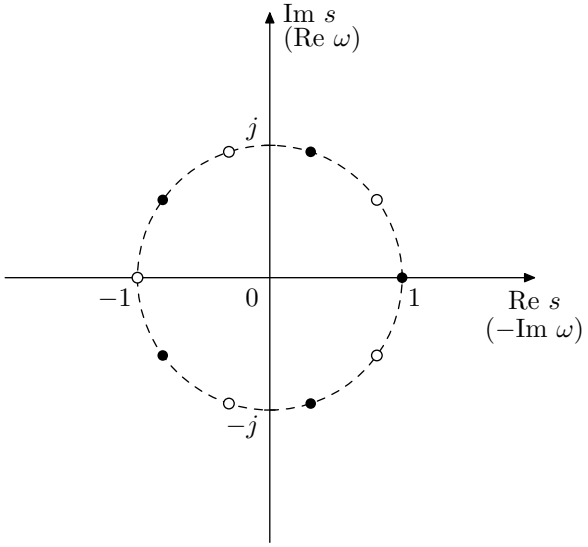


Figure 8.12: Roots of $Q'(s)$ for the Butterworth transformation of the 1st kind of an odd order ($N = 5$). White and black dots correspond to even and odd roots.

- For any possible polynomial $P'(s)$ its roots are origin-symmetric to the roots of $P'(-s)$, so our root splitting must respect this property.

- $P'(s)$ must be a real polynomial. This requires its roots to be either real or coming in complex conjugate pairs.

- If $P'(s)$ is the denominator of the filter's transfer function, then its roots must be located in the left complex semiplane (in order for the filter to be stable).

- The requrement $|P'(j\omega)|^2 = P'(j\omega)P'(-j\omega)$ implies $|P'(j\omega)| = |P'(-j\omega)|$. In order to satisfy the latter, the roots of $P'(s)$ must be symmetric to the roots of $P'(-s)$ with respect to the imaginary axis (essentially it is the same reasoning which we had in the discussion of minimum phase and maximum phase zero positioning).

Looking at Figs. 8.11 and 8.12 it's not difficult to notice that all of the above requirements will be satisfied if we choose the roots in the left complex semiplane to be the roots of $P'(s)$ and the roots in the right complex semiplane as the roots of $P'(-s)$ respectively.[10]

Having found the roots p'_n of $P'(s)$ we still need to find the leading coefficient g' of $P'(s)$:

$$P'(s) = g' \cdot \prod_n (s - p'_n)$$

In order to do so, notice that (8.11) implies $|P'(0)| = |P(0)|$. Since $P(0) = 1$ and $|P(0)| = 1$ we should have $|P'(0)| = 1$. Actually, if we let $g' = 1$ we will obtain $P'(0) = 1$. Indeed,

$$P'(0) = \prod_{n=1}^{N} (0 - p'_n) = \prod_{n=1}^{N} (-p'_n)$$

That is $P'(0)$ is equal to the product of all roots of $P'(-s)$. Looking at Figs. 8.11 and 8.12 we notice that the product of all roots of $P'(-s)$ is equal to 1 and thus $P'(0) = 1$.[11]

Thus, by finding the roots and the leading coefficient of $P'(s)$ we have obtained a real polynomial $P'(s) = \mathcal{B}[P(s)]$ in the multiplicative form. In practical filter implementations the complex conjugate pairs of factors of $P'(s)$ will be represented by 2nd-order filter sections, the purely real factor of $P'(s)$ appearing for odd N will be represented by a 1st-order filter section:

$$P'(s) = (s+1)^{N \wedge 1} \cdot \prod_n (s^2 + 2R_n s + 1)$$

where

$$N \wedge 1 = \begin{cases} 1 & \text{if } N \text{ is odd} \\ 0 & \text{if } N \text{ is even} \end{cases}$$

stands for bitwise conjunction.

[10] If $P(s)$ is the numerator of a transfer function, then the roots all being in the left semiplane imply the minimum phase implementation. However in this case we could instead pick up the right semiplane roots as the roots of $P(s)$, thereby obtaining a maximum phase transfer function. Or one could take the minimum phase implementation and exchange one conjugate pair of roots of $P(s)$ against the matching conjugate pair of roots of $P(-s)$. Or one could exchange several of such pairs. Or one could exhange the real roots of $P(s)$ and $P(-s)$ if the transformation order is odd. Still, the default choice will be to take the roots from the left semiplane.

[11] Obviously $g' = -1$ would also ensure $|P'(0)| = 1$. However, the default choice will be $g' = 1$.

Arbitrary 1st-order polynomials

Considering $P(s)$ of a more generic form $P(s) = s + a$ $(a > 0)$ we notice that essentially the procedure is the same as for $P(s) = s + a$ except that instead of the equation $\omega^{2N} + 1 = 0$ we obtain the equation

$$\omega^{2N} + a^2 = 1$$

This means that the roots of $Q'(s)$ are no longer located on the unit circle but on a circle of radius $a^{1/N}$. It is not difficult to see that the leading coefficient of $P'(s)$ is still equal to 1.

The above result also could have been obtained by rewriting $P(s)$ as $P(s) = a \cdot (s/a + 1)$ and applying properties (8.12b) and (8.12c), which on one hand gives a more intuitive understanding of why the circle of roots is scaled by $a^{1/N}$, on the other hand can serve as an explicit proof of (8.12c) for the case of Butterworth transformation of the 1st kind.

The case of $a = 0$ $(P(s) = s)$ can be obtained as a limiting case[12] $a \to +0$ resulting in $P'(s) = s^N$.

If $a < 0$ then, noticing that the amplitude responses of $P(s) = s + a$ and $P(s) = s - a$ are identical (for $a \in \mathbb{R}$), we could obtain $P'(s)$ as Butterworth transformation of $P(s) = s - a$. However, since the root of $P(s)$ is in the right semiplane, it would be logical to also pick the right semiplane roots of $Q'(s)$ as the roots of $P'(s)$. Particularly, if $P(s)$ is the numerator of a maximum phase filter, the transformation result will retain the maximum phase property.

The 1st-order polynomials of the most general form $P(s) = a_1 s + a_0$ can be treated by rewriting them as $P(s) = a_1 \cdot (s + a_0/a_1)$, if $a_1 \neq 0$. The case of $a_1 = 0$ can be simply treated as a limiting case $a_1 \to 0$, where we drop the vanishing higher-order terms of $P'(s)$, resulting in $P'(s) = a_0$.

Lowpass Butterworth filter of the 1st kind

Given

$$H(s) = \frac{1}{s+1} \tag{8.14}$$

we transform the denominator $P(s) = s+1$ according to the previous discussion of the Butterworth transformation of a 1st order polynomial. The roots of the transformed polynomial (located on the unit circle) become the poles of $H'(s)$. The numerator of $H'(s)$ is obviously unchanged by the transformation. Thus we obtain

$$H'(s) = \left(\frac{1}{s+1}\right)^{N \wedge 1} \cdot \prod_n \frac{1}{s^2 + 2R_n s + 1}$$

where the $1/(s + 1)$ term occurs in case of an odd N ("$N \wedge 1$" standing for bitwise conjunction). Therefore $H'(s)$ can be implemented as a series of 1-pole and 2-pole lowpass filters, where the 1-pole appears in case of an odd N.

Fig. 8.13 compares the amplitude response of a Butterworth lowpass filter of the 1st kind $(N = 2)$ against the prototype 1-pole lowpass filter. One can observe the increased steepness of the cutoff slope resulting from the shrinking along the logarithmic frequency axis. Fig. 8.14 compares the same Butterworth

[12] Treating as a limiting case (here and later in the text) is important because it ensures the continuity of the result at the limiting point.

lowpass filter against cascading of identical 1st order lowpasses, that is comparing the shrinking along the logarithmic frequency axis vs. stretching along the logarithmic amplitude axis. One can see that the Butterworth lowpass filter has the sharpest cutoff corner among different filters in Figs. 8.13 and 8.14.

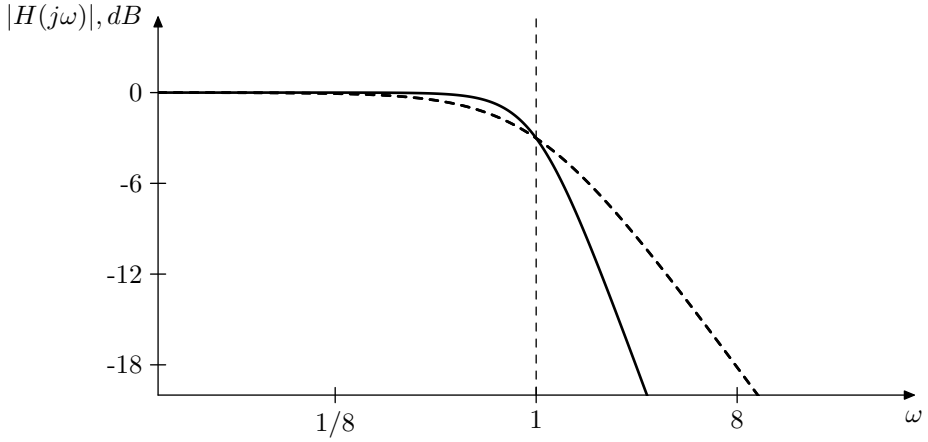


Figure 8.13: 2nd order lowpass Butterworth filter of the 1st kind (solid line) vs. 1st order lowpass filter (dashed line).

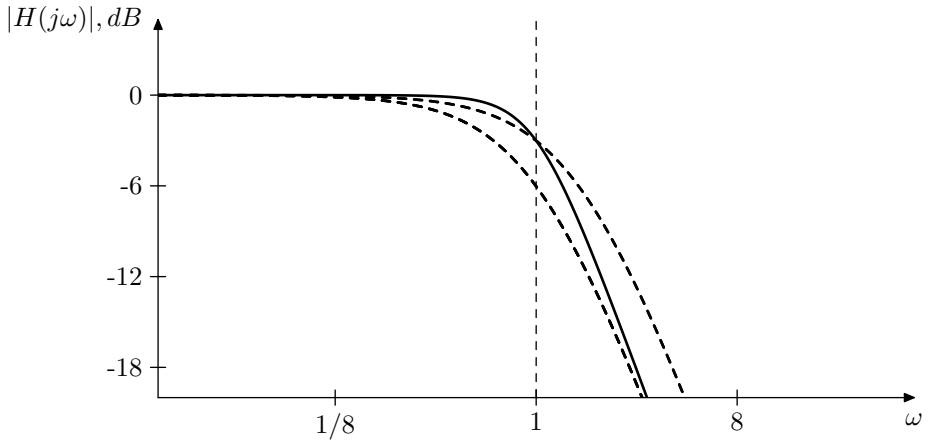


Figure 8.14: 2nd order lowpass Butterworth filter of the 1st kind (solid line) vs. duplicated 1st order lowpass filter without and with cutoff adjustment (dashed lines).

It is useful to know and recognize the expression for the squared amplitude response of a Butterworth lowpass filter of the 1st kind. Since the squared amplitude response of (8.14) is

$$|H(j\omega)|^2 = \frac{1}{1+\omega^2}$$

after the substitution $\omega \leftarrow \omega^N$ we obtain

$$|H'(j\omega)|^2 = \frac{1}{1+\omega^{2N}} \tag{8.15}$$

This expression is used in traditional derivation of Butterworth filters. Essentially the N-th order lowpass Butterworth filter is traditionally defined as a filter whose the amplitude response satisfies (8.15). Note that by (8.15) the 1-pole lowpass is the Butterworth filter of order 1. We can formally treat it as a 1st-order Butterworth transformation of itself

$$\frac{1}{1+s} = \mathcal{B}_1\left[\frac{1}{1+s}\right]$$

It is also useful to explicitly know the transfer function of the Butterworth lowpass filter of the 1st kind of order $N = 2$. It's not difficult to realize that for $P(s) = s + 1$ the roots of $P'(s)$ are located 45° away from the negative real semiaxis. Thus the respective damping is $R = \arccos 45° = 1/\sqrt{2}$ and

$$H'(s) = \frac{1}{s^2 + \sqrt{2}s + 1}$$

This damping value and the 2nd-order term $s^2 + \sqrt{2}s + 1$ appears in all Butterworth filters of the 1st kind of order $N = 2$ (highpass, bandpass, etc.) The readers may also recall the appearance of the damping value $R = 1/\sqrt{2}$ in the discussion of 2-pole filters, where it was mentioned that at $R = 1/\sqrt{2}$ the 2-pole filter turns into a Butterworth filter. This also corresponds to the fact that among all non-resonating (in the sense of the missing resonance peak) 2nd-order filters the Butterworth filter is the one with the sharpest possible cutoff corner in the amplitude response.

Highpass Butterworth filter of the 1st kind

For

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

we have the same denominator as for the respective lowpass. Thus the result of the denominator transformation is the same as for the lowpass. The result of the numerator transformation is s^N and thus

$$H'(s) = \left(\frac{s}{s+1}\right)^{N\wedge 1} \cdot \prod_n \frac{s^2}{s^2 + 2R_n s + 1}$$

That is we obtain the same result as for the 1-pole lowpass, except that instead of a series of lowpasses we should take a series of highpasses. Fig. 8.15 illustrates the respective amplitude response.

It is not difficult to verify that the highpass Butterworth filter obtained in the described above way is identical to the result of LP to HP substitution applied to the lowpass Butterworth filter of the same order, which is in agreement with (8.12d).

Bandpass Butterworth filter of the 1st kind

For an even N, by formally putting a numerator $s^{N/2}$ over the Butterworth transformation of a polynomial $P(s) = 1 + s$ we obtain a kind of a bandpass filter:

$$H'(s) = \prod_n \frac{s^{N/2}}{s^2 + 2R_n s + 1}$$

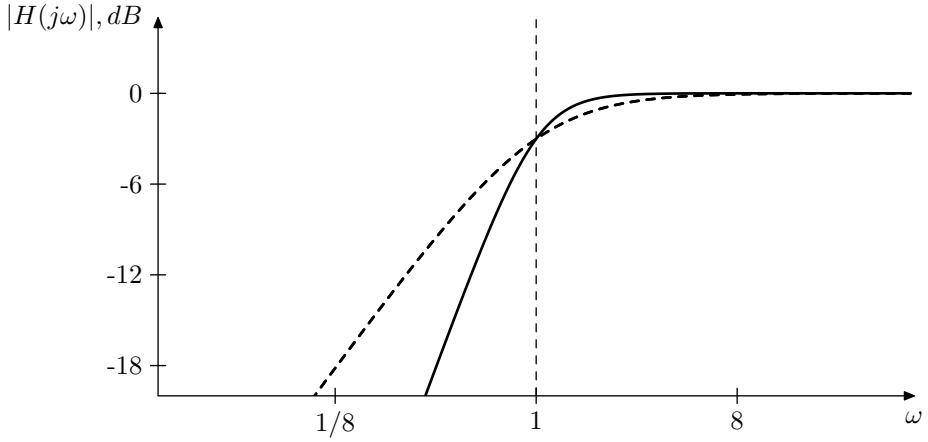


Figure 8.15: 2nd order highpass Butterworth filter of the 1st kind vs. 1st order highpass filter (dashed line).

(Fig. 8.16), which can be also formally seen as a Butterworth transformation of $H(s) = s^{1/2}/(s+1)$.

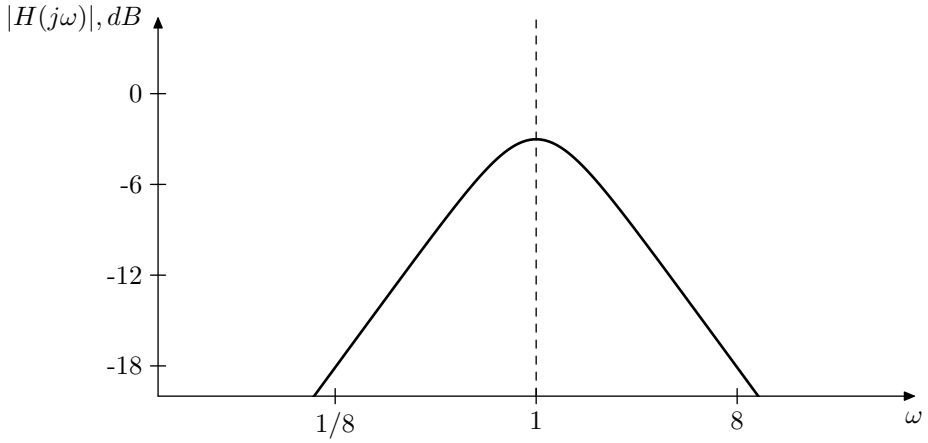


Figure 8.16: 2nd order bandpass Butterworth filter of the 1st kind.

Note that thereby this bandpass filter doesn't have any parameters to control, except the cutoff. As we will see a bit later in the discussion of Butterworth filters of the 2nd kind, this filter also can be obtained by an order $N/2$ Butterworth transformation of the 2-pole bandpass $H(s) = s/(s^2 + \sqrt{2}s + 1)$. Therefore there is not much point in specifically using Butterworth bandpass filters of the 1st kind, one can simply use Butterworth bandpass filters of the 2nd kind instead, achieving exactly the same response at a particular resonance setting.

A bandpass filter which has controllable bandwidth can be obtained by applying the LP to BP substitution to a Butterworth lowpass filter of the 1st kind. Apparently this produces a normalized bandpass (Fig. 8.17). This filter does not coincide with the result of the Butterworth transformation of the normal-

ized 2-pole bandpass $H(s) = \sqrt{2}s/(s^2 + \sqrt{2}s + 1)$. The reason is that in the first case we have a Butterworth transformation of a 1-pole lowpass $1/(1 + s)$ followed by the LP to BP substitution, while in the second case we first have the LP to BP substitution (with an appropriately chosen bandwidth) applied to $1/(1 + s)$ yielding $H(s) = \sqrt{2}s/(s^2 + \sqrt{2}s + 1)$, which is then followed by the Butterworth transformation. So it's the opposite order of the application of LP to BP substitution and the Butterworth transformation.

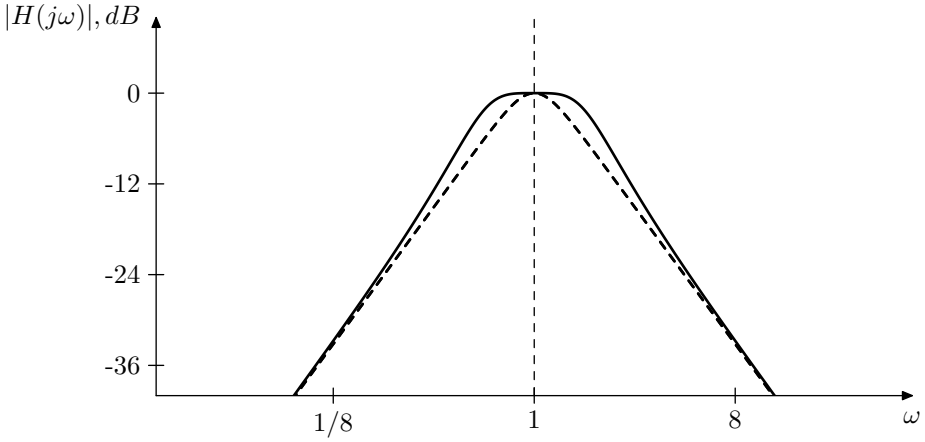


Figure 8.17: A bandwidth-tuned LP to BP substitution of a lowpass Butterworth filter of the 1st kind vs. Butterworth transformation of $H(s) = \sqrt{2}s/(s^2 + \sqrt{2}s + 1)$ (dashed line).

The LP to BP substitution can be performed algebraically on the transfer function of the Butterworth lowpass. In order to simplify things, the substitution can be applied in turn to the poles of each of the underlying 1- and 2-pole filters of the cascaded implementation of the Butterworth lowpass. After organizing the transformed poles into mutually conjugate pairs, we can simply construct the result as a series of normalized 2nd order bandpasses, defined by those pole pairs. Alternatively the LP to BP substitution can be implemented using the integrator substitution technique (Fig. 4.19).

8.7 Butterworth filters of the 2nd kind

Now we are going to apply the Butterworth transformation to 2nd order polynomials and respectively 2nd order filters. Such transformation will be referred to as *Butterworth transformation of the 2nd kind* and the filters obtained as the results of the tranformation will be referred to *Butterworth filters of the 2nd kind*.

Transformation of polynomial $P(s) = s^2 + 2Rs + 1$

We will first consider the following 2nd order polynomial

$$P(s) = s^2 + 2Rs + 1$$

corresponding to the denominator of a unit-cutoff 2-pole filter.

It will be most illustrative to obtain the Butterworth transformation of the 2nd kind as a combination of two opposite perturbations of two Butterworth transformations of the 1st kind. Factoring $P(s)$ we obtain

$$P(s) = (s + a_1)(s + a_2) = P_1(s)P_2(s)$$

At $R = 1$ we have $a_1 = a_2 = 1$ and $P(s)$ is a product of two 1st-order polynomials $P_1(s) = P_2(s) = s + 1$. Applying the Butterworth transformation of the 1st kind to each of the polynomials $P'_1(s)$ and $P'_2(s)$ we obtain two identical sets of the roots of $P'_1(s)$ and $P'_2(s)$ respectively. We can also consider the respective (also identical) extended polynomials

$$\begin{aligned} Q_1(s) &= P_1(s)P_1(-s) \\ Q_2(s) &= P_2(s)P_2(-s) \\ Q(s) &= P(s)P(-s) = Q_1(s)Q_2(s) \\ Q'_1(s) &= P'_1(s)P'_1(-s) \\ Q'_2(s) &= P'_2(s)P'_2(-s) \\ Q'(s) &= P'(s)P(-s) = Q'_1(s)Q'_2(s) \end{aligned}$$

which additionally contain the right-semiplane roots. As we should remember from the discussion of the Butterworth transformation of the 1st kind, the roots in each of the two sets corresponding to $Q'_1(s)$ and $Q'_2(s)$ are equally spaced on the unit circle.[13]

Now suppose we initially have $R = 1$ and then increase R to a value $R > 1$, resulting in a_1 growing and a_2 decreasing, staying reciprocal to each other:

$$a_1 = R + \sqrt{R^2 - 1} \qquad a_2 = R - \sqrt{R^2 - 1} \qquad (a_1 a_2 = 1)$$

(Fig. 8.18). Since a_1 and a_2 are the "cutoffs" of the 1st-order polynomials $s + a_1$ and $s + a_2$, from the properties of the Butterworth transformation of the 1st kind we obtain that the radii of the circles, on which the roots of $Q'_1(s)$ and $Q'_2(s)$ are located, become equal to

$$r'_1 = (R + \sqrt{R^2 - 1})^{1/N} \qquad r'_2 = (R - \sqrt{R^2 - 1})^{1/N} \qquad (r'_1 r'_2 = 1)$$

Thus, one circle grows and the other circle shrinks, while their radii are staying reciprocal to each other (Fig. 8.19).

Now let's decrease R from 1 to a value $0 < R < 1$. This makes a_1 and a_2 complex:

$$a_1 = e^{j\alpha} \qquad a_2 = e^{-j\alpha} \qquad (\cos\alpha = R, \quad a_1 a_2 = 1)$$

Writing out the "amplitude response" we have

$$\begin{aligned} |P(j\omega)|^2 &= P(j\omega)P(-j\omega) = P_1(j\omega)P_2(j\omega) \cdot P_1(-j\omega)P_2(-j\omega) = \\ &= P_1(j\omega)P_1(-j\omega) \cdot P_2(j\omega)P_2(-j\omega) = \\ &= Q_1(j\omega)Q_2(j\omega) = (\omega^2 + a_1^2) \cdot (\omega^2 + a_2^2) \end{aligned}$$

[13] With Butterworth transformation of the 2nd kind we won't be making a distinction between even and odd roots. Instead we will be paying attention to which roots originate from $Q_1(s)$ and which from $Q_2(s)$.

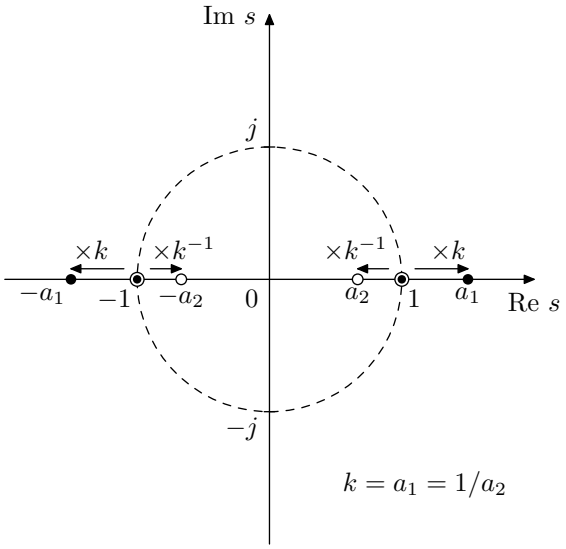


Figure 8.18: Roots of $Q(s)$ for $R > 1$ (black dots are roots of $Q_1(s)$, white dots are roots of $Q_2(s)$) and their positions at $R = 1$ (indicated by circled dots, where each such dot denotes a root of $Q_1(s)$ coinciding with a root of $Q_2(s)$).

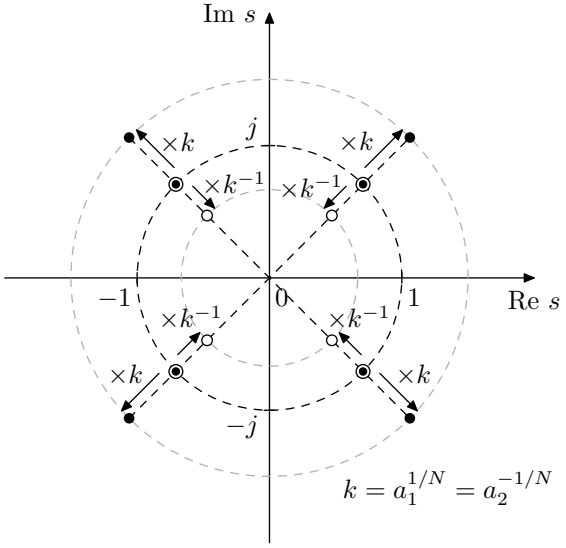


Figure 8.19: Roots of $Q'(s)$ for $R > 1$ (black dots are roots of $Q'_1(s)$, white dots are roots of $Q'_2(s)$) and their positions at $R = 1$ (indicated by circled dots, where each such dot denotes a root of $Q'_1(s)$ coinciding with a root of $Q'_2(s)$). Butterworth transformation order $N = 2$.

Respectively, our goal is to have

$$|P'(j\omega)|^2 = Q'_1(j\omega)Q'_2(j\omega) = Q_1(j\omega^N)Q_2(j\omega^N) = (\omega^{2N} + a_1^2) \cdot (\omega^{2N} + a_2^2)$$

So how do we find the roots of $Q'_1(s)$ and $Q'_2(s)$? If $a_1 = 1$ ($\alpha = 0$, $R = 1$) then, as we just discussed, $Q'_1(s)$ simply generates a set of the Butterworth roots of the 1st kind on the unit circle. Now if we replace $\alpha = 0$ with $\alpha > 0$ (corresponding to replacing $R = 1$ with $R < 1$) this means a rotation of a_1 by the angle α (Fig. 8.20). This rotates all roots of $Q'_1(j\omega) = \omega^{2N} + a_1^2$ by α/N (Fig. 8.21). At the same time a_2 will be rotated by $-\alpha$ and respectively all roots of $Q'_2(j\omega) = \omega^{2N} + a_2^2$ by $-\alpha/N$.

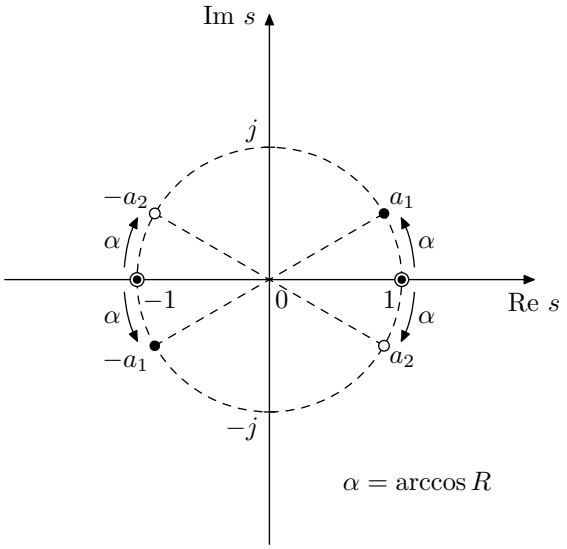


Figure 8.20: Roots of $Q(s)$ for $0 < R < 1$ (black dots are roots of $Q_1(s)$, white dots are roots of $Q_2(s)$) and their positions at $R = 1$ (indicated by circled dots, where each such dot denotes a root of $Q_1(s)$ coinciding with a root of $Q_2(s)$).

Even though generally for $\alpha > 0$ the set of roots of $\omega^{2N} + a_1^2$ is not symmetric relatively to the imaginary axis and neither is the set of roots of $\omega^{2N} + a_2^2$, the combination of the two sets is symmetric (as one can observe from Fig. 8.21). Thus we can simply drop the roots in the right semiplane, the same way as we did for $R \geq 1$. Note that this also means that we do not need to rotate the full set of roots of $Q'_1(s)$ and $Q'_2(s)$. Since at the end we are interested just in the left-semiplane roots, it suffices to rotate only the left-semiplane halves of the roots of $Q'_1(s)$ and $Q'_2(s)$ (that is, the roots of $P'_1(s)$ and $P'_2(s)$), as long as the roots do not cross the imaginary axis. It is not difficult to realize that the said crossing of the imaginary axis happens at $\alpha = \pi/2$ corresponding to $R = 0$, where one of the roots on the imaginary axis will be from $P'_1(s)$ and the other from $P'_2(s)$.

So, let's reiterate. At $R = 1$ ($\alpha = 0$) the roots of $P'(s)$ consist of two identical sets, each set being just the (left-semiplane) roots of a Butterworth transformation of a 1st-order polynomial $s+1$, all roots in such set being located on the unit cicle. For $R > 1$ we need to change the radii of both sets in a reciprocal manner:

$$r' = (R + \sqrt{R^2 - 1})^{\pm 1/N}$$

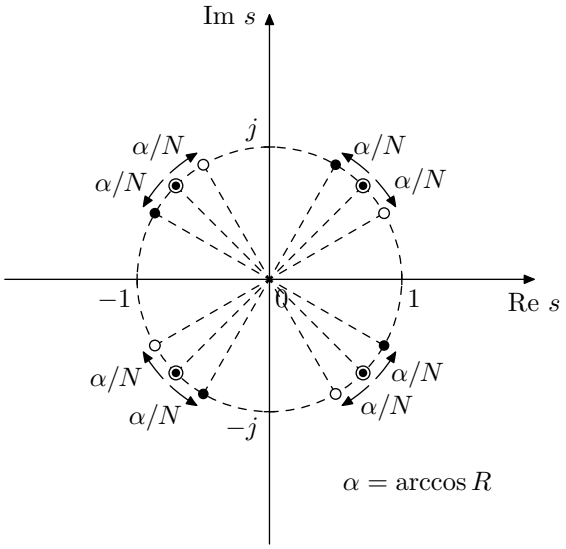


Figure 8.21: Roots of $Q'(s)$ for $0 < R < 1$ (black dots are roots of $Q'_1(s)$, white dots are roots of $Q'_2(s)$) and their positions at $R = 1$ (indicated by circled dots, where each such dot denotes a root of $Q'_1(s)$ coinciding with a root of $Q'_2(s)$). Butterworth transformation order $N = 2$.

(Fig. 8.19). For $R < 1$ we need to rotate both sets by opposite angles

$$\Delta\alpha' = \pm\alpha/N \qquad \alpha = \arccos R$$

(Fig. 8.21).

We have mentioned that at $R = 0$ ($\alpha = \pi/2$) two of the rotated roots of $P'(s)$ reach the imaginary axis. Another special case occurs when the roots of $P(s)$ are halfway from the "neutral position" ($\alpha = 0$) to selfoscillation ($\alpha = \pi/2$), that is when $\alpha = \pi/4$ ($R = 1/\sqrt{2}$). In this case the four roots of $Q(s)$ are equally spaced on the unit circle with the angular step $\pi/2$. In the process of the Butterworth transformation we rotate the roots of $Q'_1(s)$ and $Q'_2(s)$ by $\pm\alpha/N = \pm\pi/4N$, resulting in the set of roots of $Q'(s)$ being equally spaced on the unit circle by the angular step $\pi/2N$. But this is the set of roots of the Butterworth transformation of the 1st kind of order $2N$ (which produces the same polynomial order $2N$ as the order N Butterworth transformation of the 2nd kind). This result becomes obvious if we notice that at $R = 1/\sqrt{2}$ and $\alpha = \pi/4$ the polynomial $s^2 + 2Rs + 1$ is the result of the Butterworth transformation of the 1st kind of order 2 of the polynomial $s+1$. It is therefore no wonder that a Butterworth transformation of order 2 followed by a Butterworth transformation of order N is equivalent to the Butterworth transformation of order $2N$ (in other words, shrinking along the frequency axis by the factor $2N$ is equivalent to shrinking first by the factor of 2 and then by the factor of N).

Seamless transition at $R = 1$

In the derivation of the Butterworth transformation of the 2nd kind we have been treating the cases $R > 1$ and $R < 1$ separately. In practice however we would like to be able to smoothly change R from $R > 1$ to $R < 1$ and back in a seamless way (without clicks or other artifacts arising from an abrupt reconfiguration of a filter chain). This means that we need to find a way to distribute the roots of $P'(s)$ among 2nd-order factors in a continuous way, where there are no jumps in the values of the coefficients of these factors if R is varied in a continuous way. Formally saying, the coefficients of the 2nd-order factors must be continuous functions of R everywhere. The continuity for $R \neq 1$ should occur for granted, thus we are specifically concerned about continuity at $R = 1$.

First, let's assume the order of the transformation is even.

Let $R \geq 1$. There is an even count of the roots of $P'_1(s)$ and these roots come in complex-conjugate pairs (Fig. 8.19). Therefore each conjugate pair of roots of $P'_1(s)$ can be grouped into a single 2nd-order factor. The same can be done for $P'_2(s)$ and this half of our second-order factors corresponds to $P'_1(s)$ and the other half to $P'_2(s)$.

At $R = 1$ both sets of 2nd-order factors become identical, since $P'_1(s)$ becomes identical to $P'_2(s)$.

At $R < 1$ the roots of $P'_1(s)$ are rotated counterclockwise and the roots of $P'_2(s)$ are rotated clockwise (Fig. 8.21), therefore the roots of each of the polynomials won't combine into conjugate pairs and thus the polynomials won't be real anymore (Fig. 8.22).

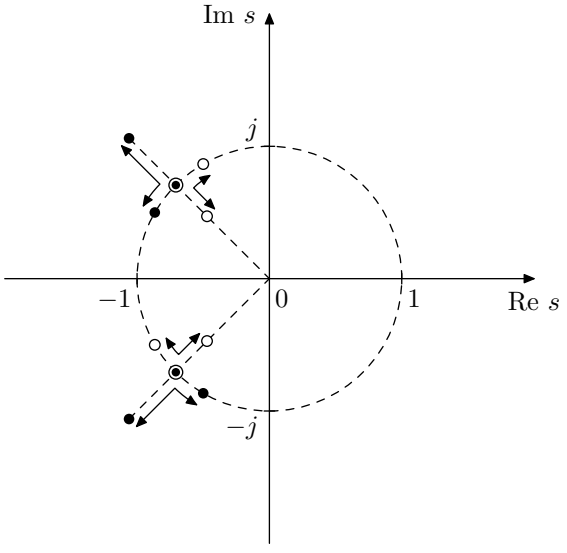


Figure 8.22: Movement of roots of $P'_1(s)$ (black dots) and $P'_2(s)$ (white dots) as R smoothly varies around $R = 1$.

However, since we started the rotation from two identical sets of roots with conjugate pairwise symmetry within each set, for each root of $P'_1(s)$ there is now a conjugate root in $P'_2(s)$ and vice versa. We can therefore formally redistribute the roots between $P'_1(s)$ and $P'_2(s)$ in such a way, that the roots $P'_1(s)$ will be

rotated by α/N *towards* the negative real semiaxis (compared to $R = 1$) and the roots $P_2'(s)$ will be rotated by α/N *away from* the negative real semiaxis (Fig. 8.23).

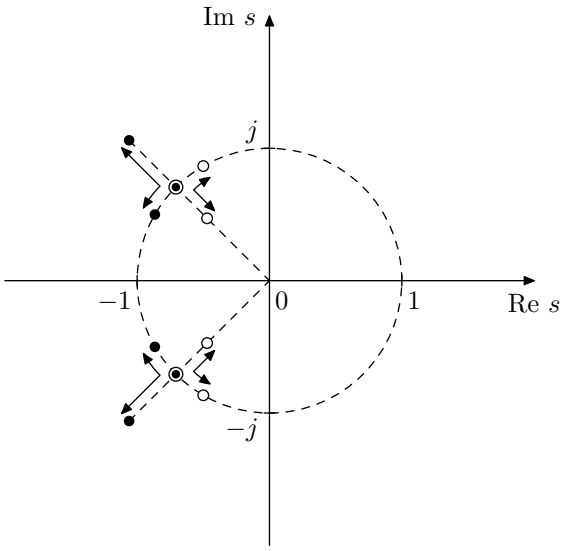


Figure 8.23: Movement of redistributed roots of $P_1'(s)$ (black dots) and $P_2'(s)$ (white dots) as R smoothly varies around $R = 1$.

Thus, at $R = 1$ we have two identical sets of roots. At $R > 1$ the roots of $P_1'(s)$ move outwards from the unit circle, at $R < 1$ the roots of $P_1'(s)$ move towards the negative real semiaxis. The roots of $P_2'(s)$ move inwards from the unit circle ($R > 1$) and away from the negative real semiaxis ($R < 1$). This way we can keep the same assignment of the roots to the 2nd-order factors.[14]

If the order of the transformation is odd, then besides the conjugate pairs that we just discussed, we get two "special" roots, corresponding to the purely real root of the Butterworth transformation of the 1st kind of $s + 1$ (Fig. 8.24). These two roots are real for $R \geq 1$ and complex conjugate for $R < 1$, where at $R = 1$ both roots are at -1. Thus, they can simply be assigned to one and the same 2nd-order factor of the form $s^2 + 2R's + 1$ (which cannot be formally assigned to $P_1'(s)$ or $P_2'(s)$, but can be thought of as being "shared" among $P_1'(s)$ and $P_2'(s)$), where R' depends on R.

Arbitrary 2nd-order polynomials

The non-unit-cutoff polynomials $P(s) = s^2 + 2Ras + a^2$ can be simply treated using (8.12c).

The case $a = 0$ can be taken in the limiting sense $a \to 0$ giving $P'(s) = s^{2N}$.

The case $R = 0$ also can be taken in the limiting sense $R \to +0$.

The case $R < 0$ can be treated by noticing that the amplitude responses of $P(s) = s^2 + 2Ras + a^2$ and $P(s) = s^2 - 2Ras + a^2$ are identical. Thus, we

[14] Of course we could have done the opposite redistribution of roots among $P_1'(s)$ and $P_2'(s)$, where the roots of $P_1'(s)$ move outwards from the unit circle and away from the negative real semiaxis, while the roots of $P_1'(s)$ move inwards from the unit circle and towards the negative real semiaxis.

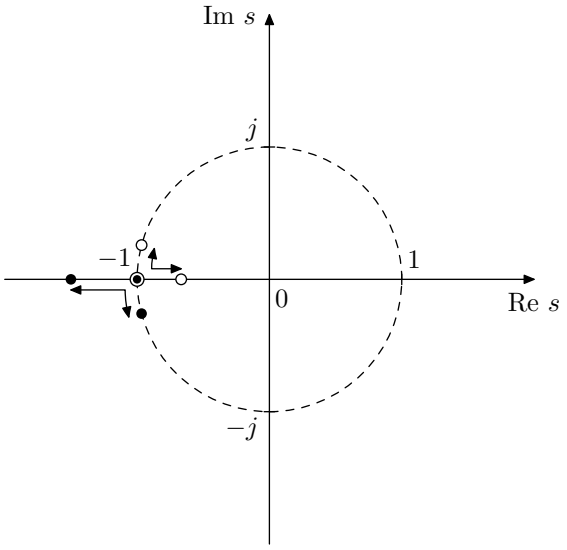


Figure 8.24: Movement of the "special" root of $P'_1(s)$ (black dot) and the "special" root of $P'_2(s)$ (white dot) as R smoothly varies around $R = 1$.

can apply the Butterworth transformation to the positive-damping polynomial $P(s) = s^2 - 2Ras + a^2$. Since the roots of $P(s)$ in case of $R < 0$ lie in the right complex semiplane, we might as well pick the right semiplane roots for $P'(s)$. Particularly, if $P(s)$ is the numerator of a maximum phase filter, the transformation result will retain the maximum phase property.

The polynomial of the most general form $P(s) = a_2 s^2 + a_1 s + a_0$ can be treated by rewriting it as $P(s) = a_2 \cdot (s^2 + (a_1/a_2)s + a_0/a_2)$ where usually $a_2 \neq 0$, $a_0/a_2 > 0$. If $a_0/a_2 < 0$ then $P(s)$ has two real roots of opposite sign and can be handled as a product of two 1st-order polynomials, to which we can apply Butterworth transformation of the 1st kind. If $a_2 = 0$, we can treat this as a limiting case $a_2 \to 0$. Noticing that at $a_2 \to 0$ the damping $a_1/2a_2 \to \infty$ we rewrite $P(s)$ as a product of real 1st-order terms:

$$P(s) = a_2 \cdot \left(s + \frac{a_1 + \sqrt{a_1^2 - 4a_2a_0}}{2a_2} \right) \cdot \left(s + \frac{a_1 - \sqrt{a_1^2 - 4a_2a_0}}{2a_2} \right) \sim$$
$$\sim (a_2 s + a_1) \cdot (s + a_0/a_1) \qquad (\text{for } a_2 \to 0)$$

and as a_2 vanishes we discard the infinitely large root of the polynomial $a_2 s + a_1$ (and the associated roots of $P'(s)$), formally replacing the polynomial $a_2 s + a_1$ with the constant factor a_1.

Lowpass Butterworth filter of the 2nd kind

Given

$$H(s) = \frac{1}{1 + 2Rs + s^2}$$

and transforming its denominator according to the previous discussion of the Butterworth transformation of a 2nd order polynomial we obtain

$$H'(s) = \prod_n \frac{1}{s^2 + 2R_n s + 1}$$

Therefore $H'(s)$ can be implemented as a series of 2-pole lowpass filters.

Fig. 8.25 compares the amplitude response of a Butterworth lowpass filter of the 2nd kind ($N = 2$) against the prototype resonating 2-pole lowpass filter. Note the increased steepness of the cutoff slope and the fact that the resonance peak height is preserved by the transformation.

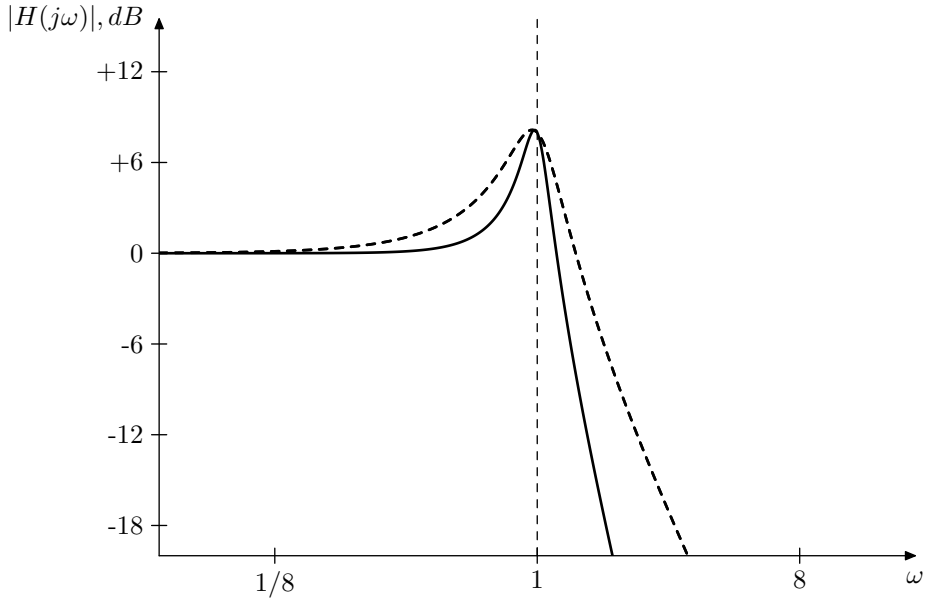


Figure 8.25: 4th order lowpass Butterworth filter of the 2nd kind vs. 2nd order lowpass filter (dashed line).

Fig. 8.26 compares the same Butterworth lowpass filter against cascading of identical 2nd order lowpasses, where the resonance has been adjusted to maintain the same resonance peak height. Note the much larger width of the resonance peak of the latter.

As mentioned before in the discussion of the Butterworth transformation of the 2nd kind, at $R = 1/\sqrt{2}$ we get the same set of poles as for order N Butterworth transformation of the 1st kind. Thus, at this resonance setting our lowpass Butterworth filter of the 2nd kind (the filter order of which is $2N$) is equal to the lowpass Butterworth filter of the 1st kind of the same filter order $2N$.

Highpass Butterworth filter of the 2nd kind

The highpass Butterworth filter of the 2nd kind is obtained from

$$H(s) = \frac{s^2}{1 + 2Rs + s^2}$$

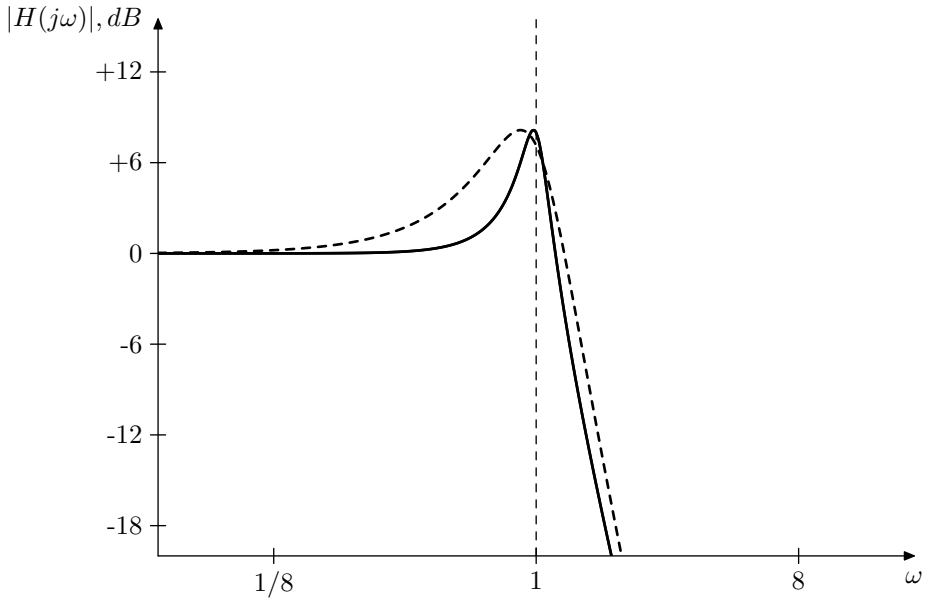


Figure 8.26: 4th order lowpass Butterworth filter of the 2nd kind vs. duplicated 2nd order lowpass filter of the same resonance peak height (dashed line).

resulting in

$$H'(s) = \prod_n \frac{s^2}{s^2 + 2R_n s + 1}$$

Therefore $H'(s)$ can be implemented as a series of 2-pole highpass filters. Fig. 8.27 shows the respective amplitude response.

As with lowpass Butterworth filter of the 1st kind, the highpass Butterworth filter of the 2nd kind can be equvalently obtained by applying the LP to HP substitution to a lowpass Butterworth filter of the 2nd kind.

As with lowpass Butterworth filter of the 2nd kind, at $R = 1/\sqrt{2}$ we get a highpass Butterworth filter of the 1st kind.

Bandpass Butterworth filter of the 2nd kind

The bandpass Butterworth filter of the 2nd kind is obtained from

$$H(s) = \frac{s}{1 + 2Rs + s^2}$$

resulting in

$$H'(s) = \prod_n \frac{s}{s^2 + 2R_n s + 1}$$

Therefore $H'(s)$ can be implemented as a series of 2-pole bandpass filters. Differently from the bandpass Butterworth filter of the 1st kind, this one allows to control the amount of resonance. Fig. 8.28 shows the respective amplitude response.

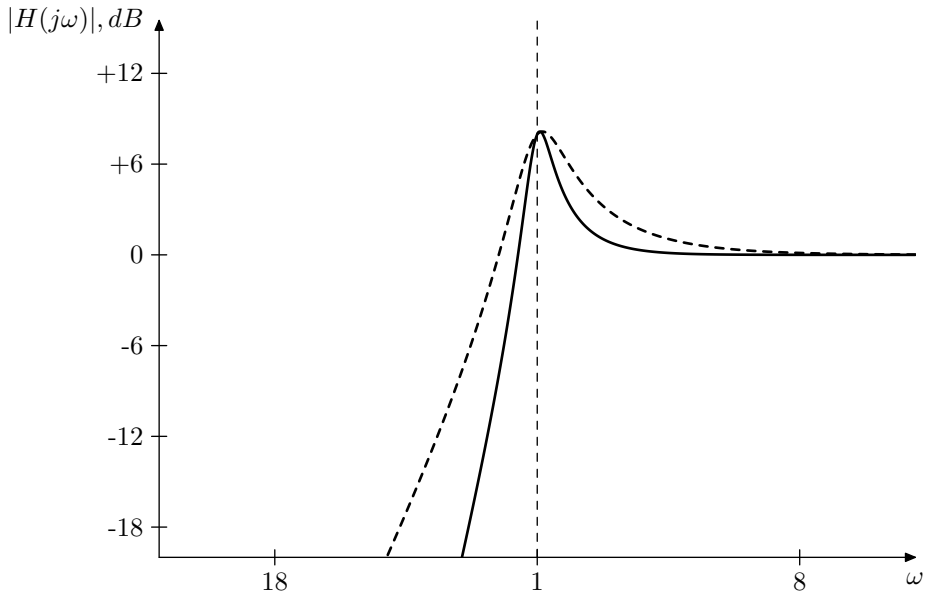


Figure 8.27: 4th order highpass Butterworth filter of the 2nd kind vs. 2nd order highpass filter (dashed line).

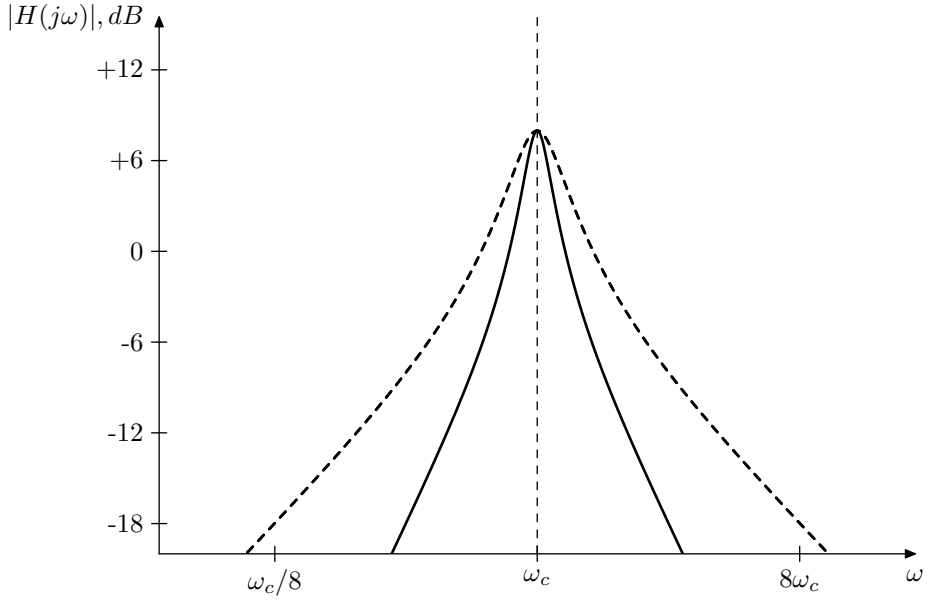


Figure 8.28: 4th order bandpass Butterworth filter of the 2nd kind vs. 2nd order bandpass filter (dashed line).

As with lowpass and highpass, at $R = 1/\sqrt{2}$ we get a bandpass Butterworth filter of the 1st kind. Since bandpass Butterworth filter of the 1st kind occurs only for an even transformation order $2N$, any bandpass Butterworth filter of the 1st kind is simply a bandpass Butterworth filter of the 2nd kind (of the

same filter order $2N$) at a particular resonance setting.

By replacing the underlying 2-pole bandpasses with their normalized versions one can obtain the normalized bandpass Butterworth filter of the 2nd kind:

$$H'(s) = \prod_n \frac{2R_n s}{s^2 + 2R_n s + 1}$$

One can of course also apply the LP to BP substitution to a Butterworth lowpass of the 2nd kind. Note, however, that if the lowpass has a resonance peak, then the resulting bandpass will have two of those, so this would be a rather special kind of a bandpass.

SUMMARY

We have introduced three general topology classes: the generalized SVF, the serial cascade form, and the parallel form. These topologies can be used to implement almost any transfer function (with the most prominent restriction being that the parallel form can't deal with repeated poles).

We also introduced two essentially different ways to obtain a higher-order filter from a given filter of a lower order: identical filter cascading and Butterworth transformation. The former is stretching the amplitude and phase responses vertically (which may cause a number of unwanted effects), while the latter is shrinking the amplitude response horizontally.

Chapter 9

Classical signal procesing filters

In Chapter 8 we have introduced, among other ideas, Butterworth filters of the 1st kind. In this chapter we are going to construct further similar filter types by allowing the amplitude response to have ripples of equal amplitude (a.k.a. *equiripples*) in the pass- or stop-band, or in both. These filters as well as Butterworth filters of the 1st kind are the filter types used in classical signal processing. They have somewhat less prominent role in music DSP, therefore we first concentrated on other filter types. Still, they are occasionally useful.

9.1 Riemann sphere

Before we begin discussing equiripple filters we need to go into some detail of complex algebra, concerning the Riemann sphere and some derived concepts. The key feature of the Riemann sphere is that infinity is treated like any other point, and this (among with some other possibilities arising out of using the Riemann sphere) will be quite helpful in our discussions. It seems there are a number of slightly different conventions regarding the Riemann sphere. We are going to introduce now one particular convention which will be most useful for our purposes.[1]

Given a complex plane $w = u + jv$ we introduce the third dimension, thereby embedding the plane into the 3-dimensional space (x, y, z). The x and y axes coincide with u and v axes, the z axis is directed upwards. The Riemann sphere will be the sphere of a unit radius $x^2 + y^2 + z^2 = 1$ (Fig. 9.1). Thus, the intersection of the Riemann sphere with the complex plane is at the "equator" which in terms of w is simply the complex unit circle $|w| = 1$. The center of projection will be at the "north pole" $(0, 0, 1)$ of the Riemann sphere, which thereby is the image of $w = \infty$. Respectively, the complex unit circle $|w| = 1$ coincides with its own projection image on the Riemann sphere.

We will denote and refer to the points on the Riemann sphere by the complex values that they represent. E.g. the "north pole" will be simply denoted as ∞,

[1] The general discussion of the idea of the Riemann sphere is not a part of this book. Readers unfamiliar with this concept are advised to consult the respective literature.

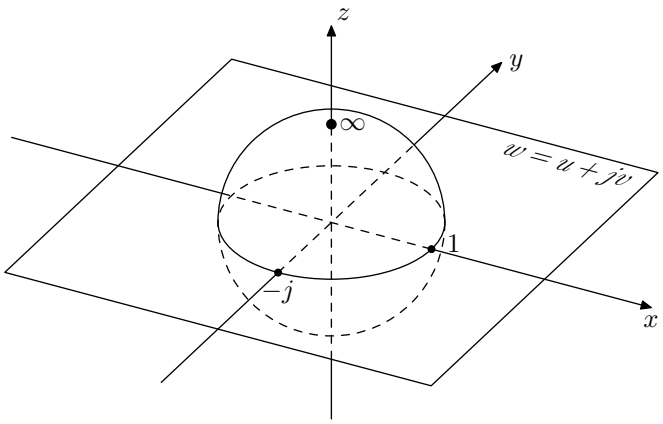


Figure 9.1: Riemann sphere.

the "south pole" as 0, the point on the "zero meridian" as 1, the point on the 90° meridian as j etc. Some of these points are shown on Fig. 9.1.

Real Riemann circle

The 2-dimensional subspace (x, z) of the (x, y, z) space in Fig. 9.1 contains just the real axis u of the complex plane and the real axis's image on the Riemann sphere which is a circle of unit radius $x^2 + z^2 = 1$ (Fig. 9.2). It will be intuitive to refer to this circle as the *real Riemann circle*.

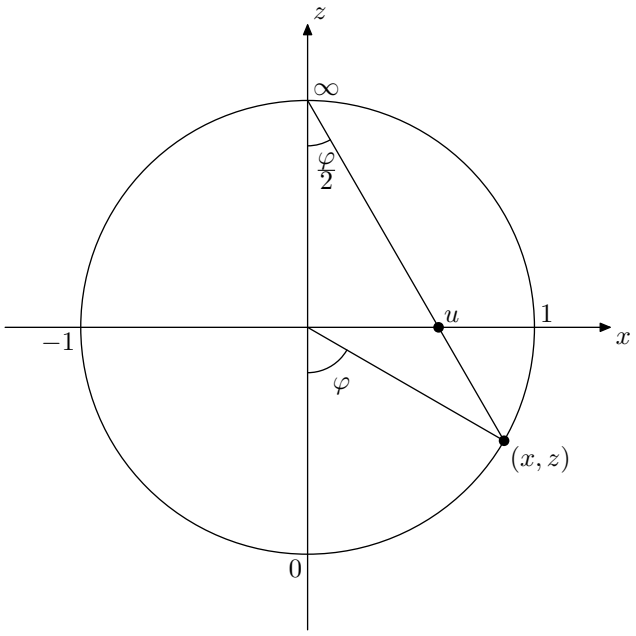


Figure 9.2: Real Riemann circle. The 0, 1, ∞, -1 labels denote the points on the circle which correspond to these values.

We can use the polar angle φ (defined as shown in Fig. 9.2) as the coordinate on the Riemann circle. One of the reasons for this choice of definition of φ are the following convenient mappings between u and φ:

$$\begin{aligned} u = 0 &\iff \varphi = 2\pi n \\ u = 1 &\iff \varphi = \frac{\pi}{2} + 2\pi n \\ u = -1 &\iff \varphi = -\frac{\pi}{2} + 2\pi n \\ u = \infty &\iff \varphi = \pi + 2\pi n \end{aligned}$$

Also, if we restrict φ to $(-\pi, \pi)$, then

$$\begin{aligned} u = 0 &\iff \varphi = 0 \\ u > 0 &\iff \varphi > 0 \\ u < 0 &\iff \varphi < 0 \end{aligned}$$

From Fig. 9.2, using some basic geometry it's not difficult to find that u and φ are related as

$$u = \tan\frac{\varphi}{2} \tag{9.1}$$

and that

$$u = \frac{x}{1-z} \tag{9.2a}$$

By introducing the "homogeneous" coordinate $\bar{z} = 1 - z$ the equation (9.2a) can be rewritten in a more intuitive form:

$$u = x/\bar{z} \tag{9.2b}$$

Conversely, using Fig. 9.2 and equation (9.1) we have

$$x = \sin\varphi = \frac{2u}{u^2+1} \tag{9.3a}$$

$$z = -\cos\varphi = \frac{u^2-1}{u^2+1} \tag{9.3b}$$

$$\bar{z} = \frac{2}{u^2+1} \tag{9.3c}$$

Symmetries on the real Riemann circle

Certain symmetries between a pair of points on the real axis correspond to symmetries on the real Riemann circle. Specifically, from (9.1) we obtain:

$$u_1 + u_2 = 0 \iff \varphi_1 + \varphi_2 = 2\pi n \tag{9.4a}$$

$$u_1 u_2 = 1 \iff \varphi_1 + \varphi_2 = \pi + 2\pi n \tag{9.4b}$$

$$u_1 u_2 = -1 \iff \varphi_1 - \varphi_2 = \pi + 2\pi n \tag{9.4c}$$

or, by restricting φ_1 and φ_2 to $[-\pi, \pi]$

$$u_1 + u_2 = 0 \iff \varphi_1 + \varphi_2 = 0 \tag{9.5a}$$

$$u_1 u_2 = 1 \iff \frac{\varphi_1 + \varphi_2}{2} = \pm\frac{\pi}{2} \tag{9.5b}$$

$$u_1 u_2 = -1 \iff \varphi_1 - \varphi_2 = \pm\pi \tag{9.5c}$$

Fig. 9.3 illustrates.

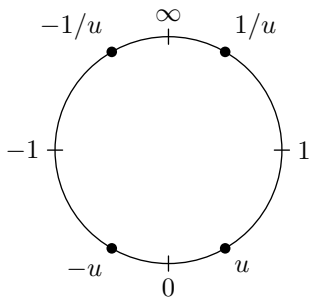


Figure 9.3: Symmetries of the values on the real Riemann circle.

Coordinate relationships for the Riemann sphere

The equations (9.2) and (9.3) generalize to the 3-dimensional space (x, y, z) containing the complex plane $w = u+jv$ and the Riemann sphere $x^2+y^2+z^2 = 1$ in an obvious way as:

$$u = \frac{x}{1-z} = x/\bar{z} \tag{9.6a}$$

$$v = \frac{y}{1-z} = y/\bar{z} \tag{9.6b}$$

$$w = u + jv = \frac{x+jy}{\bar{z}} \tag{9.6c}$$

and

$$x = \frac{2u}{|w|^2+1} \tag{9.7a}$$

$$y = \frac{2v}{|w|^2+1} \tag{9.7b}$$

$$z = \frac{|w|^2-1}{|w|^2+1} \tag{9.7c}$$

$$\bar{z} = \frac{2}{|w|^2+1} \tag{9.7d}$$

$$\tag{9.7e}$$

The equation (9.1) can be generalized if we restrict φ to $[0, \pi]$, in which case

$$|w| = \tan\frac{\varphi}{2} \tag{9.8}$$

In principle we could also introduce the spherical azimuth angle, which is simply equal to $\arg w$, but we won't do it in this book.

Imaginary Riemann circle

The (y, z) subspace of the (x, y, z) space in Fig. 9.1 contains just the imaginary axis v of the complex plane and the imaginary axis's image on the Riemann sphere which is a circle of unit radius $x^2 + y^2 = 1$. We will refer to this circle as the *imaginary Riemann circle*.

There are no essential differences to the real Riemann circle. The same illustrations and formulas hold, except that we should use v in place of u. Fig. 9.4 provides a simple illustration of the circla and its symmetries. Note that the reciprocal symmetries change sign if expressed in terms of the complex variable w, since $1/jv = -j/v$.

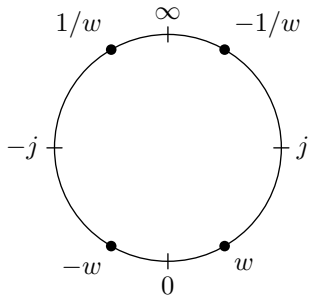


Figure 9.4: Symmetries of the values on the imaginary Riemann circle, where $w = jv$, $v = \operatorname{Im} w$.

9.2 Arctangent scale

From the Riemann circle one can derive a special scale which will be useful for plotting function graphs with interesting behavior around infinity. One commonly known special scale is a logarithmic scale $x' = \log x$ which maps the logical values x to the geometric positions x' on the plot. In a similar fashion, we introduce the *arctangent scale*

$$x' = 2 \arctan x \tag{9.9}$$

which is using the polar angle φ of the real Riemann circle as the geometric position x'. It is easy to notice that (9.9) is equivalent to (9.1), where we have x in place of u and x' in place of φ.

The actangent scale warps the entire real axis $(-\infty, +\infty)$ into the range $(-\pi, \pi)$. Due to the periodic nature of the Riemann circle's polar angle it is not unreasonable to require the scale x' to be periodic as well, in which case we can also support the value $x = \infty$ which will map to $\pi + 2\pi n$.

Treating the infinity like any other point, the arctangent scale provides a convenient means for plotting the functions where the range of the values of interest includes infinity. E.g. we could plot the graph of the cosecant function $y = \csc x = 1/\sin x$ using the arctangent scale for the function's value axis, as illustrated by Fig. 9.5

Symmetries in the arctangent scale

The symmetries of a graph plotted in the arctangent scale are occurring in agreement with Riemann circle symmetries (9.4) and (9.5) (illustrated in Fig. 9.3). Specifically:

1. Mutually opposite values are symmetric with respect to points 0 and ∞.

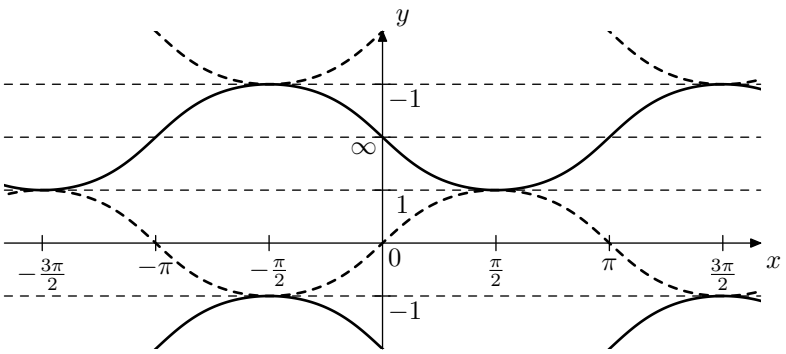


Figure 9.5: Graphs of $y = \sin x$ (dashed) and $y = \csc x = 1/\sin x$ (solid) using arctangent scale for the ordinate.

2. Mutually reciprocal values are symmetric with respect to points 1 and -1.

3. Values whose product is -1 map are spaced by the distance equal to the half of the arctangent scale's period, that is e.g. to the distance between -1 and 1 or between 0 and ∞.

One can observe all of these properties if Fig. 9.5, where the third property can be observed between the $\csc x$ and the half-period-shifted $\sin x$.

9.3 Rotations of Riemann sphere

We are going to introduce two special transformations of the complex plane:

$$\rho_{+1}(w) = \frac{1+w}{1-w} \tag{9.10a}$$

$$\rho_{-1}(w) = \frac{w-1}{w+1} \tag{9.10b}$$

where $w \in \mathbb{C} \cup \infty$. It is easy to check by direct substitution that $\rho_{-1}(\rho_{+1}(w)) = \rho_{+1}(\rho_{-1}(w)) = w$, that is the transformations ρ_1 and ρ_{-1} are each other's inverses.[2] As we shall see, $\rho_{\pm 1}$ are simply rotations of the Riemann sphere by 90° in two opposite directions.

Letting $w = u + jv$ where u and v are the real and imaginary parts of w, we have

$$w' = u' + jv' = \rho_{+1}(w) = \frac{1+w}{1-w} = \frac{1+(u+jv)}{1-(u+jv)} = \frac{(1+u)+jv)}{(1-u)+jv} =$$
$$= \frac{\big((1+u)+jv\big)\big((1-u)+jv\big)}{\big((1-u)^2+v^2\big)} = \frac{(1-u^2-v^2)+2jv}{1+u^2+v^2-2u} = \frac{(1-|w|^2)+2jv}{1+|w|^2-2u}$$

That is

$$u' = \frac{1-|w|^2}{1+|w|^2-2u}$$

[2] One could also notice that (9.10) are very similar to the bilinear transform and its inverse, where the latter two have an additional scaling by $T/2$.

$$v' = \frac{2v}{1+|w|^2-2u}$$

On the other hand,

$$|w'|^2 = \left|\frac{1+w}{1-w}\right|^2 = \frac{|1+w|^2}{|1-w|^2} = \frac{(1+u)^2+v^2}{(1-u)^2+v^2} = \frac{1+|w|^2+2u}{1+|w|^2-2u}$$

and

$$|w'|^2+1 = \frac{(1+|w|^2+2u)+(1+|w|^2-2u)}{1+|w|^2-2u} = 2\frac{1+|w|^2}{1+|w|^2-2u}$$

from where by (9.7d)

$$\bar{z}' = \frac{2}{|w'|^2+1} = \frac{1+|w|^2-2u}{1+|w|^2}$$

Then, using (9.7) we obtain

$$\begin{aligned} x' &= \bar{z}'u' = \frac{1-|w|^2}{1+|w|^2} = -z \\ y' &= \bar{z}'v' = \frac{2v}{1+|w|^2} = y \\ z' &= 1-\bar{z}' = \frac{(1+|w|^2)-(1+|w|^2-2u)}{1+|w|^2} = \frac{2u}{1+|w|^2} = x \end{aligned}$$

That is $x' = -z$, $y' = y$, $z' = x$ which is simply a rotation by 90° around the y axis in the direction from the (positive) x axis towards the (positive) z axis. Thus ρ_{+1} simply rotates the Riemann sphere around the imaginary axis of the complex plane w by 90° in the direction from 1 to ∞, or, which is the same, in the direction from 0 to 1 (where by 0, 1 and ∞ we mean the points on the Riemann sphere which are the projection images of $w = 0$, $w = 1$ and $w = \infty$ respectively). The points $\pm j$ are thereby untouched, which can be also seen by directly evaluating $\rho_{+1}(\pm j) = \pm j$. The transformation ρ_{-1} being the inverse of ρ_{+1} simply rotates in the opposite direction.

In terms of the real Riemann circle $\rho_{\pm 1}$ clearly correspond to a counterclockwise (for ρ_{+1}) or clockwise (for ρ_{-1}) rotation by 90° (Fig. 9.6).[3] Respectively, in the arctangent scale they correspond to shifts by a quarter of the arctangent scale's period.

The imaginary Riemann circle is rotated into the unit circle $|w'| = 1$, which is its own image on the Riemann sphere. Therefore for ρ_{+1} the polar angle φ from Fig. 9.2 becomes the polar angle in the complex plane (since $\varphi = 0 \iff w = 0$ and since $\arg \rho_{+1}(0) = \arg 1 = 0$), therefore $\varphi = \arg \rho_{+1}(w)$ and by (9.1) we have

$$\rho_{+1}\left(j \tan \frac{\varphi}{2}\right) = e^{j\varphi}$$

This can also be verified by direct substitution, where it's easier to use the inverse transformation:

$$\rho_{-1}(e^{j\varphi}) = = \frac{e^{j\varphi}-1}{e^{j\varphi}+1} = \frac{e^{j\varphi/2}-e^{-j\varphi/2}}{e^{j\varphi/2}+e^{-j\varphi/2}} =$$

[3] This is also the reason for the notation ρ_{+1}: the subscript simply indicates the result of the transformation of $w = 0$.

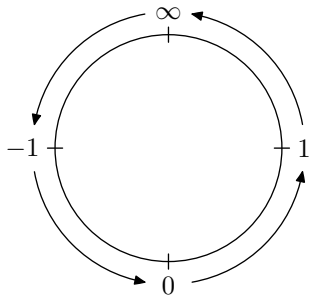


Figure 9.6: Transformation of the real Riemann circle by ρ_{+1}.

$$= j\frac{e^{j\varphi/2} - e^{-j\varphi/2}}{2j} \cdot \frac{2}{e^{j\varphi/2} + e^{-j\varphi/2}} = j\tan\frac{\varphi}{2}$$

For ρ_{-1} the polar angle φ gets mapped into $\arg\rho_{-1}(w) = \pi - \varphi$. Conversely a polar angle equal to $\pi - \varphi$ will be mapped into $\arg\rho_{-1}(w) = \varphi$, thus

$$e^{j\varphi} = \rho_{-1}\left(j\tan\frac{\pi-\varphi}{2}\right) = \rho_{-1}\left(j\tan\left(\frac{\pi}{2} - \frac{\varphi}{2}\right)\right) = \rho_{-1}\left(\frac{j}{\tan\frac{\varphi}{2}}\right)$$

Summing up:

$$\rho_{+1}\left(j\tan\frac{\varphi}{2}\right) = e^{j\varphi} \tag{9.11a}$$

$$\rho_{-1}\left(\frac{j}{\tan\frac{\varphi}{2}}\right) = e^{j\varphi} \tag{9.11b}$$

$$\rho_{-1}\left(e^{j\varphi}\right) = j\tan\frac{\varphi}{2} \tag{9.12a}$$

$$\rho_{+1}\left(e^{j\varphi}\right) = \frac{j}{\tan\frac{\varphi}{2}} \tag{9.12b}$$

Note that (9.12) do not simply give the inverse versions of (9.11), but also reflect the fact the complex unit circle gets rotated into the imaginary Riemann circle.

Symmetries

The transformations of symmetries by $\rho_{\pm1}$ can be derived in an intuitive way from the symmetries on the real Riemann circle and the arctangent scale and from the fact that these transformations are $\pm90°$ rotations of the real Riemann circle or (equivalently) shifts of the arctangent scale by the scale's quarter period, if we assume $w \in \mathbb{R} \cup \infty$.

Particularly, a $\pm90°$ rotation of the real Riemann circle maps 0 and ∞ to ±1 and vice versa. Respectively, points on the Riemann circle which are symmetric relatively to 0 and ∞ map to points symmetric relatively to ±1 and vice versa. Thus, mutually opposite values map to mutually reciprocal values and vice versa:

$$\rho_{+1}(w)\rho_{+1}(-w) = 1 \tag{9.13a}$$

$$\rho_{-1}(w)\rho_{-1}(-w) = 1 \tag{9.13b}$$
$$\rho_{+1}(w) + \rho_{+1}(1/w) = 0 \tag{9.13c}$$
$$\rho_{-1}(w) + \rho_{-1}(1/w) = 0 \tag{9.13d}$$

Fig. 9.7 illustrates, where more properties are immediately visible. E.g. one could notice that $\rho_{\pm1}(w)$ are obtained by rotating w by $\pm90°$, therefore the results are 180° apart, which means:

$$\rho_{+1}(w)\rho_{-1}(w) = -1 \tag{9.14}$$

or that rotating the opposite points $\pm w$ by 90° in opposite directions produces opposite points:

$$\rho_{\pm1}(-w) = -\rho_{\mp1}(w) \tag{9.15}$$

etc.

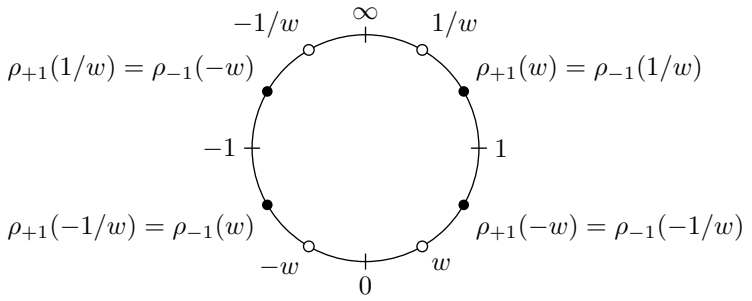


Figure 9.7: Symmetries of the transformations $\rho_{\pm1}$ on the real Riemann circle, where they represent 90° rotations.

The formulas (9.13), (9.14) and other similarly obtained symmetry-related properties of $\rho_{\pm1}$ work not only for $w \in \mathbb{R} \cup \infty$ but actually for any $w \in \mathbb{C} \cup \infty$ which can be verified algebraically. However, the interpretation of these symmetries in terms of the real Riemann circle obviously works only for $w \in \mathbb{R} \cup \infty$.

The imaginary Riemann circle gets rotated by $\rho_{\pm1}$ into the complex unit circle, where the results of the two transformations are thereby symmetric relatively to the imaginary axis. Fig. 9.8 illustrates some of the symmetries arising out of this rotation. More illustrations of this kind can be created, particularly for the transformation of the values lying on the complex unit circle, but the ones we are already having should be sufficient for our purposes in this book.

Unit circle rotations

The Riemann sphere rotations $\rho_{\pm1}$ can be described as rotations of the Riemann sphere around the imaginary axis or as rotations of the real Riemann circle. Similarly the Riemann sphere rotations around the vertical axis z can be thought of as rotations of the unit circle.

Apparently, such rotations are simply achived by a multiplications of complex values by a unit-magnitude complex constant, where we are not restricted to rotations by multiples of 90°. We won't need a special notation for this

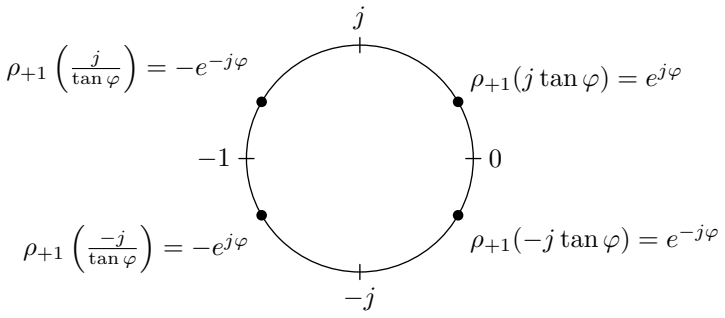


Figure 9.8: Symmetries of the results of the transformation ρ_{+1} of the imaginary axis (lying on the complex unit circle).

transformation and will simply write

$$w' = aw \qquad (|a| = 1)$$

Obviously

$$|w'| = |w|$$
$$\arg w' = \arg w + \arg a$$

The rotations by $+90°$ and $-90°$ are simply multiplications by j and $-j$ respectively.

There is not much more to say in this respect as this rotation is pretty trivial.

Imaginary rotations

We could also wish to rotate the imaginary Riemann circle. We will denote the respective transformations as $\rho_{\pm j}$, where the subscript, as with $\rho_{\pm 1}$, denotes the image of the zero after the transformation.

We could rotate the imaginary circle by doing the three steps in succession:

1. First, we rotate the Riemann sphere by $90°$ around its "vertical" axis, thereby turning the imaginary Riemann circle into the real Riemann circle, where $w = \pm j$ is transformed into $w = \pm 1$ respectively, while 0 and ∞ stay in place. Such rotation is simply achieved by multiplying w by $-j$.
2. Now we apply $\rho_{\pm 1}$ to rotate the real circle.
3. We convert the real circle back to the imaginary one by multiplying the rotation result by j.

Therefore we simply have

$$\rho_{\pm j}(w) = j\rho_{\pm 1}(-jw) = -j\rho_{\mp 1}(jw) \tag{9.16}$$

(where the second expression is obtained in the same way as the first one, except that we rotate the Riemann sphere in the other direction, both vertically and horizontally).

In order to distinguish between $\rho_{\pm 1}$ and $\rho_{\pm j}$ we could refer to the former as *real rotations* of the Riemann sphere and to the latter as *imaginary rotations* of the Riemann sphere.

9.4 Butterworth filter revisited

In Chapter 8 we have developed the lowpass Butterworth filters of the 1st kind (also simply known as Butterworth filters) as a Butterworth transformation of the 1-pole lowpass filter, where we also mentioned that the tranditional definition of the Butterworth filter simply defines the (lowpass) Butterworth filter as a (stable) filter whose amplitude response is

$$|H(j\omega)|^2 = \frac{1}{1+\omega^{2N}} \qquad (\omega \in \mathbb{R}) \tag{9.17}$$

Apparently both definitions are equivalent.

We could generalize this idea by replacing ω^N in (9.17) by some other polynomial function $f(\omega)$:

$$|H(j\omega)|^2 = \frac{1}{1+f^2(\omega)} \qquad (\omega \in \mathbb{R}) \tag{9.18}$$

The practical application of (9.18) essentially follows the steps of the Butterworth transformation of the 1st kind, which includes solving $1 + f^2 = 0$ to obtain the poles of $|H(s)|^2$ and then discarding the right-semiplane poles, thereby effectievely obtaining the desired transfer function $H(s)$ expressed in the cascade form. Note that there are two implicit conditions which need to be fulfilled in order for this procedure to work:

- There is the symmetry of the poles of $|H(s)|^2$ with respect to the real axis: if s is a pole then so is s^* (where s^* may be the same pole as s if s is real). This is necessary in order for $H(s)$ to be a real function of s.

- There is the pairwise symmetry of the poles $|H(s)|^2$ with respect to the imaginary axis: if s is a pole then $-s^*$ is *another* pole (it must be another pole even if $s = -s^*$, in which case it simply means that the pole is duplicated). This guarantees that we can split the poles into the left- and right-semiplane halves with identical contributions to the amplitude response. Therefore by discarding the right-semiplane half of the poles, we effectively go from $|H(j\omega)|^2$ to $|H(j\omega)|$.

We could expect that these properties will not be fulfilled for an arbitrary $f(\omega)$. However, let's require that

- The function $f(\omega)$ is a real function of ω.

- The function $f(\omega)$ is odd or even.

The readers can convince themselves that under these restrictions the poles of $|H(s)|^2$ defined by (9.18) will have the necessary symmetries with respect to the real and imaginary axes.

Rational $f(\omega)$

We could allow $f(\omega)$ to be not just a polynomial but a rational function. In this case $H(s)$ has not only poles, but also zeros at locations where $f(\omega)$ has poles and respectively the denominator of $|H(j\omega)|^2$ turns to ∞, which means that both $|H(j\omega)|^2$ and $|H(j\omega)|$ turn to zero. In order to see that the multiplicities

of the zeros of $H(s)$ are equal to the multiplicities of the respective poles of of $f(\omega)$ simply consider

$$|H(j\omega)|^2 = \frac{1}{1 + \dfrac{P_1^2(\omega)}{P_2^2(\omega)}} = \frac{P_2^2(\omega)}{P_1^2(\omega) + P_2^2(\omega)}$$

Thus the set of zeros of $|H(j\omega)|^2$ is the duplicated set of zeros of $P_2(\omega)$, however after switching to $H(j\omega)$ we should drop the duplicates, being left only with a single set of zeros of $P_2(\omega)$, which are the poles of $f(\omega)$. Note that $H(s)$ shouldn't have more zeros than poles, which means that the order of the denominator of $f(\omega)$ should not exceed the order of the numerator of $f(\omega)$.

In order for $H(s)$ to be real, its zeros must be conjugate symmetric, which will be ensured if $f(\omega)$ is real odd or even function. Indeed, in this case the poles of $f(\omega)$ are both conjugate symmetric and symmetric with respect to the origin, which implies that they are also symmetric with respect to the imaginary ω axis, which is the same as the symmetry with respect to the real s axis.

Representations of linear scaling

We are now going to develop another way of looking at the Butterworth filter generating function $f(x) = x^N$. It will be highly useful with other functions $f(x)$ occurring in place of $f(x) = x^N$ in (9.18).

Let $f(x) = x^N$. Since $x^N = \exp(N \ln x)$, we can write

$$f(x) = \exp(N \ln x) \tag{9.19}$$

Introducing auxiliary variables u and v we can rewrite (9.19) as a set of equations:

$$\begin{aligned} x &= \exp u \\ v &= Nu \\ f(x) &= \exp v \end{aligned}$$

which also allows to define $f(x)$ implicitly as a function satisfying the equation

$$f(\exp u) = \exp(Nu) \tag{9.20}$$

We could consider x and $f(x)$ as *representations* of u and v, where the connection between the *preimages* u and v and their respective representations x and $f(x)$ is achieved via the exponential mapping $x = \exp u$ (Fig. 9.9). In terms of preimage domain the function $f(x) = x^N$ is simply a multiplication by N.

Now consider that the function $\exp u$ is periodic in the imaginary direction:

$$\exp u = \exp(u + 2\pi j)$$

Therefore preimages are $2\pi j$-periodic, that is if u is a representation of x, then so is $u + 2\pi jn \ \forall n \in \mathbb{Z}$ (Fig. 9.10).

A multiplication by an integer in the preimage domain $v = Nu$ expands one period $\operatorname{Im} u \in [0, 2\pi]$ to N periods $\operatorname{Im} v \in [0, 2\pi N]$. Respectively the function $f(x)$ takes each value N times as x goes across all possible values in $\mathbb{C}$. Conversely, a division by an integer $u = v/N$ shrinks N periods to one period and

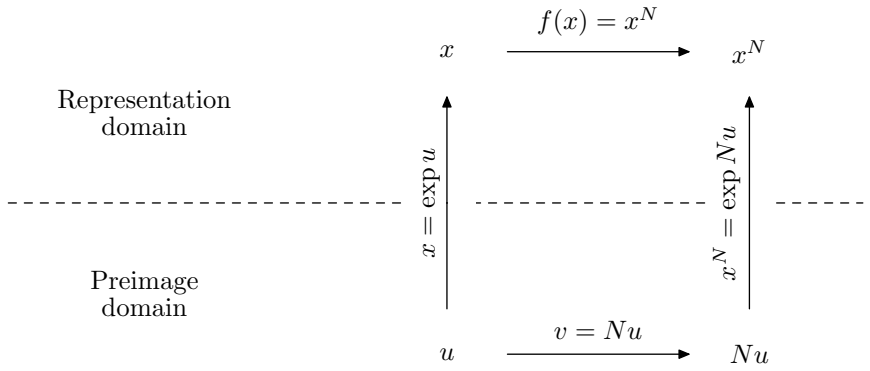


Figure 9.9: The preimage and representations domains.

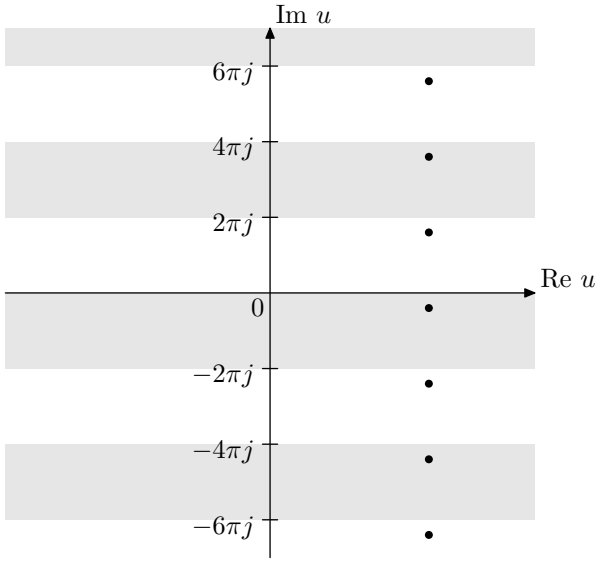


Figure 9.10: Periods of the preimage of the representation $x = \exp u$. Each strip (where there is no difference between gray and white strips) denotes a preimage of the entire complex plane with the exception of zero. The preimages denoted by the dots are preimages of one and the same value.

a previously single value of $f(x)$ turns into N different values of x. This is another possible way to explain the fact that the equation $x^N = a$ has N different solutions. Fig. 9.11 demonstrates the result of transformation of all preimages in Fig. 9.10 by a division by 2 corresponding to the equation $u = v/2$. Notice that the preimages in Fig. 9.11 correspond to two different representation values, while the peimages in Fig. 9.10 were corresponding to one and the same representation value.

The preimage of the real axis $x \in \mathbb{R}$ consists of two "horizontal" lines $\operatorname{Im} u = 0$ and $\operatorname{Im} u = \pi$, or, more precisely, of their periodic repetitions $\operatorname{Im} u = 2\pi n$ and $\operatorname{Im} u = \pi + 2\pi n$, where $n \in \mathbb{Z}$. The line $\operatorname{Im} u = 0$ (and its repetitions)

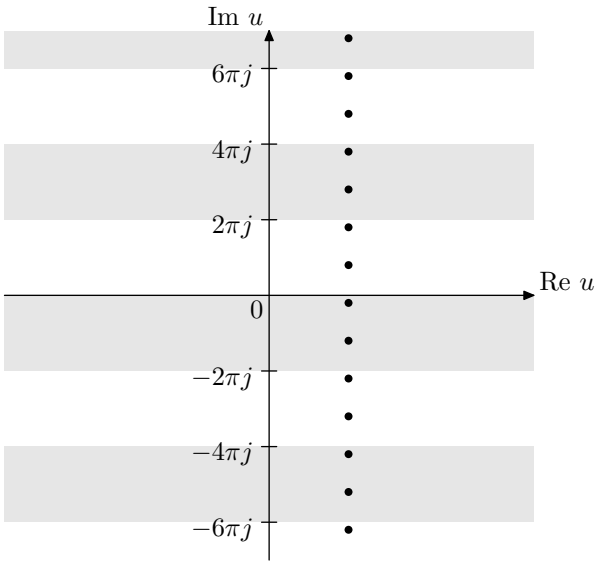


Figure 9.11: Transformation of the preimage points in Fig. 9.10 by division by 2.

corresponds to the positive real numbers, the line $\mathrm{Im}\, u = \pi$ (and its repetitions) corresponds to the negative real numbers. The preimage of the zero $x = 0$ exists only in the limiting sense $\mathrm{Re}\, u \to -\infty$. Thus the multiplication of u by N is mapping the preimage of the real axis onto itself, therefore this multiplication's representation x^N maps the real axis onto itself.

The "vertical" lines $\mathrm{Re}\, u = a$ are preimages of circles of radius e^a, where moving upwards along such lines corresponds to the counterclockwise movement along the respective circle (Fig. 9.12), Particularly the imaginary axis is the preimage of the unit circle (note that a section of such line extending over a single imaginary period $\mathrm{Im}\, u \in [b, b + 2\pi)$ is sufficient to generate all possible values on such circle). Multiplication by N maps the imaginary axis onto itself, thereby its representation x^N maps the unit circle onto itself. A single preimage period $[0j, 2\pi j]$ is thereby mapped to N periods $[0, 2\pi jN]$, which corresponds to the unit circle being mapped to itself "N times". We will shortly see that the mapping of the unit circle onto itself is the reason for the Butterworth poles being located on the unit circle.

Even/odd poles

The poles of (9.18) are given by $1 + f^2 = 0$, which can be rewritten as $f = \pm j$. In the Butterworth case the solutions of $f = \pm j$ were interleaved on the unit circle and also corresponded to even and odd values of n in the solution formula for $1 + f^2 = 0$, where $f = j$ was defining the even poles and $f = -j$ was defining the odd poles.

We will take effort to keep the same correspondence between the equations $f = \pm j$ and the even/odd values of n for other functions $f(\omega)$. In that regard it is instructive to first review the Butterworth case, but now using the just introduced linear scaling representation form, as it will then nicely generalize to

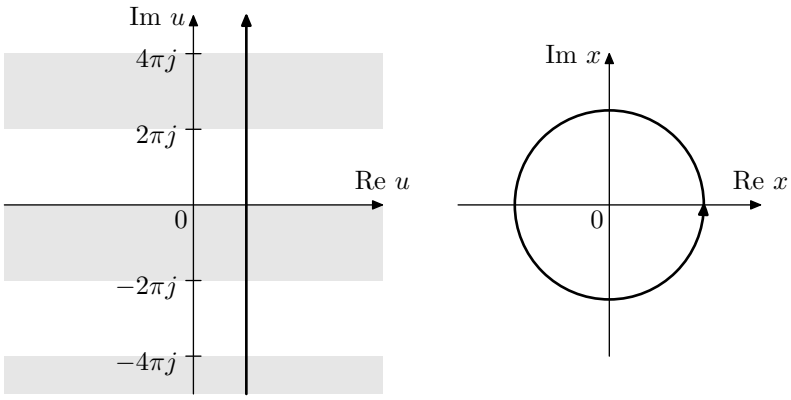


Figure 9.12: A circular trajectory and its preimage.

other $f(\omega)$ that we are going to use.

Let ω move along the unit circle in the counterclockwise direction. Its preimage u defined by $\omega = \exp u$ will respectively move upwards along the imaginary axis and so will $v = Nu$. Respectively $f(\omega) = \exp v$ moves along the unit circle in the counterclockwise direction, just N times faster, so while $f(\omega)$ completes one circle, ω will complete only $1/N$-th of a circle. The value of $f(\omega)$ will be passing through the points j and $-j$, since they are lying on the unit circle. At these moments the value of ω will be the solution of the equations $f(\omega) = j$ and $f(\omega) = -j$ respectively (Figs. 9.13 and 9.14). There will be no other solutions since if ω moves in a circle of any other radius, this circle will map to a circle of a non-unit radius and $f(\omega)$ will not go through the points $\pm j$. Thus Fig. 9.14 contains the full set of Butterworth poles, where the interleaving of the white/black dots on the circle in Fig. 9.14 arises from the interleaving of the white/black dots on the circular tajectory in Fig. 9.13.

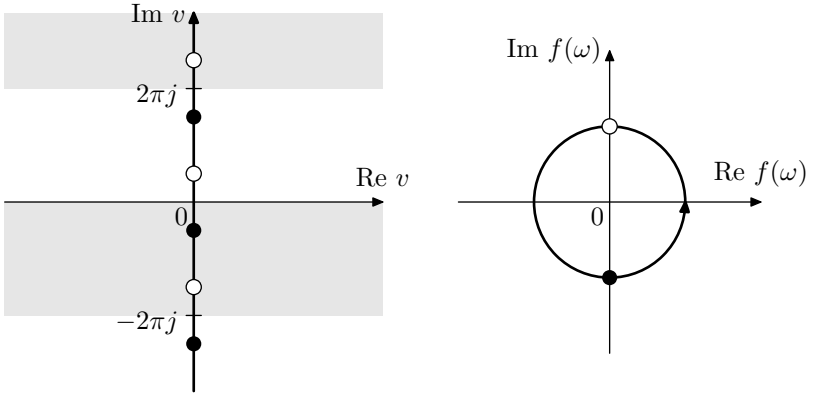


Figure 9.13: $f(\omega)$ moving in a unit-radius circular trajectory, the points $f(\omega) = \pm j$ and their preimages.

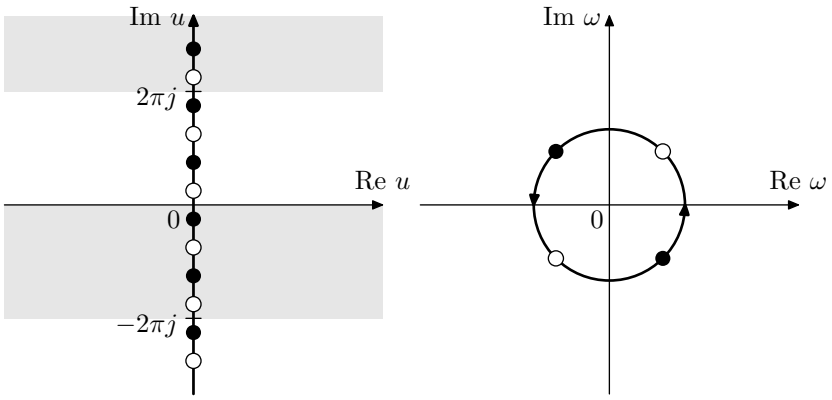


Figure 9.14: Transformation of Fig. 9.13 by $u = v/N$ (for $N = 2$). The white and black dots on the circle are even/odd Butterworth poles in terms of ω.

Apparently, the passing of $f(\omega)$ through $\pm j$ corresponds to

$$v = j\pi \left(\frac{1}{2} + n \right)$$

where $f(\omega) = j$ occurs at even n and $f(\omega) = -j$ occurs at odd n. The value of u at these moments is

$$u = j\pi \frac{\frac{1}{2} + n}{N}$$

and the value of ω is

$$\omega = \exp\left(j\pi \frac{\frac{1}{2} + n}{N} \right)$$

which is pretty much the same as the expression (8.13) we have developed before. The even/odd values of n still correspond to the solutions of $f = j$ and $f = -j$ respectively and thus we have a constency in referring to the solutions of $f = j$ as even poles and to the solutions of $f = -j$ as odd poles.

Lowpass, bandpass and highpass filters

If we want $H(s)$ in (9.18) to be a (unit cutoff) lowpass filter, then we should impose some additional requirements on $f(\omega)$:

$$\begin{aligned} f(\omega) &\approx 0 \qquad \text{for } \omega \ll 1 \\ f(\omega) &\approx \infty \qquad \text{for } \omega \gg 1 \end{aligned} \tag{9.21}$$

where around $\omega = 1$ the absolute magnitude of $f(\omega)$ should smoothly grow from 0 to ∞. Apparently the Butterworth filter's function $f(\omega) = \omega^N$ satisfies these requirements.

Similarly to Butterworth filter, with other filter types arising from (9.18) we will not be constructing $f(x)$ which give a highpass or bandpass response. Instead, highpass and bandpass filters can be simply obtained by LP to HP and LP to BP substitutions respectively.

9.5 Trigonometric functions on complex plane

The trigonometric functions, such as $\sin x$, $\cos x$, $\tan x$ and so on can be evaluated for complex argument values. In that regard they occur to be closely related to the hyperbolic functions $\sinh x$, $\cosh x$, $\tanh x$ and so on. The extention to $x \in \mathbb{C}$ can be obtained by simply evaluating the formulas

$$\cosh x = \frac{e^x + e^{-x}}{2} \tag{9.22}$$

$$\sinh x = \frac{e^x - e^{-x}}{2} \tag{9.23}$$

$$\tanh x = \frac{\sinh x}{\cosh x} \tag{9.24}$$

$$\cos x = \frac{e^{jx} + e^{-jx}}{2} = \cosh jx \tag{9.25}$$

$$\sin x = \frac{e^{jx} - e^{-jx}}{2j} = -j \sinh jx \tag{9.26}$$

$$\tan x = \frac{\sin x}{\cos x} = \frac{1}{j} \cdot \frac{e^{jx} - e^{-jx}}{e^{jx} + e^{-jx}} = -j \tanh jx \tag{9.27}$$

etc., where the function e^x is allowed to take complex argument values. Notice that thereby we immediately obtain the "imaginary argument properties":

$$\sin jx = j \sinh x \tag{9.28a}$$

$$\cos jx = \cosh x \tag{9.28b}$$

$$\sinh jx = j \sin x \tag{9.28c}$$

$$\cosh jx = \cos x \tag{9.28d}$$

etc., where intuitively we assume $x \in \mathbb{R}$, but the formulas also work for $x \in \mathbb{C}$.

By direct evaluation one could verify that all basic properties and fundamental trigonometric and hyperbolic identities continue to hold in complex domain. Particularly

$$\begin{aligned}
\sin(-x) &= -\sin x \\
\cos(-x) &= \cos x \\
\cos(x + 2\pi) &= \cos x \\
\cos(x - \pi/2) &= \sin x \\
\sin^2 x + \cos^2 x &= 1 \\
\sinh(-x) &= -\sinh x \\
\cosh(-x) &= \cosh x \\
\cosh^2 x - \sinh^2 x &= 1
\end{aligned}$$

etc. Also, apparently, conjugation commutes with the respective functions:

$$\begin{aligned}
\sin x^* &= (\sin x)^* \\
\cos x^* &= (\cos x)^* \\
\sinh x^* &= (\sinh x)^* \\
\cosh x^* &= (\cosh x)^*
\end{aligned}$$

etc.

A direct corollary of (9.28) and the trigonometric formulas for the sum of arguments are the formulas allowing to express a trigonometric function a complex argument via the real and imaginary parts of the argument. E.g.

$$\cos(u+jv) = \cos u \cos jv - \sin u \sin jv = \cosh v \cos u - j \sinh v \sin u \quad (9.29a)$$

$$\sin(u+jv) = \sin u \cos jv + \cos u \sin jv = \cosh v \sin u + j \sinh v \cos u \quad (9.29b)$$

etc.

Periodicity

Since the periodicity property is retained in the complex domain, the former real periods of the trigonometric functions turn into strips on the complex plane. E.g. the 2π periods of $\cos x$ are shown in Fig. 9.15.

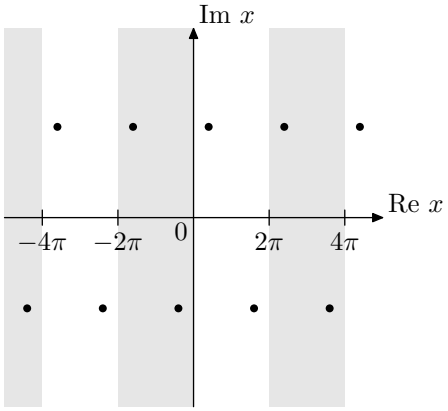


Figure 9.15: Periods of $\cos x$ in the complex plane. All dots are preimages of one and the same value.

Due to the even symmetry of the cosine, almost every value occurs twice on a period (as illustrated by the dots in Fig. 9.15). That is if the value y occurs at x (that is $y = \cos x$), then y also occurs at $-x$. The exceptions are being $\cos x = 1$ and $\cos x = -1$, which are mapped to themselves by $x \leftarrow -x$ if the periodicity of $\cos x$ is taken into account.

Inverse functions

Inverting (9.22) and (9.23) we obtain

$$\cosh^{-1} x = \ln\left(x \pm \sqrt{x^2-1}\right)$$

$$\sinh^{-1} x = \ln\left(x \pm \sqrt{x^2+1}\right)$$

The principal value of the complex square root is defined as

$$\sqrt{x} = \exp \frac{\ln x}{2} = \sqrt{|x|} \cdot \exp j \frac{\arg x}{2} \quad (9.30)$$

where $\arg x \in [-\pi, \pi]$,[4] in which case $\operatorname{Re}\sqrt{x} \geq 0\ \forall x \in \mathbb{C}$ thereby (9.30) is a generalization of arithmetic square root of real argument to complex domain. Notice that (9.30) gives the values on the upper imaginary semiaxis for real $x < 0$ (provided $\arg x = \pi$ for $x < 0$).

The principal value of $\ln x$ is defined in the usual way:

$$\ln x = \ln |x| + j \arg x$$

Respectively we can introduce the principal values of $\cosh^{-1}$ and $\sinh^{-1}$ as:

$$\cosh^{-1} x = \ln\left(x + \sqrt{x^2 - 1}\right) \tag{9.31a}$$

$$\sinh^{-1} x = \ln\left(x + \sqrt{x^2 + 1}\right) \tag{9.31b}$$

where we chose the signs in front of the square roots in such as way as to ensure that $\cosh^{-1} x \geq 0\ \forall x \geq 1$ and $\sinh^{-1} x \in \mathbb{R}\ \forall x \in \mathbb{R}$.

The formulas (9.31a), (9.31b) raise concerns of numerical robustness in cases where the two terms under the logarithm sign are nearly opposite. By reciprocating the values under the logarithm signs we can rewrite them equivalently as

$$\cosh^{-1} x = -\ln\left(x - \sqrt{x^2 - 1}\right) \tag{9.31c}$$

$$\sinh^{-1} x = -\ln\left(\sqrt{x^2 + 1} - x\right) \tag{9.31d}$$

where the choice between (9.31a), (9.31b) and (9.31c), (9.31d) should be made based on comparing the complex arguments of the two terms under the logarithm sign. We should choose the formulas where we are adding two numbers whose complex arguments are not further than 90° apart. Particularly, for real x we may write

$$\sinh^{-1} x = \operatorname{sgn} x \cdot \ln\left(|x| + \sqrt{x^2 + 1}\right) \qquad (x \in \mathbb{R}) \tag{9.31e}$$

whereas the formula (9.31a) already works well for real $x \geq 1$.

Using (9.28) we can construct the principal values:

$$\begin{aligned} \arccos x &= -j \cosh^{-1} x = \\ &= -j \ln\left(x + \sqrt{x^2 - 1}\right) = j \ln\left(x - \sqrt{x^2 - 1}\right) \end{aligned} \tag{9.32a}$$

$$\begin{aligned} \arcsin x &= -j \sinh^{-1} jx = \\ &= -j \ln\left(jx + \sqrt{1 - x^2}\right) = j \ln\left(\sqrt{1 - x^2} - jx\right) \end{aligned} \tag{9.32b}$$

However besides the precision issues there are issues related to the principal values of $\arg x$ switching leaf on the negative real axis. Technically this means that there is a discontinuity in the principal values of $\sqrt{\ }$ and ln on the negative real axis. With (9.31) this was generally tolerable, as the discontinuities weren't

[4] We specifically leave it undefined, whether $\arg x = \pi$ or $-\pi$ for negative real numbers, as this is anyway a line on which the principal value of $\arg x$ has a discontinuity and thus, considering the usual computation precision losses, one often can't rely on the exact value of $\arg x$ being returned for $x < 0$.

arising for the "usual" values of the argument, which are $x \geq 1$ for $\cosh^{-1} x$ and $x \in \mathbb{R}$ for $\sinh^{-1} x$, since neither the argument of the square root nor the argument of the logarithm become real negative in such cases. With (9.32a), on the other hand, we do have a negative expression under the square root for real $x \in (-1, 1)$, which is the most important argument range.

We could therefore adjust (9.32a) to

$$\arccos x = -j \ln\left(x + j\sqrt{1 - x^2}\right) = j \ln\left(x - j\sqrt{1 - x^2}\right) \tag{9.32c}$$

The formulas (9.32b) and (9.32c) work well for $x \in [-1, 1]$, particularly precision-wise it doesn't matter which of the two options in (9.32b) and (9.32c) are taken for $x \in [-1, 1]$, however they exibit a discontinuity for real $x : |x| > 1$. On the other hand, for purely imaginary argument values the formula (9.32b) can be rewritten essentially as (9.31e) to automatically choose the best option precision-wise:

$$\arcsin jx = j \sinh^{-1} x = j \operatorname{sgn} x \cdot \ln\left(|x| + \sqrt{x^2 + 1}\right) \qquad (x \in \mathbb{R}) \tag{9.32d}$$

Preimages of the real line with respect to $\cos x$

By (9.29a) $\cos x$ attains purely real values iff $x \in \mathbb{R}$ or $\operatorname{Re} x = \pi n$ where $n \in \mathbb{Z}$. However, due to periodicity and evenness properties, each value is attained infinitely many times. We would like to choose a principal preimage of the real line with respect to $\cos x$. That is, we are interested in a (preferably continuous) minimal set of points, whose image under transformation $y = \cos x$ is $\mathbb{R}$.

Note, that we do not really have to choose this principal preimage, as the discussions where we are going to refer to it should lead to exactly the same results no matter which of the preimages of the real line is taken. However, for the sake of clarity of discussion it is convenient to have an unambiguous reference preimage.

Apparently there are infinitely many choice possibilities, among which there are at least several "reasonable" ones. For the purposes of this text we will choose the principal preimage as shown in Fig. 9.16. The same figure also shows the periodic repetitions of the principal preimage.[5]

This principal preimage of the real axis thereby consists of three parts:

$$\begin{aligned} x \in [0, \pi] \quad &\Longleftrightarrow y \in [-1, 1] \\ x \in [0, +j\infty) \quad &\Longleftrightarrow y \in [1, +\infty) \\ x \in [\pi, \pi + j\infty) \quad &\Longleftrightarrow y \in (-\infty, -1] \end{aligned}$$

Apparently the principal preimage alone doesn't cover all preimage points of the real line. Neither does it if we add its periodic repetitions in Fig. 9.16, since the lower-semiplane points of lines $\operatorname{Re} x = \pi n$ are still not included. We can include them by simply rotating all preimages in Fig. 9.16 around the origin, which corresponds to multiplication of all points x by -1. Notice that by adding periodic repetitions we addressed the periodicity of $\cos x$, while by adding the preimages multiplied by -1 we addressed the evenness property of $\cos x$.

[5] Notice that the principal preimage in Fig. 9.16 doesn't necessarily coincide with the set of values returned by the formulas (9.32), particularly since the arccos formulas in (9.32) exhibit discontinuities either for $x \in [-1, 1]$ or for real $x : |x| > 1$. The main reason to choose this specific preimage is that its generalization to the case of Jacobian elliptic cosine will be convenient for our purposes.

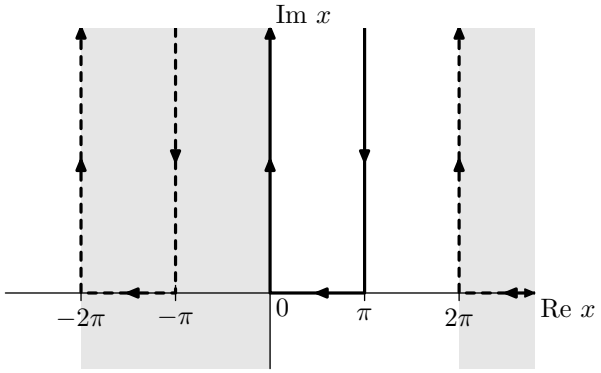


Figure 9.16: The principal preimage (solid line) of the real axis, with respect to $y = \cos x$, and its periodic repetitions (dashed lines).

Representations of horizontal preimage lines by $\cos x$

Equation (9.29a) implies that if the imaginary part v is fixed and the real part u is varying, that is the argument of the cosine is moving in a line parallel to the real axis, then the value of $\cos(u+jv)$ is moving along an ellipse in the complex plane, the real semiaxis of the ellipse being equal to $\cosh v$ and the imaginary semiaxis being equal to $\sinh v$. Fig. 9.17 illustrates. At $v = 0$ the real semiaxis is 1 and the imaginary semiaxis is zero. As $|v|$ grows both semiaxes grow, the imaginary semiaxis staying smaller than the real one in absolute magnitude (Fig. 9.18). Both semiaxes become equal in the limit $v \to \infty$ where the ellipse turns into a circle.

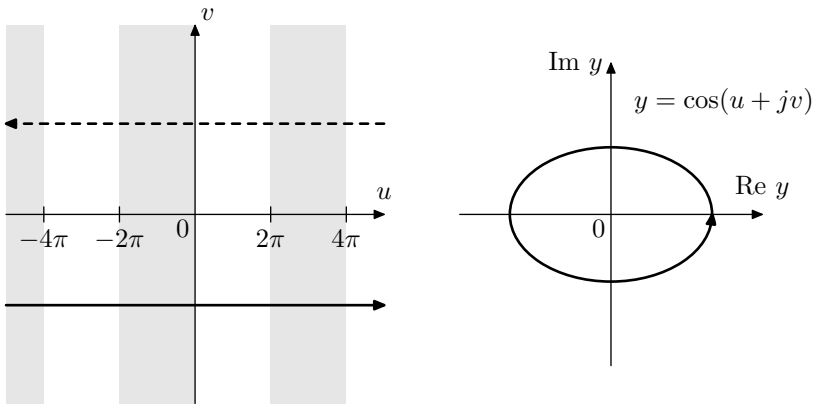


Figure 9.17: An elliptic trajectory and two its preimages. The picture is qualitative. In reality, for this kind of ellipse proportions the preimages would need to be located much closer to the real line.

Given $v > 0$ and increasing u, the movement of the point $\cos(u + jv)$ along the ellipse will be in the negative (clockwise) direction, due to the $-$ sign in front of the imaginary part in (9.29a). Respectively the positive (counterclockwise)

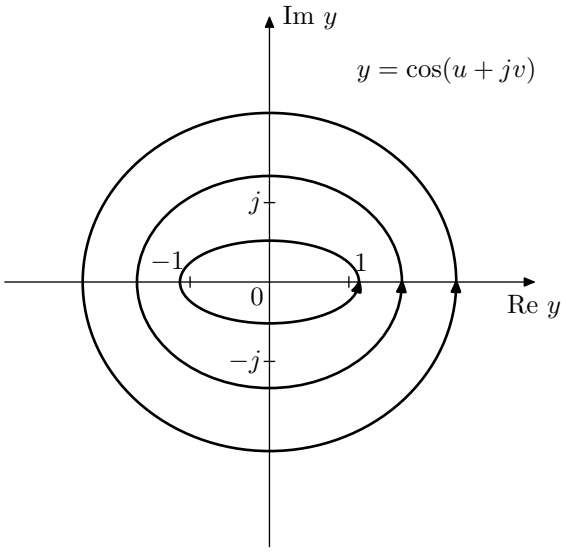


Figure 9.18: A family of elliptic trajectories generated from horizontal preimages $v = \text{const} < 0$.

direction movement will occur either for a decreasing u or for a negative v, where the latter is illustrated in Fig. 9.17.

The even symmetry of the cosine ($\cos(-x) = \cos x$) implies that for each horizontal trajectory $u + jv$ of the cosine's argument, there is a symmetric trajectory $-(u + jv)$ which produces exactly the same cosine trajectory. This other trajectory is shown in Fig. 9.17 by the dashed line. Notice how this is related to the fact that flipping the sign of v flips the direction of the movement along the ellipse: flipping the sign of v is the same as flipping the sign of the entire cosine's argument (that is flipping the signs of both u and v), which leaves the elliptic trajectory unaffected, and then flipping the sign of u, which reverts the direction of movement of both $u + jv$ and $\cos(u + jv)$.

From the fact that the semiaxes of the ellipse are $\cosh v$ and $\sinh v$ and therefore their absolute magnitudes are monotonically growing with $|v|$ (as one can see in Fig. 9.18) we can deduce that ellipses corresponding to different v do not overlap, except for a switch from v to $-v$, which simply changes the direction of the movement along the ellipse. That is for a given ellipse with $\cosh v$ and $\sinh v$ semiaxes there are only two preimages, as shown in Fig. 9.17. An exception occurs when the imaginary semiaxis of the ellipse is zero, in which case there is only one preimage, which is the real line.

Representations of horizontal preimage lines by $\sec x$

The secant function $\sec x = 1/\cos x$ is obviously 2π-periodic. In fact it bears quite a few further similarities to $\cos x$, which are easier to see if we write it in the polar form:

$$|\sec x| = \frac{1}{|\cos x|} \tag{9.33a}$$

$$\arg \sec x = -\arg \cos x \tag{9.33b}$$

Consider a horizontal line $u + jv$ (where u is varying and $v = \text{const}$) and its respective representation $y = \sec(u + jv)$. According to (9.33), y moves in an ellipse-like curve around the origin, as shown in Fig. 9.19. At smaller magnitudes of v the curve begins to look more like a figure of 8 (Fig. 9.20). The curve is not an ellipse anymore due to the reciprocation in (9.33a). On the other hand, by (9.33b) the "angular velocity" is the same as with $y = \cos x$, except that is has the opposite sign, therefore the rotation is happening in the opposite direction. Therefore for an increasing u we get counterclockwise rotation iff $v > 0$ rather than iff $v < 0$.

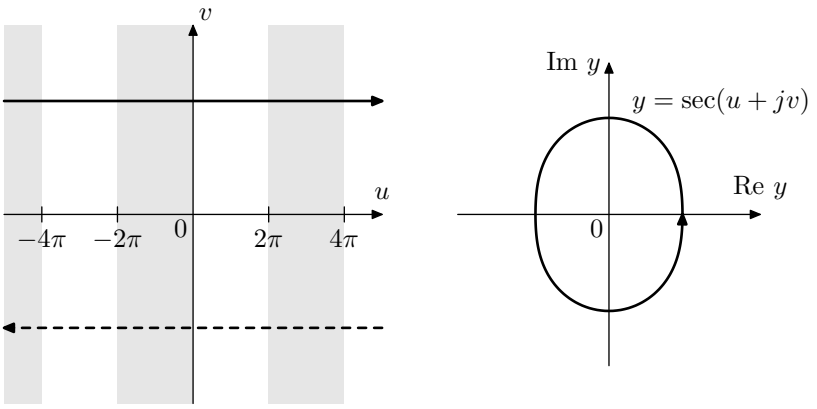


Figure 9.19: A quasielliptic trajectory and two its preimages (qualitatively).

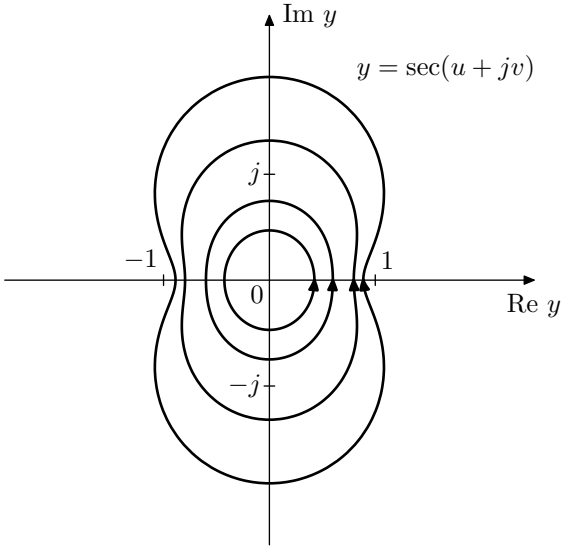


Figure 9.20: A family of quasielliptic trajectories generated from horizontal preimages $v = \text{const} > 0$.

9.6 Chebyshev polynomials

An N-th order Chebyshev polynomial is defined as:

$$T_N(x) = \cos\left(N \arccos x\right) \tag{9.34}$$

Fig. 9.21 illustrates. Notice the bipolar oscillations of equal amplitude (referred to as *equiripples*) on the range $[-1, 1]$. As one can see in Fig. 9.21, the equiripple amplitude is unity.

Somewhat surprisingly, (9.34) can be equivalently written as an N-th order real polynomial of x at any $N \in \mathbb{N}$, e.g. for $N = 4$ we have $T_4(x) = 8x^4 - 8x^2 + 1$, which is why they are called polynomials.

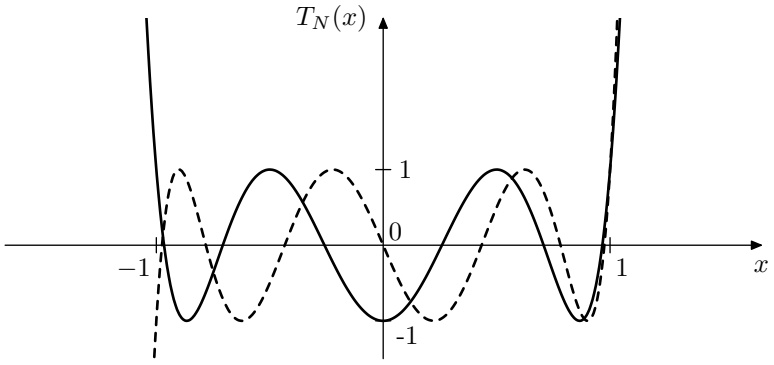


Figure 9.21: Chebyshev polynomials of even (solid) and odd (dashed) orders.

Note that $\arccos x$ takes complex values for $x > 1$ and $x < -1$. We will also often assume x taking complex values, therefore $\arccos x$ and $T_N(x)$ will be complex as well. For $|x| > 1$ even though $\arccos x$ becomes complex, the value $\cos(N \arccos x)$ is still real, and so is the polynomial itself.

A proper discussion of Chebyshev polynomials falls outside the scope of the book, as the respective information can be easily found elsewhere. Here we shall concentrate on the details which will be important for our purposes.

Chebyshev polynomials as representations of linear scaling

Introducing auxiliary variables u and v we rewrite (9.34) as

$$\begin{aligned} x &= \cos u \\ v &= Nu \\ T_N(x) &= \cos v \end{aligned}$$

or, in the implicit form:

$$T_N(\cos u) = \cos(Nu) \tag{9.35}$$

Thus the function $T_N(x)$ is a representation of the linear scaling $v = Nu$, the mapping function being $x = \cos u$. Note that by multiplying the whole preimage domain by j we obtain a different (but equivalent) representation:

$$x = \cosh u$$

$$v = Nu$$
$$T_N(x) = \cosh v$$

which gives us another equivalent expression for (9.34)

$$T_N(x) = \cosh\left(N \cosh^{-1} x\right) \tag{9.36}$$

and its respective implicit form

$$T_N(\cosh u) = \cosh(Nu)$$

In our discussion we will stick to using the cosine-based representation. The readers may draw parallels to the hyperbolic cosine-based representation if they wish.

In case of Butterworth filter-generating functions x^N represented via $\exp u$ mapping, the preimages were $2\pi j$-periodic. This time they are 2π-periodic. Additionally there is an even symmetry: u and $-u$ are preimages of the same x.

Considering the effect the linear scaling $v = Nu$ on the principal preimage of the real line shown in Fig. 9.16, we obtain the following:

- The principal real half-period $[0, \pi]$ is expanded to to $[0, \pi N]$, which is responsible for the occurrence of the equiripples on the segment $x \in [-1, 1]$. Since N is integer, other real half-periods expand similarly, without generating yet more representation values of $f(x)$. Thus $f(x)$ is a single-valued function.

- The principal preimage $[0, +j\infty)$ of $x \in [1, +\infty)$ maps onto itself. The non-principal preimages $[2\pi n + 0j, 2\pi n + j\infty)$ of $x \in [1, +\infty)$ map onto some other preimages $[2\pi Nn + 0j, 2\pi Nn + j\infty)$ of $x \in [1, +\infty)$. Similar mappings occur for the preimages of $x \in [1, +\infty)$ located in the lower complex semiplane. Thus $x \in [1, +\infty)$ is mapped by $T_N(x)$ onto itself, corresponding to the monotonically growing behavior of $T_N(x)$ for $x \geq 1$.

- The principal preimage $[\pi + 0j, \pi + j\infty)$ of $x \in (-\infty, -1]$ is mapped onto $[\pi N + 0j, \pi N + j\infty)$, which is a preimage of $x \in [1, +\infty)$ is N is even and of $x \in (-\infty, -1]$ if N is odd. The non-principal preimages of $x \in (-\infty, -1]$ (both those in the upper complex semiplane and in the lower complex semiplane) are mapped similarly. Thus $x \in (-\infty, -1]$ is mapped by $T_N(x)$ onto itself if N is odd, or onto $x \in [1, +\infty)$ if N is even, corresponding to the monotonic behavior of $T_N(x)$ for $x \leq -1$.

Notice that these results correspond to the graphs in Fig. 9.21.

Now consider a line $\operatorname{Im} u = \beta$ (where β is some real constant value) parallel to the real axis in the preimage domain. Such lines are are, as we know from the previous discussion of the cosine of complex argument, the preimages of ellipses of various sizes, where the size grows with $|\beta|$ (Fig. 9.18). These ellipses are also not overlapping each other, except that the ellipses corresponding to β and $-\beta$ are identical (but have opposite orientations). Therefore $T_N(x)$ maps any ellipse from this family onto another ellipse from this family and vice versa, similarly to how x^N mapped the unit circle onto itself and mapped circles onto

other circles. This time, however, the ellipse which is mapped onto itself, is the one with a zero imaginary semiaxis.

Notice that since the line $\operatorname{Im} u = \beta$ is mapped to $\operatorname{Im} v = N\beta$, the line stays in the same (upper or lower) semiplane after such mapping and goes in the same (to the right or to the left) direction. Thus, if x is moving in an ellipse in a counterclockwise or respectively clockwise direction, then $T_N(x)$ moves in the same direction.

Even/odd property

Since $\cos(u \pm \pi) = -\cos u$, a negation of x corresponds to a shift of its preimage u by π. Respectively v is shifted by $N\pi$, which will result in a negation of $T_N(x)$ if N is odd and will not change $T_N(x)$ is N is even. Therefore $T_N(x)$ is even/odd if N is even/odd:

$$T_N(-x) = (-1)^N T_N(x) \tag{9.37}$$

Values at special points

The principal preimage of $x = 1$ is $u = 0$. Therefore $v = 0$ and $T_N(x) = 1$. Therefore

$$T_N(1) = 1$$

By (9.37)

$$T_N(-1) = (-1)^N$$

The principal preimage of $x = 0$ is $u = \pi/2$. Respectively $v = N\pi/2$ and

$$T_N(0) = \operatorname{Re} j^N = \begin{cases} 0 & \text{if } N \text{ is odd} \\ (-1)^{N/2} & \text{if } N \text{ is even} \end{cases}$$

where $\operatorname{Re} j^N$ is a way of writing the sequence $1, 0, -1, 0, \ldots$ in the same way how $(-1)^N$ is a way of writing the sequence $1, -1, 1, -1, \ldots$.

Leading coefficient

Knowing that $T_N(x)$ is a real polynomial of order N, we can obtain its leading coefficient a_N by evaluating the limit

$$\begin{aligned} a_N &= \lim_{x\to+\infty} \frac{T_N(x)}{x^N} = \lim_{x\to+\infty} \frac{\cosh(N \cosh^{-1} x)}{x^N} = \lim_{x\to+\infty} \frac{\exp(N \cosh^{-1} x)}{2x^N} = \\ &= \lim_{x\to+\infty} \frac{\exp(N \ln 2x)}{2x^N} = \lim_{x\to+\infty} \frac{(2x)^N}{2x^N} = \lim_{x\to+\infty} \frac{2^N x^N}{2x^N} = 2^{N-1} \end{aligned}$$

For the purposes of this book's material, we won't need to be able to explicitly find the other coefficients and will therefore skip this topic.

Zeros

Rather than being interested in the values of the coefficients of Chebyshev polynomials, for our purposes it will be more practical to know the locations of their zeros. Letting $T_N(x) = 0$ we have

$$v = \pi \left(\frac{1}{2} + n \right)$$

Respectively

$$u = \pi \frac{\frac{1}{2} + n}{N}$$

and

$$x = \cos\left(\pi \frac{\frac{1}{2} + n}{N}\right)$$

which means that the zeros are

$$z_n = \cos\left(\pi \frac{\frac{1}{2} + n}{N}\right) \tag{9.38}$$

where there are N distinct values corresponding to $0 < u < \pi$. Notice that the zeros are all real and lie within $(-1, 1)$. Also notice that $z_n = -z_{N-1-n}$, therefore the zeros are positioned symmetrically around the origin. Consequently, if N is odd, one of z_n will be at the origin.

Using (9.38) we can write $T_N(x)$ in the factored form:

$$T_N(x) = x^{N \wedge 1} \cdot \prod_{z_n > 0} \frac{x^2 - z_n^2}{1 - z_n^2} \tag{9.39}$$

where we are taking the product only over the positive zeros using the symmetry of the zeros relatively to the origin, and the odd factor $x^{N \wedge 1}$ (where $N \wedge 1$ denotes bitwise conjunction) appears only for odd N where one of the zeros is at the origin. The normalizations by $(1 - z_n^2)$ are simply appearing from the requirement that each factor must be equal to 1 at $x = 1$, so that $T_N(x) = 1$.

Renormalized Chebyshev polynomials

The factored form (9.39) offers some nice insights into the comparison of the behavior of $T_N(x)$ and x^N. Writing x^N is a comparable factored form we have

$$x^N = x^{N \wedge 1} \cdot \prod_{z_n > 0} x^2 \tag{9.40}$$

where "taking the product over $z_n > 0$" means that we are having as many factors as there are positive zeros in the Chebyshev polynomial $T_N(x)$. Thus the difference between x^N and $T_N(x)$ is that the factors x^2 are replaced by $(x^2 - z_n^2)/(1 - z_n^2)$.

Apparently, if $z_n \to 0 \; \forall n$ then $(x^2 - z_n^2)/(1 - z_n^2) \to x^2$ and respectively (9.39) is approaching x^N. Unfortunately we cannot express this as simply $T_N(x) \to x^N$, since the zeros of $T_N(x)$ are fixed.

To mathematically express this variation of zeros, we can notice that the value of (9.39) at $x = 1$ is always unity. Therefore we can introduce the polynomials

$$\underset{\wedge}{T}_N(x, \lambda) = \frac{T_N(x/\lambda)}{T_N(1/\lambda)} \tag{9.41}$$

to which we refer as *renormalized* Chebyshev polynomials. By construction $\underset{\wedge}{T}_N(1, \lambda) = 1 \; \forall \lambda$, while the zeros of $\underset{\wedge}{T}_N(x, \lambda)$ are $\underset{\wedge}{z}_n = \lambda z_n$. Therefore

$$\underset{\wedge}{T}_N(x, \lambda) = x^{N \wedge 1} \cdot \prod_{z_n > 0} \frac{x^2 - (\lambda z_n)^2}{1 - (\lambda z_n)^2} \tag{9.42}$$

and

$$\lim_{\lambda\to 0} \underset{\wedge}{T}_N(x,\lambda) = x^N$$

The formula (9.41) cannot be evaluated for $\lambda = 0$, however we apparently can take the limit at $\lambda \to 0$ as the value of $T_N(x,0)$ and thus

$$\underset{\wedge}{T}_N(x,0) = x^N$$
$$\underset{\wedge}{T}_N(x,1) = T_N(x)$$

Notice that the formula (9.42) perfectly works at $\lambda = 0$.

Since the equiripple amplitude of $T_N(x)$ is unity, by (9.41) the equiripple amplitude of $\underset{\wedge}{T}_N(x)$ is $1/T_N(1/\lambda)$. The equiripple range $x \in [-1,1]$ of $T_N(x)$ is respectively transformed into the equiripple range $x \in [-\lambda,-\lambda]$ of $\underset{\wedge}{T}_N(x,\lambda)$. Thus λ simultaneously controls the equiripple amplitude and the equiripple range of $\underset{\wedge}{T}_N(x,\lambda)$ (Fig. 9.22).

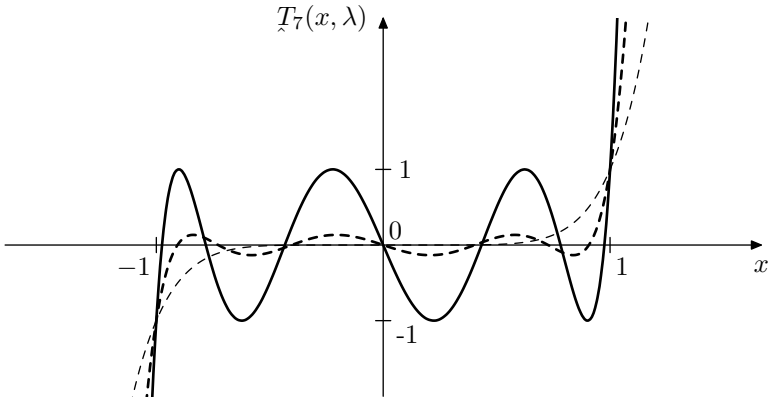


Figure 9.22: Renormalized Chebyshev polynomial $\underset{\wedge}{T}_7(x,\lambda)$ for $\lambda = 1$ (solid), $\lambda = 0.93$ (dashed) and $\lambda = 0$ (thin dashed).

As we won't need $\lambda < 0$, we won't consider that option. As for the large values of λ, it will be practical to restrict the value of λ so that $|\lambda z_n| < 1\ \forall n$. Apparently this means $\lambda_{\max} = 1/\max\{z_n\}$ where $0 \le \lambda < \lambda_{\max}$. Notice that since $|z_n| < 1\ \forall n$, it follows that $\lambda_{\max} > 1$.

Often it will be even more practical to restrict λ to $0 \le \lambda \le 1$. At $\lambda = 1$ the equiripple amplitude of $\underset{\wedge}{T}_N(x,\lambda)$ is already unity. As λ grows further the equiripple amplitude quickly grows, reaching ∞ at $\lambda = \lambda_{\max}$. Also the equiripple range exceeds $[-1,1]$, which could become inconvenient for our purposes.

In order to simplify the notation, often we will omit the λ parameter, understanding it implicitly, and simply write $\underset{\wedge}{T}_N(x)$ instead of $\underset{\wedge}{T}_N(x,\lambda)$.

Slope at $|x| \ge 1$

Let's compare the factors of (9.42) and (9.40). Computing the differences:

$$\frac{x^2-(\lambda z_n)^2}{1-(\lambda z_n)^2} - x^2 = \frac{x^2-(\lambda z_n)^2 - x^2 + (\lambda z_n)^2 x^2}{1-(\lambda z_n)^2} = \frac{(\lambda z_n)^2(x^2-1)}{1-(\lambda z_n)^2} \qquad (9.43)$$

and taking into account that $0 < z_n < 1$, we notice that for $|x| > 1$ and $0 < \lambda \le \lambda_{\max}$ the differences (9.43) are strictly positive and respectively the

factors of (9.42) are larger than those of (9.40). In the range $x > 1$, since all factors are positive, we have

$$\underset{\wedge}{T}_N(x) > x^N \qquad (x > 1,\ N > 1)$$

By the even/odd symmetries of $\underset{\wedge}{T}_N(x)$ and x^N:

$$|\underset{\wedge}{T}_N(x)| > |x^N| \qquad (|x| > 1,\ N > 1)$$

From (9.43) we can also notice that the difference grows with λ, thus at larger λ the polynomial $\underset{\wedge}{T}_N(x)$ exceeds x^N (in absolute magnitude) by a larger amount.

At $\lambda = 1$ we have $\underset{\wedge}{T}_N(x) = T_N(x)$. For this specific case we would like to get a more exact estimation of the steepness of the slope at $x = 1$, to get an idea of how much steeper is the slope of $T_N(x)$ compared to x^N. We have already seen that the leading coefficient of $T_N(x)$ is 2^{N-1}, which means that at $x \to \infty$ the polynomial $T_N(x)$ grows 2^{N-1} times faster than x^N. It would be also informative to compare their slope at $x = 1$.

An attempt to compute the derivative of $T_N(x)$ at $x = 1$ in a straightforward manner results in an uncertainty, thus it's easier to take a way around. At points infinitely close to $x = 1$ we expand the cosine into Taylor series up to the second order term:

$$x = \cos u = 1 - \frac{u^2}{2}$$
$$T_N(x) = \cos v = 1 - \frac{v^2}{2}$$

This scaling by N times in the preimage domain ($v = Nu$) corresponds to scaling by N^2 times in the representation domain ($v^2 = N^2u^2$) and we obtain

$$\frac{\mathrm{d}}{\mathrm{d}x}T_N(x)\Big|_{x=1} = N^2$$

On the other hand

$$\frac{\mathrm{d}}{\mathrm{d}x}x^N\Big|_{x=1} = N$$

Thus at $x = 1$ Chebyshev polynomials grow N times faster than x^N.

9.7 Chebyshev type I filters

Chebyshev (or, more precisely, Chebyshev type I) filters arise by using renormalized Chebyshev polynomials $\underset{\wedge}{T}_N(\omega)$ as $f(\omega)$ in (9.18).[6] The main motivation to use renormalized Chebyshev polynomials instead of ω^N (which is used in Butterworth filters) is that, as we already know they grow faster than ω^N for $|\omega| > 1$, which results in a steeper transition band compared to Butterworth filters. The tradeoff is that in order to achieve a steeper transition band we need

[6] Classically, Chebyshev filters are obtained from Chebyshev polynomials $T_N(\omega)$ by letting $f(\omega) = \varepsilon T_N(\omega)$ where $\varepsilon > 0$ is some small value. This way however usually requires some cutoff correction afterwards. The way how we introduce Chebyshev filters is essentially the same, but directly results in a better cutoff positioning. One way is related to the other via (9.44) combined with a cutoff adjustment by the factor λ.

to allow ripples in the passband. At the same time, the analytical expressions (9.34) and (9.36) allow to easily obtain the function inversion of the polynomial, allowing analytical computation of the filter's internal variables (such as pole positions) for arbitrarily high polynomial orders N, which would have been impossible for polynomials of a fully general form.

Thus, in (9.18) we let

$$f(\omega) = \underset{\wedge}{T}_N(\omega)$$

that is

$$|H(j\omega)|^2 = \frac{1}{1 + \underset{\wedge}{T}_N^2(\omega)}$$

The λ parameter of $\underset{\wedge}{T}_N(\omega)$ is affecting the equiripple amplitude of $\underset{\wedge}{T}_N$ and thereby the equiripple amplitude in the passband of $|H(j\omega)|$. It is convenient to introduce the additional variable

$$\varepsilon = \frac{1}{T_N(1/\lambda)} \tag{9.44}$$

which is simply equal to the equiripple amplitude of $\underset{\wedge}{T}_N$. Using (9.44) we particularly may write

$$f(\omega) = \underset{\wedge}{T}_N(\omega) = \varepsilon T_N(\omega/\lambda)$$

Notice that (9.44) allows to compute ε from λ and vice versa. Therefore, if we are given a desired equiripple band $[-\lambda, \lambda]$, we thereby have specified λ and can use (9.44) to compute ε. Conversely, if we are given a desired equiripple amplitude (which is a more common case), we thereby have specified ε and can invert (9.44) to compute λ:

$$\frac{1}{\lambda} = T_N^{-1}(1/\varepsilon) = \cosh\left(\frac{1}{N}\cosh^{-1}\frac{1}{\varepsilon}\right)$$

(where T_N^{-1} denotes the inverted function T_N).

The amplitude response $|H(j\omega)|$ is thus varying within $[1/\sqrt{1+\varepsilon^2}, 1]$ on the equiripple range $\omega \in [-\lambda, \lambda]$. On the other hand, λ (or, equivalently, ε) affects the slope of $\underset{\wedge}{T}_N$ (and respectively the slope of $|H(j\omega)|$) at $\omega = 1$. The slope steepness is thereby traded against the equiripple amplitude, where steeper slopes are achieved at larger equiripple amplitudes. Fig. 9.23 illustrates.

Poles of Chebyshev type I filters

We have mentioned that the (9.34) is actually a polynomial of x. Therefore the denominator of (9.18) is a polynomial of ω and we can find the roots of this polynomial, which are simultaneously the poles of $|H(s)|^2 = H(s)H(-s)$. The equation for these poles is thus

$$1 + \underset{\wedge}{T}_N^2(\omega) = 0$$

or

$$\underset{\wedge}{T}_N(\omega) = \pm j$$

or

$$\varepsilon T_N(\omega/\lambda) = \pm j$$

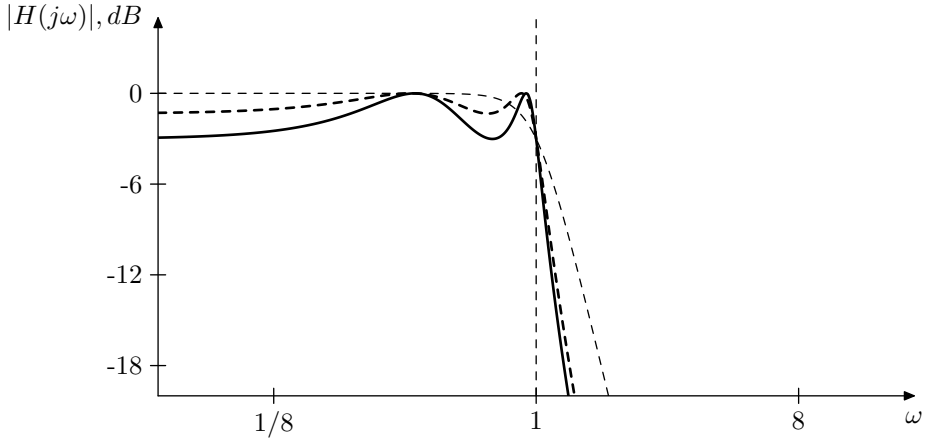


Figure 9.23: Chebyshev type I filter's amplitude responses for $N = 4$ and $\varepsilon = 1$ (solid), $\varepsilon = 0.6$ (dashed) and $\varepsilon = 0$ (Butterworth, thin dashed).

or, introducing $\bar{\omega} = \omega/\lambda$

$$T_N(\bar{\omega}) = \pm\frac{j}{\varepsilon} \tag{9.45}$$

It is quite helpful to use the interpretation of T_N in terms of representation preimage domain to solve (9.45). Recall that T_N maps ellipses (of a special ellipse family, where the real and imaginary semiaxes a and b are related as $a^2 - b^2 = 1$, so that they can be represented as $a = \cosh\beta$, $b = \sinh\beta$ for some β) to ellipses (of the same family). In the preimage domain these ellipses correspond to lines $\operatorname{Im} u = \beta$ parallel to the real axis.

Suppose $\bar{\omega}$ is moving in such an ellipse. This corresponds to its preimages moving along two lines $\operatorname{Im} u = \pm\beta$. Let u be one of the preimages in $\operatorname{Im} u = \beta$, to which we will refer as the principal preimage. Respectively the full family of preimages is $\pm u + 2\pi n$. The principal preimage v of $T_N(\bar{\omega})$ is therefore $v = Nu$, moving along the line $\operatorname{Im} v = N\beta$. The full family of preimages of $T_N(\bar{\omega})$ is respectively $\pm Nu + 2\pi Nn$ and is moving along the lines $\operatorname{Im} v = \pm N\beta$. Therefore $T_N(\bar{\omega})$ is moving in an ellipse whose real and imaginary semiaxes are $\cosh N\beta$ and $\sinh N\beta$ respectively.

We wish $T_N(\bar{\omega})$ to move *in a counterclockwise direction* along an ellipse which goes through the $\pm j/\varepsilon$ points. Then at the moments when $T_N(\bar{\omega}) = \pm j/\varepsilon$ we will obtain solutions of (9.45). We additionally wish the real part of the preimage v of $T_N(\bar{\omega})$ to be increasing during such movement,[7] therefore for a counterclockwise movement of $T_N(\bar{\omega})$ we need $\operatorname{Im} v = N\beta < 0$. Therefore, in order for the ellipse to go through $\pm j/\varepsilon$, the imaginary semiaxis $\sinh N\beta$ of this ellipse must be equal to $-1/\varepsilon$ and thus we obtain:

$$\beta = -\frac{1}{N}\sinh^{-1}\frac{1}{\varepsilon}$$

[7]This choice is arbitrary, we simply better like the option of increasing real part of v. Alternatively we could let the real part of v decrease, obtaining $\beta > 0$. However then we would need to have a negative coefficient in front of n in (9.46).

According to (9.29a), the purely imaginary values of a cosine are attained when the real part of the cosine's argument is equal to $\pi(\frac{1}{2}+n)$, where $n \in \mathbb{Z}$. Thus, the values $\pm j/\varepsilon$ will be attained by $T_N(\bar{\omega})$ at

$$v = jN\beta + \pi\left(\frac{1}{2}+n\right) \tag{9.46}$$

where, since $\beta < 0$, the value $T_N(\bar{\omega}) = j/\varepsilon$ is attained at $n = 0$ and other even values of n. Thus, the solutions of the even pole equation $f = j$ will occur at even values of n. Fig. 9.24 illustrates.

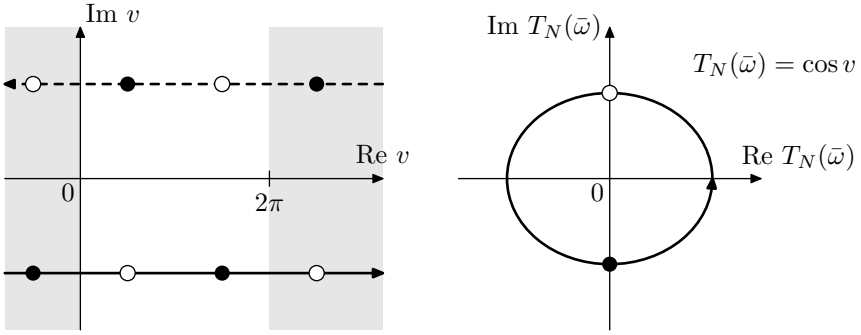


Figure 9.24: Preimages of $T_N(\bar{\omega}) = \pm j/\varepsilon$ (qualitatively, the scales of the real and imaginary axes in the v plane are not equal).

From (9.46) we obtain

$$u = j\beta + \pi\frac{\frac{1}{2}+n}{N}$$

where there are $2N$ essentially different preimages of $\bar{\omega}$ occuring at $2N$ consecutive values of n all lying on the line $\operatorname{Im} u = \beta$. Going back to the representation domain we obtain $\bar{\omega}$ lying on the respective ellipse:

$$\bar{\omega} = \cos\left(j\beta + \pi\frac{\frac{1}{2}+n}{N}\right) \tag{9.47}$$

Fig. 9.25 illustrates.

Switching to $\omega = \lambda\bar{\omega}$ we have:

$$\omega = \lambda\cos\left(j\beta + \pi\frac{\frac{1}{2}+n}{N}\right) \tag{9.48}$$

Note that formally allowing n to take real values and letting $n = -1/2$ we obtain $u = j\beta$ and $\omega = \lambda\cos(j\beta) = \lambda\cosh\beta$ which is a real positive value. Since the imaginary part of the cosine's argument is negative, the values of ω are moving counterclockwise for increasing n, starting from the value on the positive real semiaxis occuring at $n = -1/2$. That is, the values of ω are moving counterclockwise starting from the positive real semiaxis. As we already found out, the values occurring at even/odd n correspond to even/odd poles respectively, and thus the even and odd poles are interleaved on the ellipse.

Switching from ω to $s = j\omega$ we obtain the expression for the poles:

$$s = j\lambda\cos\left(j\beta + \pi\frac{\frac{1}{2}+n}{N}\right) =$$

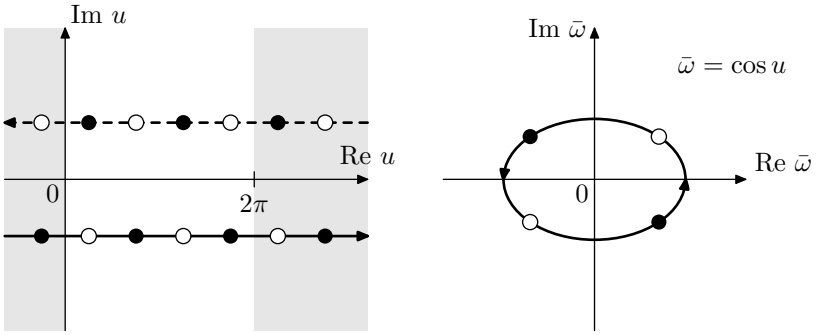


Figure 9.25: Transformation of Fig. 9.24 by $u = v/N$ (for $N = 2$). The white and black dots on the ellipse are even/odd Chebyshev poles in terms of $\bar\omega$. (The picture is qualitative, as the scales of the real and imaginary axes in the u plane are not equal.)

$$= \lambda \sinh\beta \sin\pi\frac{\frac{1}{2}+n}{N} + j\lambda\cosh\beta\cos\pi\frac{\frac{1}{2}+n}{N} \tag{9.49}$$

Since the values of ω are moving counterclockwise starting from the real positive semiaxis, the values of s are moving counterclockwise starting from the imaginary "positive" semiaxis, which means that starting at $n = 0$ we first obtain the stable poles at $n = 0, \ldots, N-1$. The next N values of n will give the unstable poles.

Note that since $(\frac{1}{2}+n)/N$ never takes integer values, the real part of s is never zero and there are no poles on the imaginary axis. The poles are also symmetric relatively to the real and imaginary axes and we can discard the half of the poles located in the right complex semiplane in the same way how we did it with the Butterworth filter. Figs. 9.26 and 9.27 illustrate.[8]

In Figs. 9.26 and 9.27 one could notice that the poles are condensed close to the imaginary axis while the Butterworth poles were evenly spacing. This is easily explained in terms of (9.29a), which gives

$$\tan\arg\cos(u+jv) = -\frac{\sinh v\sin u}{\cosh v\cos u} = -\tan u\cdot\tanh v$$

Now, if $\tanh v$ had been equal to 1, we would have had $\arg\cos(u+jv) = -u$, which would have resulted in an even angular distribution of poles. However, since $|\tanh v| < 1$, the poles are located closer to the real axis in the ω plane or closer to the imaginary axis in the s plane.

Gain adjustments

Having obtained the poles and keeping in mind that there are no zeros, we can obtain the transfer function in the form

$$H(s) = g\cdot\prod\frac{1}{s-p_n}$$

[8]It might seem that the imaginary semiaxes of the ellipses in Figs. 9.26 and 9.27 are of unit length. This is not exactly so, although they are very close, being equal to approximately 1.00003 and 1.00004 respectively.

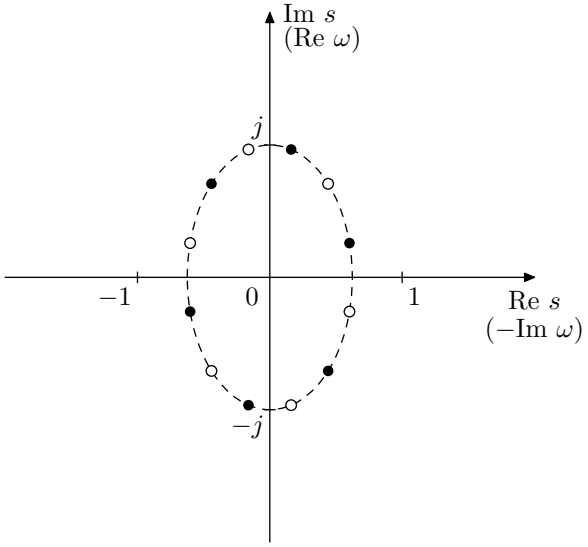


Figure 9.26: Chebyshev type I filter's even (white) and odd (black) poles for $N = 6$ (including the poles of $H(-s)$).

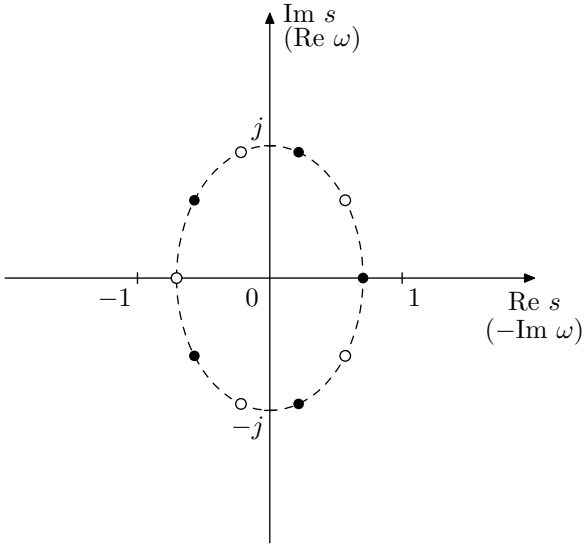


Figure 9.27: Chebyshev type I filter's even (white) and odd (black) poles for $N = 5$ (including the poles of $H(-s)$).

where the gain g could be obtained by evaluating the above product at $\omega = 0$ and comparing to $H(0)$. It could be a bit more practical though, to write $H(s)$ as a product of 1-pole lowpasses with unity passband gains:

$$H(s) = g \cdot \prod \frac{1}{s/(-p_n) + 1} \tag{9.50}$$

where $-p_n$ are the (possibly complex) cutoffs and where the coefficient g is different than in the previous formula. For (9.50) we are having $H(0) = g$ and thus, using (9.18), we can obtain g from

$$H(0) = \frac{1}{\sqrt{1+\hat{T}_N^2(0)}} = \frac{1}{\sqrt{1+\varepsilon^2 T_N^2(0)}} = \frac{1}{1+\varepsilon^2\left(\operatorname{Re} j^N\right)^2} = \\ = \begin{cases} \dfrac{1}{1+\varepsilon^2} & \text{for } N \text{ even} \\ 1 & \text{for } N \text{ odd} \end{cases} \tag{9.51}$$

Another option is to obtain the leading gain g from the requirement $|H(j)| = 1/\sqrt{2}$ arising from

$$|H(j)|^2 = \frac{1}{1+\hat{T}_N^2(1)} = \frac{1}{2}$$

However this might accidentally result in a 180° phase response at $\omega = 0$, (since we used $|H(j)|$ rather than $H(j)$ as a reference) therefore one needs to be careful in this regard.

Using the default normalization of the Chebyshev filter's gain given by (9.51), we obtain the amplitude response varying within $[1/\sqrt{1+\varepsilon^2}, 1]$ on the range $\omega \in [-\lambda, \lambda]$. We could choose some other normalizations, though. E.g. we could require $|H(0)| = 1$, which will be automatically achieved if we simply let $g = 1$ in (9.50). Or we could require the ripples to be symmetric relatively to the zero decibel level, which is achieved by multiplying (9.51) by $(1+\varepsilon^2)^{1/4}$:

$$H(0) = \sqrt{\frac{\sqrt{1+\varepsilon^2}}{1+\varepsilon^2 T_N^2(0)}}$$

so that $|H(j\omega)|$ varies within $[1/(1+\varepsilon^2)^{1/4}, (1+\varepsilon^2)^{1/4}]$ within the equiripple band.

Butterworth limit

Since at $\lambda \to 0$ we have $\hat{T}_N(x) \to x^N$, in the limit $\lambda \to 0$ Chebyshev type I filter turns into a Butterworth filter of the same order N.

Simultaneously the ellipse semiaxes $\lambda \sinh\beta$ and $\lambda \cosh\beta$ in (9.49) are both approaching the unity length. Indeed, letting $\varepsilon \to 0$ (which is equivalent to $\lambda \to 0$), we have

$$\frac{1}{\lambda} = \cosh\left(\frac{1}{N}\cosh^{-1}\frac{1}{\varepsilon}\right) \sim \exp\left(\frac{1}{N}\cosh^{-1}\frac{1}{\varepsilon}\right) = \\ = \exp\frac{\ln\left(\varepsilon^{-1}+\sqrt{\varepsilon^{-2}-1}\right)}{N} = \left(\varepsilon^{-1}+\sqrt{\varepsilon^{-2}-1}\right)^{1/N} \sim \left(2\varepsilon^{-1}\right)^{1/N}$$

$$\cosh\beta = \cosh\left(-\frac{1}{N}\sinh^{-1}\frac{1}{\varepsilon}\right) \sim \exp\left(\frac{1}{N}\sinh^{-1}\frac{1}{\varepsilon}\right) = \\ = \exp\frac{\ln\left(\varepsilon^{-1}+\sqrt{\varepsilon^{-2}+1}\right)}{N} = \left(\varepsilon^{-1}+\sqrt{\varepsilon^{-2}+1}\right)^{1/N} \sim \left(2\varepsilon^{-1}\right)^{1/N}$$

Thus $1/\lambda$ and $\cosh\beta$ are asymptotically identical and therefore $\lambda\cosh\beta \to 1$. In a similar way we can show that $\lambda\sinh\beta \to 1$.

9.8 Chebyshev type II filters

Chebyshev polynomials $T_N(x)$ have equiripple behavior on $[-1,1]$ and grow to infinity outside of that range. By reciprocating the argument: $T_N(1/x)$, we obtain equiripple behavior on $|x| \geq 1$ and infinite growth for $x \to 0$. By further reciprocating the value of the polynomial: $1/T_N(1/x)$, we have again small values on the range $[-1,1]$, while for $|x| \geq 1$ the polynomial's value exhibits equiripple oscillations around infinity. We therefore introduce the function

$$\mathcal{I}_N(x) = \frac{1}{T_N(1/x)}$$

where Fig. 9.28 illustrates the behavior of $\mathcal{I}_N$. We will refer to $\mathcal{I}_N(x)$ as a *double-reciprocated* (once in argument and once in value) Chebyshev polynomial.

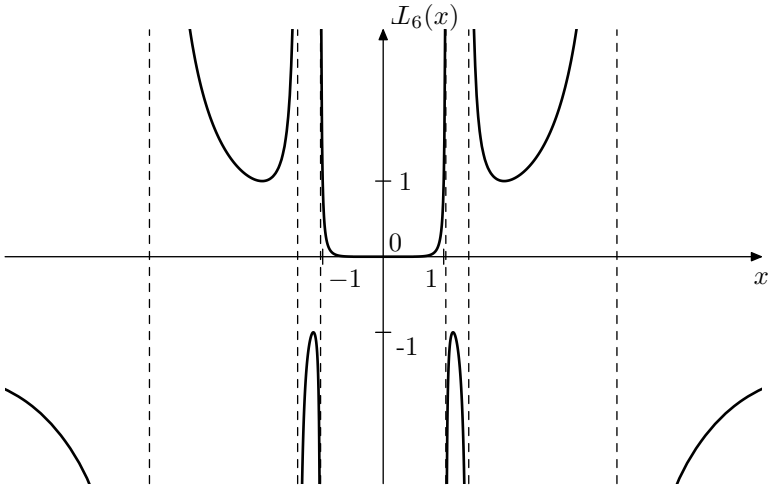


Figure 9.28: Double-reciprocated Chebyshev polynomial $\mathcal{I}_N$.

The equiripple oscillations around infinity are also better visible in the arctangent scale (Fig. 9.29). More specifically, on $[1,+\infty)$ and $(-\infty,-1]$ the value oscillates between ± 1 and ∞, never becoming less than 1 in the absolute magnitude. We could refer to that fact by saying that the amplitude of these oscillations around ∞ is unity, thereby taking the minimum absolute magnitude of the oscillating value as the oscillation amplitude, even though that might be considered some kind of a misnomer. We will be using this definition of amplitude of oscillations around infinity further in the text.

We also introduce the renormalized version of the double-reciprocated Chebyshev polynomials by double-reciprocating $\underset{\wedge}{T}_N$:

$$\underset{\wedge}{\mathcal{I}}_N(x,\lambda) = \frac{1}{\underset{\wedge}{T}_N(1/x,\lambda)} = \frac{T_N(1/\lambda)}{T_N(1/\lambda x)} = \frac{\mathcal{I}_N(\lambda x)}{\mathcal{I}_N(\lambda)} \tag{9.52}$$

where we have $\underset{\wedge}{\mathcal{I}}_N(1,\lambda) = 1$. Notice that we didn't reciprocate the λ parameter. The idea is that $0 \leq \lambda \leq 1$ and that

$$\underset{\wedge}{\mathcal{I}}_N(x,0) = \frac{1}{\underset{\wedge}{T}_N(1/x,0)} = \frac{1}{1/x^N} = x^N$$

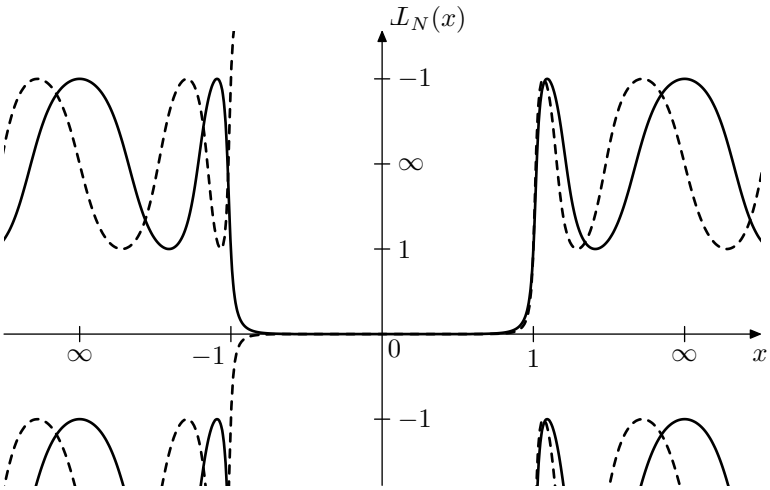


Figure 9.29: Double-reciprocated Chebyshev polynomials of even (solid) and odd (dashed) orders, using arctangent scale in both axes.

$$\underset{\wedge}{\mathcal{L}}_N(x,1) = \frac{1}{\underset{\wedge}{T}_N(1/x,1)} = \frac{1}{1/T_N(1/x)} = \mathcal{L}_N(x)$$

Fig. 9.30 illustrates. As usual, we will often omit the λ parameter, understanding it implicitly.

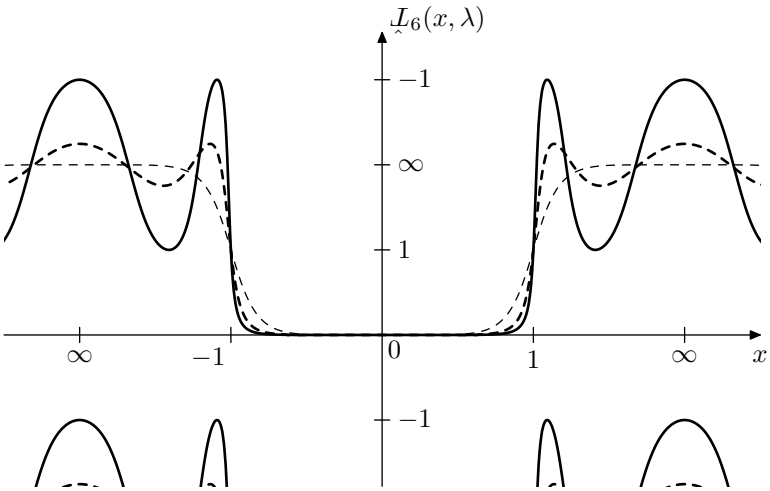


Figure 9.30: Renormalized double-reciprocated Chebyshev polynomial $\underset{\wedge}{\mathcal{L}}_6(x,\lambda)$ for $\lambda = 1$ (solid), $\lambda = 0.93$ (dashed) and $\lambda = 0$ (thin dashed).

By (9.52) the amplitude of the equiripples of $\underset{\wedge}{\mathcal{L}}_N(x)$ is $T_N(1/\lambda)$. By using the same equation (9.44) as we have been using for Chebyshev type II filters we have $T_N(1/\lambda) = 1/\varepsilon$, that is the equiripple amplitude is $1/\varepsilon$. This is actually a convenient notation, since in this case we are having smaller (closer to ∞)

equiripples at smaller ε. We also thereby have:

$$\mathit{\hat{L}}_N(x) = \frac{1}{\varepsilon T_N(1/\lambda x)} = \frac{\mathit{L}_N(\lambda x)}{\varepsilon}$$

By writing the Chebyshev polynomial as a polynomial:

$$T_N(x) = \sum_{n=0}^{N} a_n x^n$$

we find that the double-reciprocated Chebyshev polynomial is a rational function of x:

$$\mathit{L}_N(x) = \frac{1}{\sum_{n=0}^{N} a_n x^{-n}} = \frac{x^N}{\sum_{n=0}^{N} a_n x^{N-n}}$$

The same apparently is true for $\mathit{\hat{L}}_N(x)$ and therefore we could try using $\mathit{\hat{L}}_N(\omega)$ as $f(\omega)$ in (9.18).

Letting $f(\omega) = \mathit{\hat{L}}_N(\omega)$ in (9.18) we obtain a Chebyshev type II filter, this time trading the ripples in the stopband against the transition band's rolloff (Fig. 9.31). The stopband peaks are achieved at $\mathit{\hat{L}}_N(\omega) = 1/\varepsilon$, thus the ripple amplitude is $1/\sqrt{1+\varepsilon^{-2}}$.

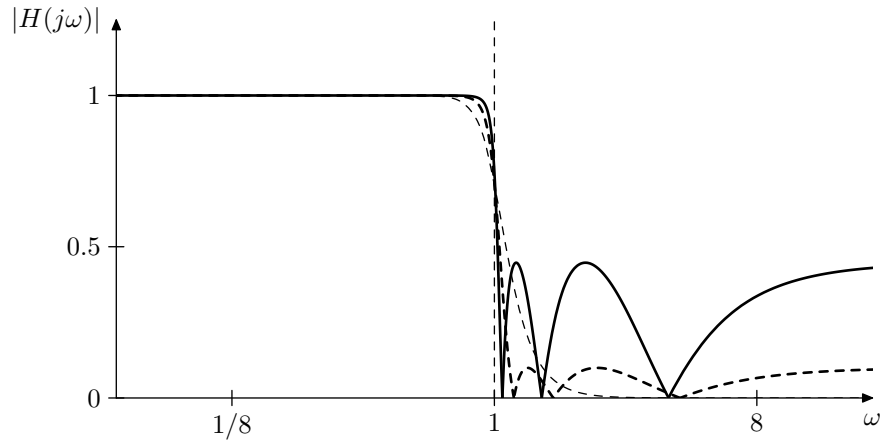


Figure 9.31: Chebyshev type II filter's amplitude responses for $N = 6$ and $\varepsilon = 0.5$ (solid), $\varepsilon = 0.1$ (dashed) and $\varepsilon = 0$ (Butterworth, thin dashed). Notice the usage of the linear amplitude scale, which is chosen in order to be able to show the amplitude response zeros.

$\mathit{L}_N(x)$ as representations of linear scaling

In order to find the poles of a Chebyshev type II filter we are going, as usual, to interpret the function $\mathit{L}_N(x)$ as a representation of the linear scaling $v = Nu$. This time we need the following mapping:

$$x = \frac{1}{\cos u} = \sec u$$

$$v = Nu$$
$$\cancel{L}_N(x) = \frac{1}{\cos v} = \sec v$$

Same as with $T_N(x)$, the multiplication by N expands the principal real period $[0, 2\pi]$ to $[0, 2\pi N]$ and there are similar transformations of the preimages of the real axis. The transformations of quasielliptic curves (shown in Fig. 9.20) which are representations of horizontal lines $\operatorname{Im} u = \text{const}$ in the preimage domain are also occurring in a similar fashion, each such curve being transformed into another such curve.

Poles of Chebyshev type II filters

The pole equation is

$$1 + \cancel{L}_N^2(\omega) = 0$$

The even/odd pole equations are respectively

$$\cancel{L}_N(\omega) = \pm j$$

or

$$\frac{\cancel{L}_N(\lambda\omega)}{\varepsilon} = \pm j$$

or, introducing $\bar{\omega} = \lambda\omega$

$$\cancel{L}_N(\bar{\omega}) = \pm j\varepsilon$$

where the "+" sign corresponds to the even poles and the "−" sign to odd poles.

Suppose $\bar{\omega}$ is moving in a counterclockwise direction in a quasielliptic curve which is a representation of $\operatorname{Im} u = \beta$ (where, according to our previous discussion of the properties of the $x = 1/\cos u$ mapping, $\beta > 0$). This results in a similar counterclockwise motion of $\cancel{L}_N(\bar{\omega})$. We wish $\cancel{L}_N(\bar{\omega})$ to pass through the points $\pm j\varepsilon$ going counterclockwise.

At this point we could follow similar steps as we did for Chebyshev type I filters, using the preimage linear scaling interpretation of $\cancel{L}_N$ to obtain the points $\bar{\omega}$ where $\cancel{L}_N(\bar{\omega}) = \pm j\varepsilon$. However we also could notice that $\cancel{L}_N(\bar{\omega}) = \pm j\varepsilon \iff T_N(\bar{\omega}^{-1}) = \mp j/\varepsilon$. Therefore we could reuse the results of our discussion of Chebyshev type I filters, where we needed T_N to go clockwise through $\mp j/\varepsilon$. That is, we need the value of T_N to move in the same trajectory going through the same points as in the Chebyshev type I case, just in the opposite direction. This can be achieved by flipping its preimage line $\operatorname{Im} v = N\beta$ from the lower semiplane to the upper semiplane. Therefore the values of T_N passing through $\mp j/\varepsilon$ going clockwise should occur at $\sinh N\beta = 1/\varepsilon$ and thus

$$\beta = \frac{1}{N} \sinh^{-1} \frac{1}{\varepsilon}$$

The other difference to the case of Chebyshev type I filters is that the argument of T_N is $\bar{\omega}^{-1}$ rather than $\bar{\omega}$. Therefore we need to replace $\bar{\omega}$ in (9.47) with $\bar{\omega}^{-1}$ obtaining

$$\bar{\omega}^{-1} = \cos\left(j\beta + \pi \frac{\frac{1}{2} + n}{N}\right)$$

and

$$\omega^{-1} = \lambda \cos\left(j\beta + \pi \frac{\frac{1}{2} + n}{N}\right)$$

Since $s = j\omega = j/\omega^{-1}$, we have $-1/s = -\omega^{-1}/j = j\omega^{-1}$ and thus

$$-s^{-1} = j\lambda \cos\left(j\beta + \pi \frac{\frac{1}{2} + n}{N}\right) =$$
$$= \lambda \sinh\beta \sin\pi \frac{\frac{1}{2} + n}{N} + j\lambda \cosh\beta \cos\pi \frac{\frac{1}{2} + n}{N} \tag{9.53}$$

The poles of Chebyshev type II filters are therefore negated reciprocals of the poles of Chebyshev type I filters.[9] By the interpretation of $\mathit{\Lambda}_N$ as linear scaling in the preimage domain, Chebyshev type II poles should lie on quasielliptic trajectories, such as the ones shown in Fig. 9.19 and Fig. 9.20. Notice that, despite the negated reciprocation, the formula (9.53) still first gives the stable poles, due to the flipped sign of β compared to (9.49).

Zeros of Chebyshev type II filters

According to our discussion in Section 9.4 of using rational $f(\omega)$, Chebyshev type II filters also should have zeros, which in terms of ω coincide with poles of $f(\omega)$. The zero equation is thereby

$$\underset{\wedge}{\mathit{\Lambda}}_N(\omega, \lambda) = \infty$$

or

$$\mathit{\Lambda}_N(\lambda\omega) = \infty$$

or, equivalently,

$$T_N(1/\lambda\omega) = 0 \tag{9.54}$$

The solutions of (9.54) thereby obtained from the zeros z_n of T_N (given by (9.38)) by

$$\frac{1}{\lambda\omega} = z_n = \cos\left(\pi \frac{\frac{1}{2} + n}{N}\right)$$

or

$$\omega^{-1} = \lambda \cos\left(\pi \frac{\frac{1}{2} + n}{N}\right)$$

or, in terms of s

$$-s^{-1} = j\lambda \cos\left(\pi \frac{\frac{1}{2} + n}{N}\right)$$

An additional consideration arises at odd N where one of the values given by (9.38) occurs at the origin, which after the reciprocation gives the infinity. This

[9] Alternatively one could notice that the poles of Chebyshev type II (lowpass) filters are identical to the poles of Chebyshev type I hipass filters, since both can be obtained from the poles of Chebyshev type I lowpass filters by the LP to HP transformation. If Chebyshev type II lowpass poles are obtained this way, the order of their enumeration will be symmetrically flipped relatively to the one of the prototype Chebyshev type I lowpass poles. That is, if Chebyshev type I poles are going counterclockwise starting from the "positive" imaginary axis, then Chebyshhev type II poles obtained by the LP to HP tranfromation will be going clockwise from the "negative" imaginary axis (thereby stable poles will be converted to stable poles, but the even/odd property of the poles will be switched if N is even).

means that there is no corresponding finite zero of $H(s)$ and no corresponding factor in the numerator of $H(s)$. Respectively the order of the numerator of $H(s)$ becomes 1 less than the order of the denominator. This automatically results in $H(\infty) = 0$, that is $H(s)$ has a zero at the infinity, as required by the reciprocation of the values given by (9.38). Fig. 9.32 provides an example.

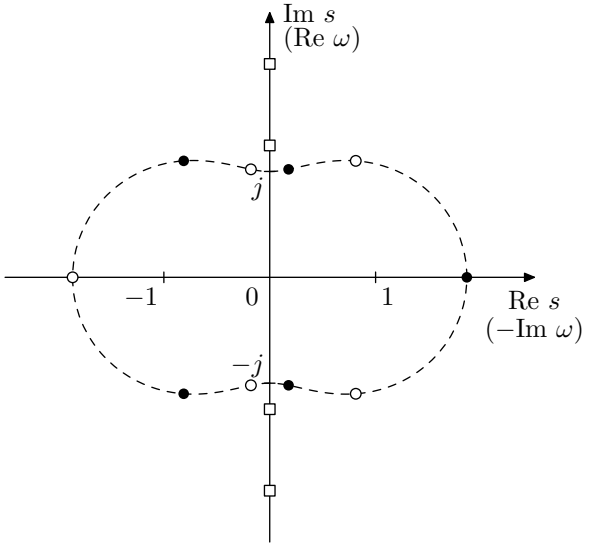


Figure 9.32: Poles (white and black dots) and zeros (white squares) of a Chebyshev type II filter of order $N = 5$. Each of the zeros is duplicated, but the duplicates are dropped together with the unstable poles.

Filter gain

Since there are no ripples in the passband, there is little reason to attempt any gain adjustments and the filter gain needs simply to be chosen from the requirement $H(0) = 1$, thereby defining the leading gain coefficient g of the cascade form (8.1).

Butterworth limit

Since $\hat{L}_N(x, 0) = x^N$ in the limit $\lambda \to 0$ Chebyshev type II filter turns into Butterworth filter.

Notice that the pole formula (9.53) is essentially the negated reciprocal of (9.49). On the other hand, negation and/or reciprocation turn Butteworth poles into themselves (if nonstable poles are included), therefore, since in the limit (9.49) gives Butterworth poles, (9.53) also does the same.

9.9 Jacobian elliptic functions

The next class of equiripple filters which we would like to introduce are elliptic filters. Chebyshev filters were based on the cosine function and required a wider

spectrum of trigonometric and hyperbolic functions for their analysis. Similarly, elliptic filters are based on Jacobian elliptic cosine function and require other Jacobian elliptic functions for their analysis. Since Jacobian elliptic functions are not a part of widely spread common knowledge, on the contrary, the freely available resources are rather scarce, we are going to introduce them and discuss their properties relevant for this book's material.

Additional information can be found in the reference texts listed at the end of this chapter. The results presented here and in the rest of this chapter without any kind of proof or justification are either taken directly or derived from these texts.

Elliptic integrals of the first kind

One of the most common ways to introduce Jacobian elliptic functions is as some kind of special inverses of the elliptic integral of the first kind, which we therefore will briefly discuss first.

The *elliptic integral of the first kind* is the function notated $F(\varphi, k)$ defined by the formula:

$$F(\varphi, k) = \int_0^{\varphi} \frac{\mathrm{d}\theta}{\sqrt{1 - k^2 \sin^2 \theta}} \tag{9.55}$$

The parameter k is referred to as *elliptic modulus*. Normally $0 \leq k \leq 1$. At $k = 0$ we simply have $F(\varphi, 0) = \varphi$.

The value of $F(\varphi, k)$ at $\varphi = \pi/2$ is often of a particular interest, which motivates the introduction of the *complete elliptic integral of the first kind*:

$$K(k) = F(\pi/2, k)$$

Respectively we are having

$$K(0) = F(\pi/2, 0) = \frac{\pi}{2} \tag{9.56a}$$

$$K(1) = F(\pi/2, 1) = \int_0^{\pi/2} \frac{\mathrm{d}\theta}{\sqrt{1 - \sin^2 \theta}} = \int_0^{\pi/2} \frac{\mathrm{d}\theta}{\cos \theta} = \infty \tag{9.56b}$$

The graph of $K(k)$ is shown in Fig. 9.33. Notice that $K(k)$ grows with k (which is obvious from (9.55)).

The elliptic modulus is sometimes expressed as $k = \sin \alpha$ where α is referred to as the *modular angle*. Given a modular angle α, often one also needs the *complementary modular angle* α' which is simply defined as

$$\alpha' = \frac{\pi}{2} - \alpha$$

Respectively there is the complementary elliptic modulus:

$$k' = \sqrt{1 - k^2}$$

and the complementary complete elliptic integral of the first kind:

$$K'(k) = K(k') = F(\pi/2, k')$$

Notice that k' decreases as k increases and vice versa. On the other hand $K(k)$ is a monotonically increasing function. Therefore the ratio $K'(k)/K(k)$ monotonically decreases with growing k. Fig. 9.34 illustrates, where we use the modular angle in the abscissa scale in order to make the symmetry between K and K' explicitly visible.

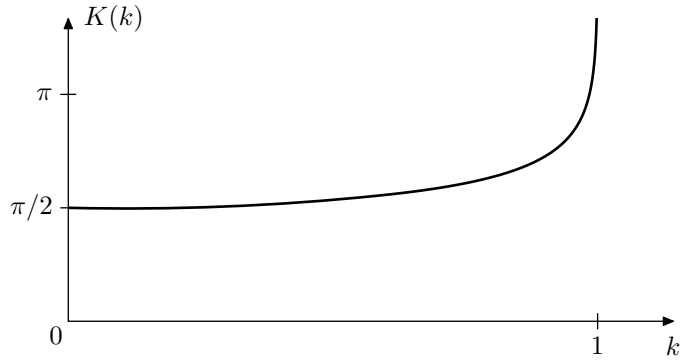


Figure 9.33: Complete elliptic integral of the first kind $K(k)$.

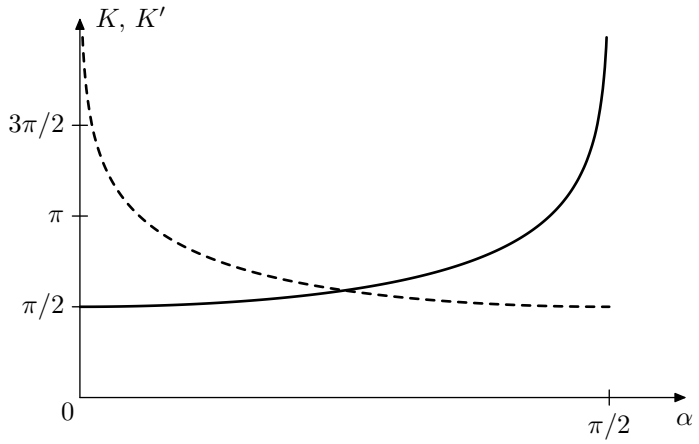


Figure 9.34: Complete elliptic integral of the first kind K (solid line) and the complemetary complete elliptic integral of the first kind K' (dashed line), plotted against the modular angle α.

Jacobian elliptic functions

There are 12 different Jacobian elliptic functions but we will concentrate only on 6 of them. The ones we introduce will bear strong similarities to certain trigonometric and/or hyperbolic functions, becoming equal to them in the limit.

We will define Jacobian elliptic functions in terms of the so called *amplitude*, which is defined as a function $\varphi = \mathrm{am}(x, k)$, which is the inverse of the elliptic integral of the first kind:

$$F(\mathrm{am}(x, k), k) = x \tag{9.57}$$

That is, for a given x the function $\varphi = \mathrm{am}(x, k)$ gives such φ that $F(\varphi, k) = x$. Note that since in the limit $k \to 0$ we have $F(\varphi, 0) = \varphi$, we are respectively having $\mathrm{am}(x, 0) = x$.

As with elliptic integral $F(\varphi, k)$, the second argument k serves a role of the function's parameter, the "primary" argument of the function being x. Often this parameter is simply omitted and understood implicitly: $\varphi = \mathrm{am}\, x$. Even more commonly, this is done for Jacobian elliptic functions (which are having

exactly the same arguments as the amplitude).

Now, from the six Jacobian elliptic functions that we are going to introduce, the four of our primary interest will be:

- Jacobian elliptic "sine" $\mathrm{sn}(x,k)$ is defined by the equation

$$\mathrm{sn}(x,k) = \sin\varphi \tag{9.58}$$

  where $\varphi = \mathrm{am}(x,k)$. Or simply, $\mathrm{sn}(x,k) = \sin\mathrm{am}(x,k)$.

  Fig. 9.35 provides example graphs of $\mathrm{sn}\,x$, where we could also notice that $\mathrm{sn}\,x$ is $4K$-periodic. The value K is simply a short notation for the complete elliptic integral $K(k)$, evaluated for the same modulus k which is used in $\mathrm{sn}(x,k)$. Please also note that the graphs in Fig. 9.35 are plotted "in terms of K", that is different abscissa scales are used for different graphs in the same figure. This has been done in order to provide a better visual comparison of different $\mathrm{sn}(x,k)$ with different periods.

  Since the limit $k \to 0$ we have $\varphi = \mathrm{am}(x,0) = x$, the elliptic sine turns into into $\mathrm{sn}(\varphi,0) = \sin\varphi$.

- Jacobian elliptic "cosine"[10] $\mathrm{cd}(x,k)$ is defined by the equation:

$$\mathrm{cd}(x,k) = \frac{\cos\varphi}{\sqrt{1-k^2\sin^2\varphi}} \tag{9.59}$$

  where $\varphi = \mathrm{am}(x,k)$. Fig. 9.36 illustrates, where one could observe that $\mathrm{cd}\,x$ is $4K$-periodic. Apparently $\mathrm{cd}(x,0) = \cos x$.

- Jacobian elliptic "tangent"/elliptic "hyperbolic sine" $\mathrm{sc}(x,k)$ is defined by the equation:

$$\mathrm{sc}(x,k) = \tan\varphi \tag{9.60}$$

  where $\varphi = \mathrm{am}(x,k)$. Fig. 9.37 illustrates, where one could observe that $\mathrm{sc}\,x$ is $2K$-periodic.

  Apparently $\mathrm{sc}(x,0) = \tan x$. However, the function $\mathrm{sc}\,x$ also bears strong similarities to the hyperbolic sine, becoming equal to it in the limit $k \to 1$. In this book we will be mostly using the similarity of sc to sinh, therefore we will typically refer to sc as elliptic "hyperbolic sine".

- Jacobian elliptic "hyperbolic cosine" $\mathrm{nd}(x,k)$ is defined by the equation:

$$\mathrm{nd}(x,k) = \frac{1}{\sqrt{1-k^2\sin^2\varphi}} \tag{9.61}$$

  where $\varphi = \mathrm{am}(x,k)$. Fig. 9.38 illustrates, where one could observe that $\mathrm{nd}\,x$ is $2K$-periodic. This function becomes equal to cosh in the limit $k \to 1$.

[10] There is yet another Jacobian elliptic function, which is simply equal to $\cos\varphi$ rather than $(\cos\varphi)/\sqrt{1-k^2\sin^2\varphi}$. Depending on the purpose either of these functions may be referred to as elliptic cosine. Each of these two functions inherits different properties of $\cos\varphi$. For the purposes of this book we will need the one defined by (9.59) and this is the one to which we will refer to as elliptic cosine.

We will also introduce two "auxiliary" functions:

- Jacobian elliptic "cosecant" $\operatorname{ns} x = 1/\operatorname{sn} x$ (Fig. 9.39)

- Jacobian elliptic "secant" $\operatorname{dc} x = 1/\operatorname{cd} x$ (Fig. 9.40).

Since these two are simply reciprocals of sn and cd, we won't be discussing them much, however they will be used occasionally.

From the previous discussion we could conclude that $4K$ is the common period of all six introduced elliptic functions (where the elliptic "trigonometric" functions are $4K$-periodic and the elliptic "hyperbolic" functions are $2K$-periodic). For that reason $K = K(k)$ is referred to as the *quarter-period*.

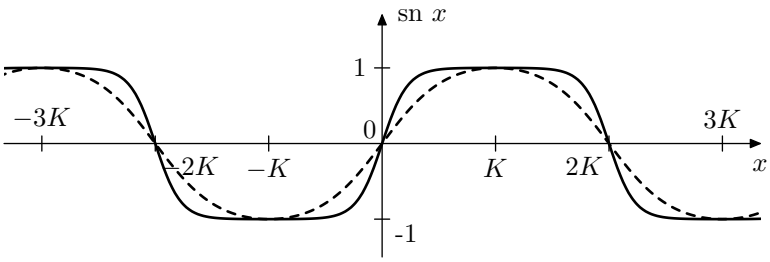


Figure 9.35: Jacobian elliptic sine for $k = 0.8$ (dashed) and $k = 0.999$ (solid).

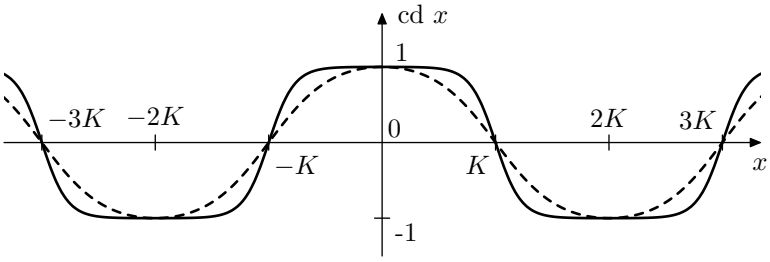


Figure 9.36: Jacobian elliptic cosine for $k = 0.8$ (dashed) and $k = 0.999$ (solid).

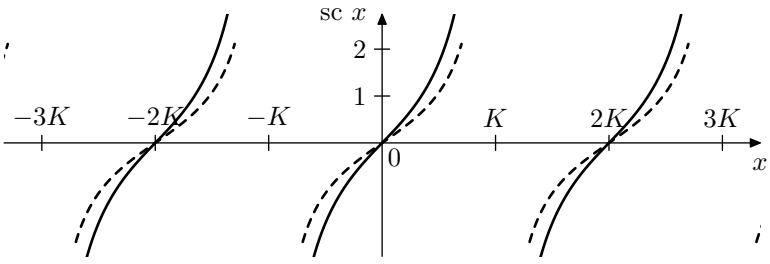


Figure 9.37: Jacobian elliptic "hyperbolic sine" (or "trigonometric tangent") for $k = 0.5$ (dashed) and $k = 0.94$ (solid).

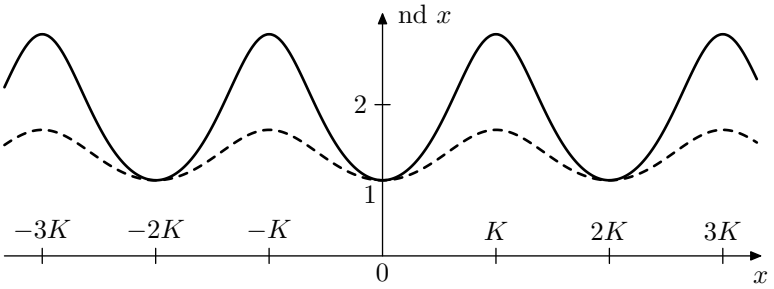


Figure 9.38: Jacobian elliptic "hyperbolic cosine" $k = 0.8$ (dashed) and $k = 0.94$ (solid). The maxima are at $1/k'$.

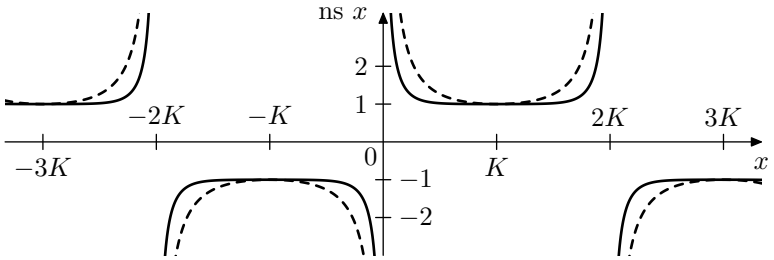


Figure 9.39: Jacobian elliptic cosecant $k = 0.8$ (dashed) and $k = 0.999$ (solid).

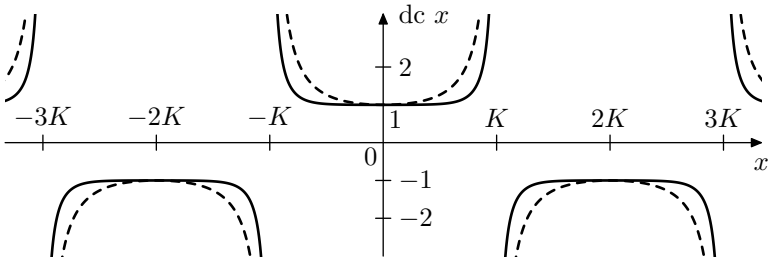


Figure 9.40: Jacobian elliptic secant $k = 0.8$ (dashed) and $k = 0.999$ (solid).

Complex argument

Jacobian elliptic functions can be generalized to complex argument values. Remarkably, all of the six introduced functions can be obtained from each other by shifts and/or rotations of the argument in the complex plane (with some possible scaling of the resulting function's value).

Before discussing any Jacobian elliptic function on the complex plane we need to introduce the imaginary quarter period K' which is simply equal to the complementary complete elliptic integral: $K' = K'(k) = K(k')$. Jacobian elliptic functions are also periodic in the imaginary direction, where the elliptic 'trigonometric" functions are $2jK'$-periodic and the elliptic 'hyperbolic" functions are $4jK'$-periodic, e.g. $\mathrm{cd}(x, k) = \mathrm{cd}(x + 2jK', k)$.

The real and imaginary quarter periods create a virtual grid on the complex

plane (Fig. 9.41). We will be particularly interested in the values that Jacobian elliptic functions are taking along the lines of this grid.

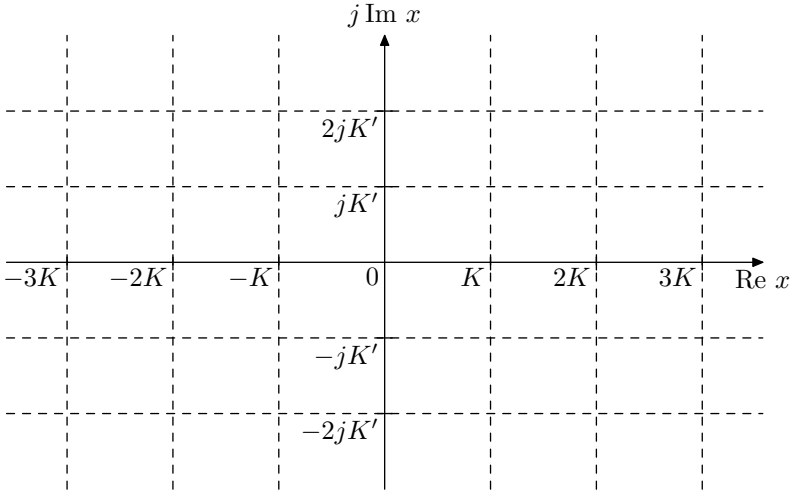


Figure 9.41: Quarter-period grid.

Let's start with $\mathrm{cd}\,x$. It turns out that the values of $\mathrm{cd}\,x$ on this grid are always equal to the (possibly scaled by some real or imaginary coefficient) values of one of the six introduced Jacobian functions evaluated for the real or the imaginary part of $\mathrm{cd}\,x$. Fig. 9.42 illustrates.

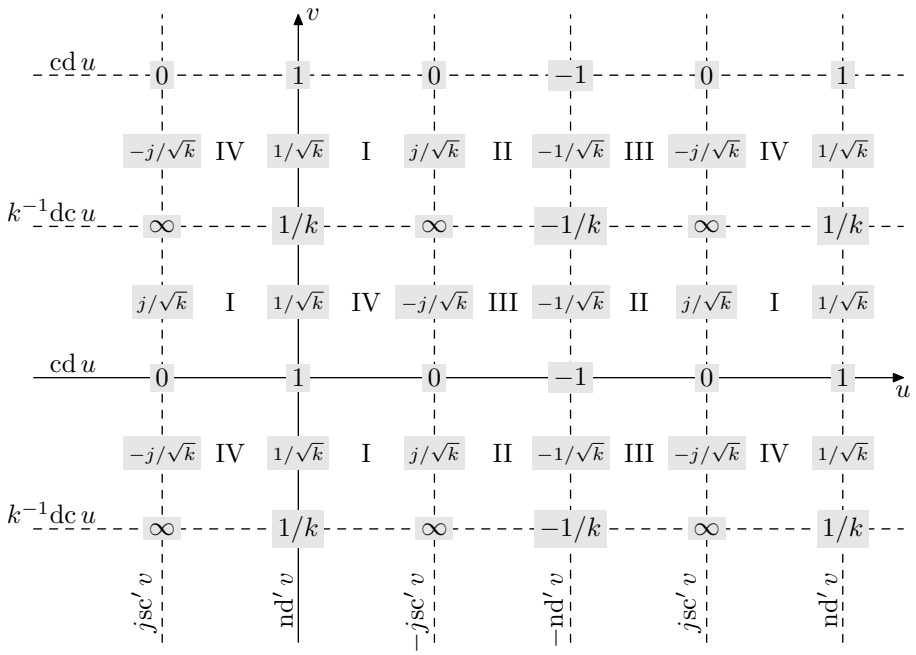


Figure 9.42: Values of $\mathrm{cd}\,x = \mathrm{cd}(u+jv)$ on the quarter-period grid.

The expressions at the ends of the quarter-grid lines in Fig. 9.42 are what $\mathrm{cd}(x,k)$ is equal to on each of these lines, where the notation is $u = \mathrm{Re}\, x$, $v = \mathrm{Im}\, x$. E.g. for $x = u + jK'$ the function's value is $\mathrm{cd}\, x = \mathrm{cd}(u+jK') = k^{-1}\,\mathrm{dc}\, u$. For $x = K$ the function's value is $\mathrm{cd}\, x = \mathrm{cd}(K+jv) = -j\,\mathrm{sc}'\, v = -j\,\mathrm{sc}(v,k')$, that is the primed notation denotes the usage of the complementary elliptic modulus. Apparently, the complementary modulus needs to be used for all grid lines parallel to the imaginary axis, since the quarter period in that direction is K'.

The roman numerals in the middle of the grid cells denote the complex quadrant to which the values of $\mathrm{cd}\, x$ belong for x inside the respective grid cell. The quadrants are numbered starting from the positive real semiaxis in the counterclockwise direction. Fig. 9.42 also shows the function values at the intersections of the grid lines. Additionally the values exactly in the middle between the horizontal grid lines are shown. E.g. $\mathrm{cd}(jK'/2) = 1/\sqrt{k}$. The readers are encouraged to compare the values listed in Fig. 9.42 to the graphs in Figs. 9.35 through 9.40.

Similarly to how the trigonometric sine is obtained from the trigonometric cosine by a shift by the quarter-period $\pi/2$ (which also holds for complex arguments), the Jacobian sine is obtained from the Jacobian cosine by a shift by the quarter period K: $\mathrm{sn}\, x = \mathrm{cd}(x-K)$. Respectively, the content of Fig. 9.42 becomes shifted by K resulting in the picture in Fig. 9.43.

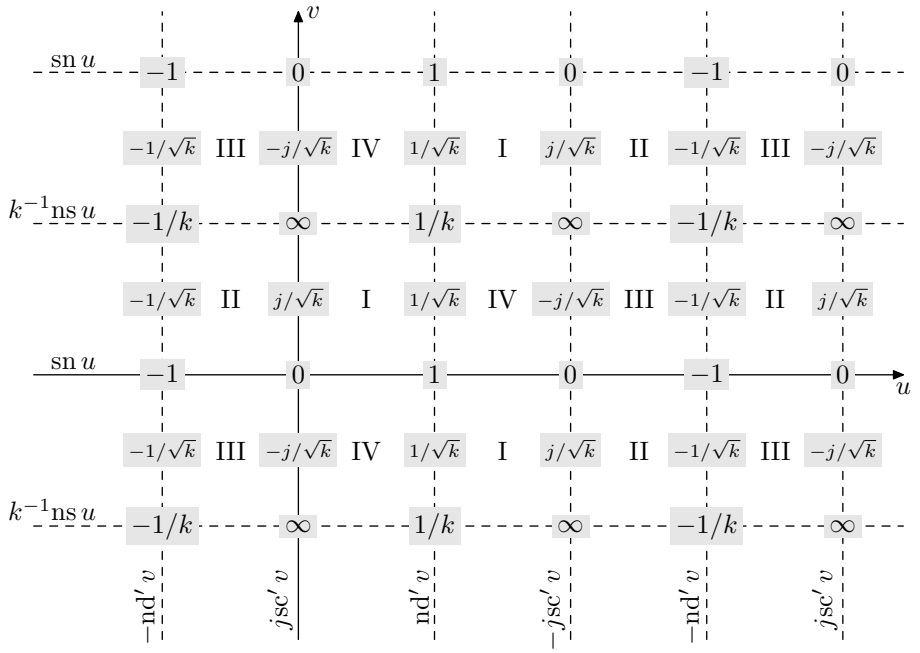


Figure 9.43: Values of $\mathrm{sn}\, x = \mathrm{sn}(u+jv)$ on the quarter-period grid.

From Fig. 9.42 one could notice that $\mathrm{cd}(jv,k) = \mathrm{nd}(v,k')$. It turns out that this equality holds not only for real v but for any complex v. That is, Jacobian hyperbolic cosine can be obtained from Jacobian cosine by rotation of

the complex plane by 90° and swapping of k and k' (which effectively swaps K and K').[11] Fig. 9.44 illustrates.

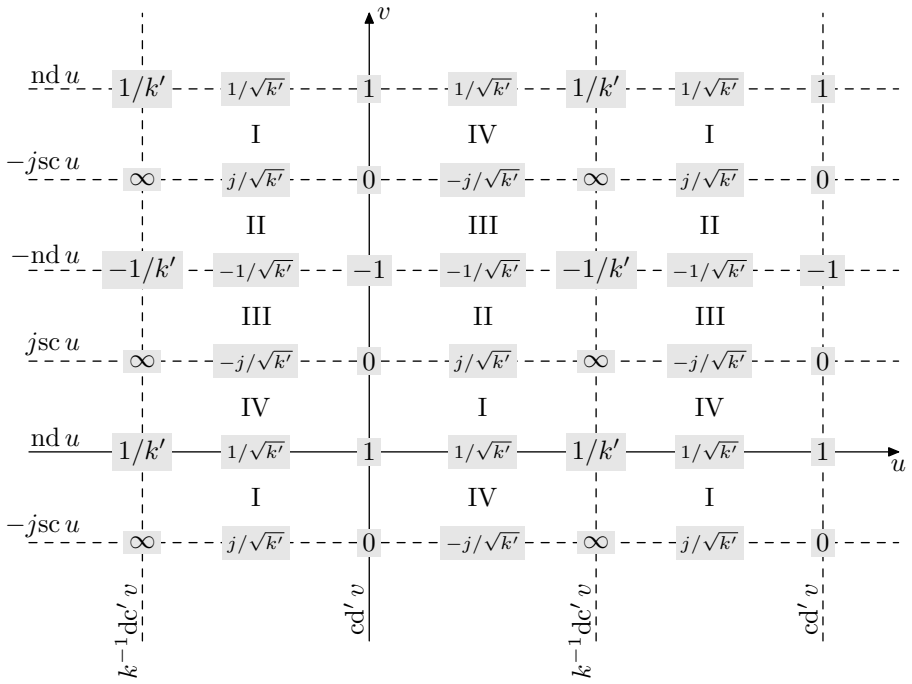


Figure 9.44: Values of $\operatorname{nd} x = \operatorname{nd}(u + jv)$ on the quarter-period grid.

In a similar fashion, in Fig. 9.43 one could notice that $\operatorname{sn}(jv, k) = j \operatorname{sc}(v, k')$. This equality also holds not only for real v but for any complex v. That is, Jacobian hyperbolic sine can be obtained from Jacobian sine by rotation of the complex plane by 90° clockwise, swapping of k and k' and dividing the result by j (or, equivalently, multiplying by $-j$). Alternatively, recalling that sn is obtained from cd by a shift by a real quarter period, we could have simply shifted the content of Fig. 9.44 downwards by an imaginary quarter period and divided it by j. Fig. 9.45 illustrates.

The functions $\operatorname{dc} x$ and $\operatorname{ns} x$ are easily obtainable from Figs. 9.42 and 9.43 by a shift by one imaginary period (and a multiplication by k).

Properties of Jacobian elliptic functions

There are lots of analogies between trigonometric/hyperbolic and Jacobian elliptic functions including similarities between their shapes, which one can see from Figs. 9.35 through 9.40. We are going to list some of the properties of Jacobian elliptic functions comparing them against similar properties of their trigonometric/hyperbolic counterparts, where possible. The value of the argument x will be assumed complex, unless otherwise noted. It is highly recommended to refer

[11] Both cd and nd are even functions. Therefore it doesn't matter in which direction to rotate by 90°.

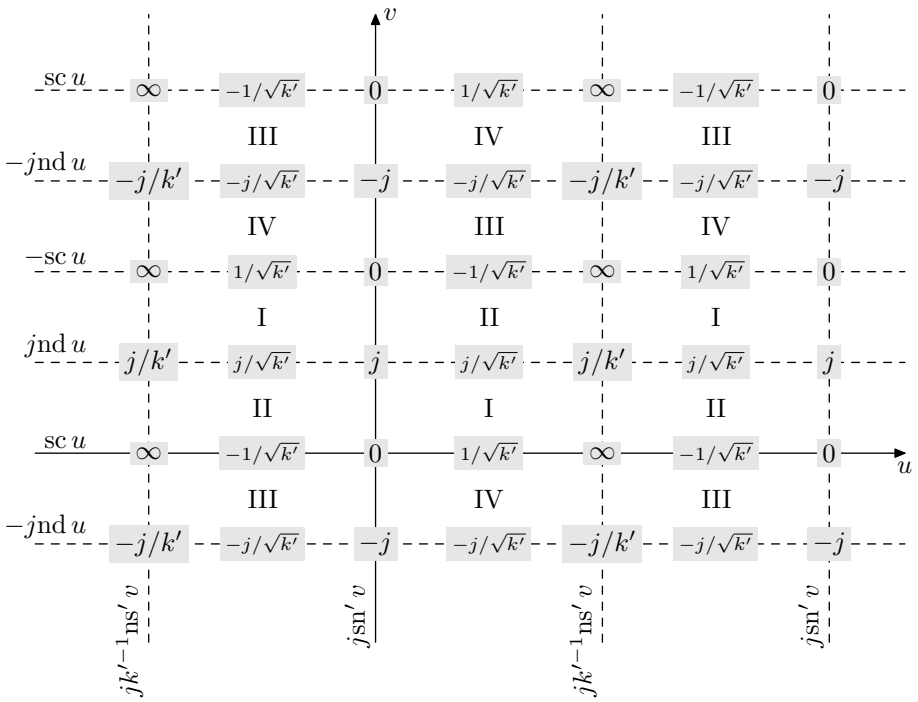


Figure 9.45: Values of $\operatorname{sc} x = \operatorname{sc}(u + jv)$ on the quarter-period grid.

to Figs. 9.35 through 9.40 and to Figs. 9.42 through 9.45 while studying the properties below.

- Reduction to trigonometric/hyperbolic functions at $k \to 0$ or $k \to 1$:

$$\operatorname{sn}(x, 0) = \sin x \tag{9.62a}$$
$$\operatorname{cd}(x, 0) = \cos x \tag{9.62b}$$
$$\operatorname{sc}(x, 0) = \tan x \tag{9.62c}$$
$$\operatorname{nd}(x, 0) \equiv 1 \tag{9.62d}$$
$$\operatorname{sc}(x, 1) = \sinh x \tag{9.62e}$$
$$\operatorname{nd}(x, 1) = \cosh x \tag{9.62f}$$
$$\operatorname{ns}(x, 0) = \csc x \tag{9.62g}$$
$$\operatorname{dc}(x, 0) = \sec x \tag{9.62h}$$

- The functions are analytic (except at their poles).

- Real function values for real argument $x \in \mathbb{R}$:

$$\operatorname{sn} x \in \mathbb{R} \qquad \sin x \in \mathbb{R}$$
$$\text{etc.}$$

- The functions commute with complex conjugation:

$$\operatorname{sn} x^* = (\operatorname{sn} x)^* \qquad \sin x^* = (\sin x)^*$$

etc.

- Odd/even symmetries

$$\begin{aligned} \mathrm{sn}(-x) &= -\mathrm{sn}\,x & \sin(-x) &= -\sin x \\ \mathrm{cd}(-x) &= \mathrm{cd}\,x & \cos(-x) &= \cos x \\ \mathrm{sc}(-x) &= -\mathrm{sc}\,x & \sinh(-x) &= -\sinh x \\ \mathrm{nd}(-x) &= \mathrm{nd}\,x & \cosh(-x) &= \cosh x \end{aligned}$$

- Imaginary argument

$$\mathrm{sn}(jx,k) = j\,\mathrm{sc}(x,k') \qquad \sin(jx) = j\sinh x \tag{9.63a}$$
$$\mathrm{cd}(jx,k) = \mathrm{nd}(x,k') \qquad \cos(jx) = \cosh x \tag{9.63b}$$
$$\mathrm{sc}(jx,k) = j\,\mathrm{sn}(x,k') \qquad \sinh(jx) = j\sin x \tag{9.63c}$$
$$\mathrm{nd}(jx,k) = \mathrm{cd}(x,k') \qquad \cosh(jx) = \cos x \tag{9.63d}$$

where we intuitively assume $x \in \mathbb{R}$, although the properties hold for any $x \in \mathbb{C}$.

- Periodicity along real axis:

$$\begin{aligned} \mathrm{sn}(x+4K) &= \mathrm{sn}\,x & \sin(x+2\pi) &= \sin x \\ \mathrm{cd}(x+4K) &= \mathrm{cd}\,x & \cos(x+2\pi) &= \cos x \\ \mathrm{sc}(x+2K) &= \mathrm{sc}\,x & &\text{n/a} \\ \mathrm{nd}(x+2K) &= \mathrm{nd}\,x & &\text{n/a} \end{aligned}$$

and along imaginary axis:

$$\begin{aligned} \mathrm{sn}(x+2jK') &= \mathrm{sn}\,x & &\text{n/a} \\ \mathrm{cd}(x+2jK') &= \mathrm{cd}\,x & &\text{n/a} \\ \mathrm{sc}(x+4jK') &= \mathrm{sc}\,x & \sinh(x+2j\pi) &= \sinh x \\ \mathrm{nd}(x+4jK') &= \mathrm{nd}\,x & \cosh(x+2j\pi) &= \cos xh \end{aligned}$$

Note that the periodicity property of $\mathrm{sn}\,x$ and $\mathrm{cd}\,x$ along the imaginary axis is the dual of the periodicity property of $\mathrm{sc}\,x$ and $\mathrm{nd}\,x$ along the real axis, the duality arising from the imaginary argument property.

- Shift by function's half-period[12] in the real direction

$$\mathrm{sn}(x \pm 2K) = -\mathrm{sn}\,x \qquad \sin(x \pm \pi) = -\sin x \tag{9.65a}$$
$$\mathrm{cd}(x \pm 2K) = -\mathrm{cd}\,x \qquad \cos(x \pm \pi) = -\cos x \tag{9.65b}$$
$$\mathrm{sc}(x \pm K) = -1/k'\,\mathrm{sc}\,x \qquad \text{n/a} \tag{9.65c}$$
$$\mathrm{nd}(x \pm K) = 1/k'\,\mathrm{nd}\,x \qquad \text{n/a} \tag{9.65d}$$

and in the imaginary direction

$$\mathrm{sn}(x \pm jK') = 1/k\,\mathrm{sn}\,x \qquad \text{n/a} \tag{9.65e}$$

[12]Note that here (and further where we specifically refer to *function's* period, or half- or quarter-period) we mean the period of the function itself, rather than one of the least common periods $4K$ and $4K'$ of all Jacobian elliptic functions.

$$\operatorname{cd}(x \pm jK') = 1/k \operatorname{cd} x \qquad \text{n/a} \tag{9.65f}$$
$$\operatorname{sc}(x \pm 2jK') = -\operatorname{sc} x \qquad \sinh(x \pm j\pi) = -\sinh x \tag{9.65g}$$
$$\operatorname{nd}(x \pm 2jK') = -\operatorname{nd} x \qquad \cosh(x \pm j\pi) = -\cosh x \tag{9.65h}$$

- Shift by function's quarter-period

$$\operatorname{sn}(x + K) = \operatorname{cd} x \qquad \sin(x + \pi/2) = \cos x \tag{9.66a}$$
$$\operatorname{cd}(x - K) = \operatorname{sn} x \qquad \cos(x - \pi/2) = \sin x \tag{9.66b}$$
$$\operatorname{sc}(x + jK') = j \operatorname{nd} x \qquad \sinh(x + j\pi/2) = j \cosh x \tag{9.66c}$$
$$\operatorname{nd}(x + jK') = j \operatorname{sc} x \qquad \cosh(x + j\pi/2) = j \sinh x \tag{9.66d}$$

- Symmetry around function's quarter-period point (this follows from the odd/even symmetries and the shift by function's half-period property) in the real direction:

$$\operatorname{sn}(2K - x) = \operatorname{sn}(x) \qquad \sin(\pi - x) = \sin x \tag{9.67a}$$
$$\operatorname{cd}(2K - x) = -\operatorname{cd}(x) \qquad \cos(\pi - x) = -\cos x \tag{9.67b}$$
$$\operatorname{sc} x \operatorname{sc}(K - x) = 1/k' \qquad \text{n/a} \tag{9.67c}$$
$$\operatorname{nd} x \operatorname{nd}(K - x) = 1/k' \qquad \text{n/a} \tag{9.67d}$$

and in the imaginary direction:

$$\operatorname{sn} x \operatorname{sn}(jK' - x) = -1/k \qquad \text{n/a} \tag{9.67e}$$
$$\operatorname{cd} x \operatorname{cd}(jK' - x) = 1/k \qquad \text{n/a} \tag{9.67f}$$
$$\operatorname{sc}(2jK' - x) = \operatorname{sc} x \qquad \sinh(j\pi - x) = \sinh x \tag{9.67g}$$
$$\operatorname{nd}(2jK' - x) = -\operatorname{nd} x \qquad \cosh(j\pi - x) = -\cosh x \tag{9.67h}$$

- Pythagorean theorem

$$\operatorname{sn}^2 x + \operatorname{cd}^2 x = 1 + k^2 \operatorname{sn}^2 x \operatorname{cd}^2 x \qquad \sin^2 x + \cos^2 x = 1 \tag{9.68}$$

(we won't need the respective properties for the "hyperbolic" functions).

There is another useful Pythagorean-like identity:

$$k^2 \operatorname{cd}^2 x + k'^2 \operatorname{nd}^2 x = 1 \tag{9.69}$$

- Sum of arguments

$$\operatorname{cd}(x + y, k) = \frac{\operatorname{cd} x \operatorname{cd} y - \operatorname{sn} x \operatorname{sn} y}{1 - k^2 \operatorname{sn} x \operatorname{sn} y \operatorname{cd} x \operatorname{cd} y} \tag{9.70}$$
$$\cos(x + y) = \cos x \cos y - \sin x \sin y$$

(we won't need the respective properties for the other functions).

- Complex argument

$$\operatorname{cd}(u + jv, k) = \frac{\operatorname{cd} u \operatorname{nd}' v - j \operatorname{sn} u \operatorname{sc}' v}{1 - jk^2 \operatorname{sn} u \operatorname{sc}' v \operatorname{cd} u \operatorname{nd}' v} \tag{9.71}$$
$$\cos(u + jv) = \cos u \cosh v - j \sin u \sinh v$$

where $\operatorname{nd}' v = \operatorname{nd}(v, k')$, $\operatorname{sc}' v = \operatorname{sc}(v, k')$. This is a direct corollary of (9.70). One can use (9.71) to show that the values of cd on a single grid cell in Fig. 9.42 belong to one and the same complex quadrant.

- Logarithmic derivative

$$\frac{\mathrm{d}}{\mathrm{d}x}\ln\operatorname{cd}x = -k'^2\operatorname{sc}x\operatorname{nd}x \qquad \frac{\mathrm{d}}{\mathrm{d}x}\ln\cos x = -\tan x \tag{9.72a}$$

$$\frac{\mathrm{d}^2}{\mathrm{d}x^2}\ln\operatorname{cd}x = k\left(k\operatorname{cd}^2 x - \frac{1}{k\operatorname{cd}^2 x}\right) \qquad \frac{\mathrm{d}^2}{\mathrm{d}x^2}\ln\cos x = -\frac{1}{\cos^2 x} \tag{9.72b}$$

Periodicity

As we already mentioned, Jacobian elliptic functions are periodic in real and imaginary direction. E.g. $\operatorname{cd}x$ is $4K$- and $2jK'$-periodic. Thus the periods of $\operatorname{cd}x$ are rectangles in the complex plane, the horizontal dimension of each rectangle being equal to $4K$ and the vertical dimension being equal to $2K'$. Fig. 9.46 illustrates.

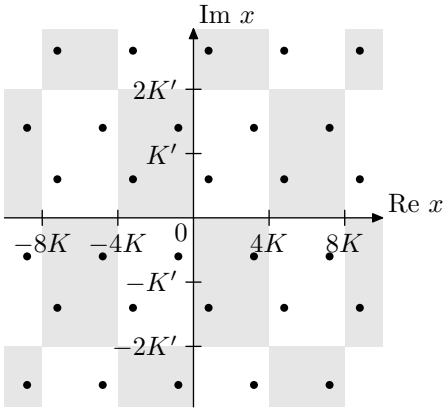


Figure 9.46: Periods of $\operatorname{cd}x$ in the complex plane. All dots are preimages of one and the same value.

Due to the even symmetry of the elliptic cosine, almost every value occurs twice on a period (as illustrated by the dots in Fig. 9.46). That is if the value y occurs at x (that is $y = \operatorname{cd}x$), then y also occurs at $-x$. The exceptions are being $\operatorname{cd}x = \pm 1$ and $\operatorname{cd}x = \pm 1/\sqrt{k}$, which are mapped to themselves by $x \leftarrow -x$ if the periodicities of $\operatorname{cd}x$ are taken into account.

Similar considerations apply to $\operatorname{sn}x$, $\operatorname{sc}x$ and $\operatorname{nd}x$.

Preimages of the real line

By (9.71) $\operatorname{cd}x$ attains purely real values iff $x \in \mathbb{R}$ or $\operatorname{Re}x = 2Kn$ where $n \in \mathbb{Z}$, which is also illustrated by Fig. 9.42. Similarly to $\cos x$, we would like to choose a principal preimage of the real line with respect to the transformation $y = \operatorname{cd}x$. Since $\operatorname{cd}x \to \cos x$ for $k \to 0$, we would like the principal real line preimage for $\operatorname{cd}x$ to approach to the respective principal preimage for $\cos x$ as $k \to 0$. Under this requirement there is only one choice, which is shown in Fig. 9.47.

This principal preimage of the real axis thereby consists of five parts:

$$x \in [0, 2K] \qquad \Longleftrightarrow y \in [-1, 1]$$
$$x \in [0, jK'] \qquad \Longleftrightarrow y \in [1, 1/k]$$

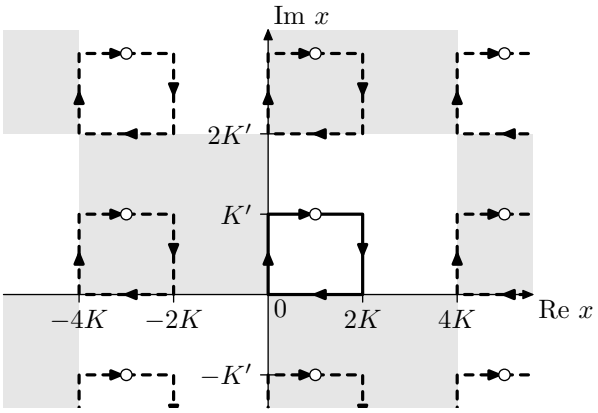


Figure 9.47: The principal preimage (solid line) of the real axis, with respect to $y = \operatorname{cd} x$, and its periodic repetitions (dashed lines).

$$\begin{aligned} x \in [2K, 2K + jK'] &\iff y \in [-1/k, 1] \\ x \in [jK', jK' + K) &\iff y \in [1/k, +\infty) \\ x \in (jK' + K, jK' + 2K] &\iff y \in (-\infty, -1/k] \end{aligned}$$

where Fig. 9.42 can serve as additional reference.

The punctured point at $x = K + jK'$ in Fig. 9.47 corresponds to $y = \infty$. In principle it can be included into the preimage if we consider the extended complex plane $\mathbb{C} \cup \infty$ as the codomain of $\operatorname{cd} x$, in which case the preimage consists only of four parts:

$$\begin{aligned} x \in [0, 2K] &\iff y \in [-1, 1] \\ x \in [0, jK'] &\iff y \in [1, 1/k] \\ x \in [2K, 2K + jK'] &\iff y \in [-1/k, 1] \\ x \in [jK', jK' + 2K] &\iff y \in [1/k, -1/k] \end{aligned}$$

where $[1/k, -1/k] = [1/k, +\infty) \cup \infty \cup (-\infty, -1/k]$ denotes a range on the real Riemann circle containing the infinity in its middle.

As with $\cos x$, the principal preimage alone doesn't cover all preimage points of the real line. Neither does it if we add its periodic repetitions in Fig. 9.47, since we are convering only half of the entire length of each of the lines $\operatorname{Re} x = \pi n$. We can cover the remaining halves by rotating all preimages in Fig. 9.47 around the origin, which corresponds to multiplication of all points x by -1. Notice that by adding periodic repetitions we addressed the periodicity of $\operatorname{cd} x$, while by adding the preimages multiplied by -1 we addressed the evenness property of $\operatorname{cd} x$.

9.10 Normalized Jacobian elliptic functions

In Fig. 9.42 we can observe that the "four building blocks" of the Jacobian cosine's values on the quarter-period grid lines are cd, $k^{-1}\operatorname{dc}$, nd', $j\operatorname{sc}'$, $-\operatorname{nd}'$ and $-j\operatorname{sc}'$. Plotting these functions in the arctangent scale we obtain the picture in Fig. 9.48.

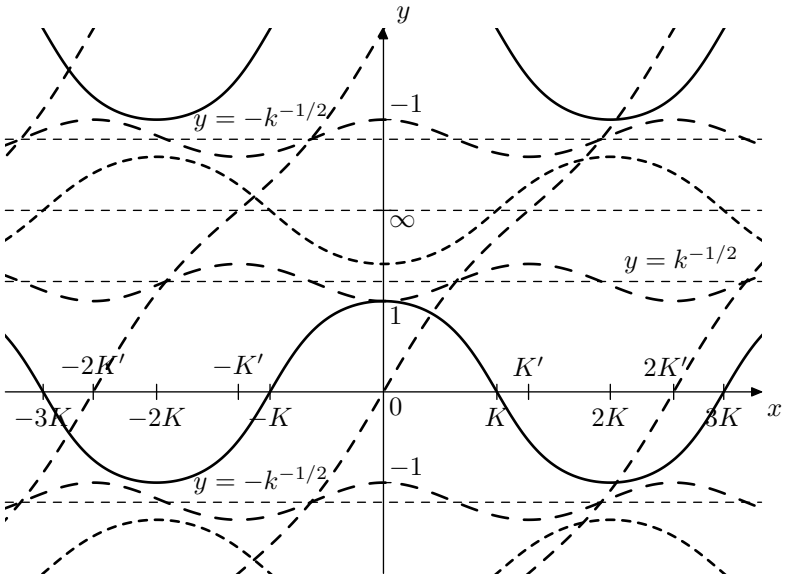


Figure 9.48: $\operatorname{cd}(x,k)$, $\operatorname{nd}(x,k')$, $k^{-1}\operatorname{dc}(x,k)$ and $\operatorname{sc}(x,k')$.

The collection of the function graphs in Fig. 9.48 has obvious symmetries with respect to the horizontal lines $y = 0$ and $y = \infty$. It is also *approximately* symmetric with respect to $y = \pm 1/\sqrt{k}$ (where, since $k < 1$, it follows that $1/\sqrt{k} > 1$, so the line $y = 1/\sqrt{k}$ is located above $y = 1$). This approximate symmetry obviously arises from the reciprocal symmetries due to (9.67):

$$\operatorname{nd}(K' - x, k') = 1/k \operatorname{nd}(x, k') \tag{9.73a}$$

$$\operatorname{sc}(K' - x, k') = 1/k \operatorname{sc}(x, k') \tag{9.73b}$$

$$k^{-1} \operatorname{dc}(x, k) = 1/k \operatorname{cd}(x, k) \tag{9.73c}$$

(where (9.73c) is not due to (9.67) but simply follows from the definition of the dc function: $\operatorname{dc}(x,k) = 1/\operatorname{cd}(x,k)$).

Therefore the centers of this reciprocal symmetry are at $\pm 1/\sqrt{k}$. By multiplying all functions plotted in Fig. 9.48 by $\sqrt{k}$ we will shift the centers of the reciprocal symmetry to $y = \pm 1$. Thus consideration motivates the introduction of *normalized Jacobian elliptic functions*

$$\begin{aligned}
\overline{\operatorname{cd}}(x,k) &= \sqrt{k}\operatorname{cd}(x,k) \\
\overline{\operatorname{sn}}(x,k) &= \sqrt{k}\operatorname{sn}(x,k) \\
\overline{\operatorname{dc}}(x,k) &= 1/\sqrt{k}\cdot\operatorname{dc}(x,k) = 1/\overline{\operatorname{cd}}(x,k) \\
\overline{\operatorname{ns}}(x,k) &= 1/\sqrt{k}\cdot\operatorname{ns}(x,k) = 1/\overline{\operatorname{sn}}(x,k) \\
\overline{\operatorname{sc}}(x,k) &= \sqrt{k'}\operatorname{sc}(x,k) \\
\overline{\operatorname{nd}}(x,k) &= \sqrt{k'}\operatorname{nd}(x,k)
\end{aligned}$$

(note that for the "hyperbolic" functions we are using $\sqrt{k'}$ rather than $\sqrt{k}$ for the normalization!). Thereby (9.73) become:

$$\overline{\operatorname{nd}}(K' - x, k') = 1/\overline{\operatorname{nd}}(x, k') \tag{9.74a}$$

$$\overline{\mathrm{sc}}(K'-x,k') = 1/\,\overline{\mathrm{sc}}(x,k') \tag{9.74b}$$
$$\overline{\mathrm{dc}}(x,k) = 1/\,\overline{\mathrm{cd}}(x,k) \tag{9.74c}$$

and Fig. 9.48 turns into Fig. 9.49.

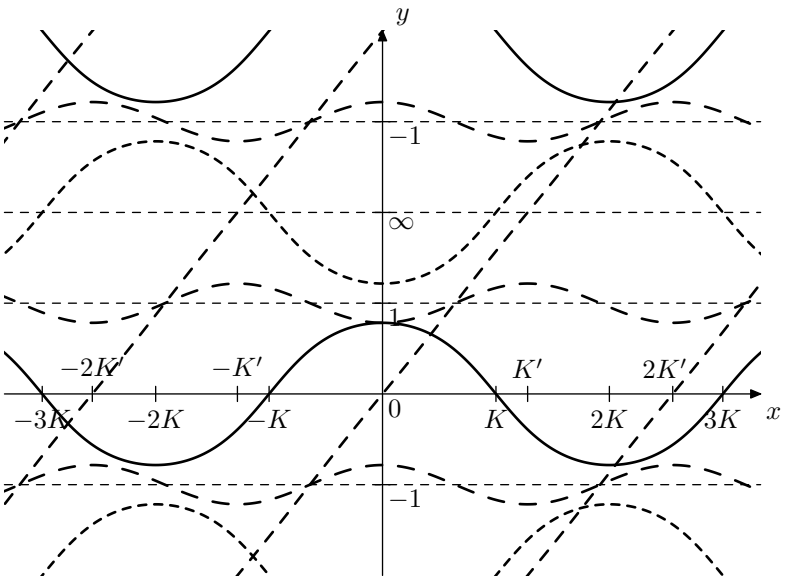


Figure 9.49: $\overline{\mathrm{cd}}(x,k)$, $\overline{\mathrm{nd}}(x,k')$, $\overline{\mathrm{dc}}(x,k)$ and $\overline{\mathrm{sc}}(x,k')$.

Apparently, $\overline{\mathrm{cd}}$, $\overline{\mathrm{dc}}$, $\overline{\mathrm{nd}}'$, $j\,\overline{\mathrm{sc}}'$, $-\,\overline{\mathrm{nd}}'$ and $-j\,\overline{\mathrm{sc}}'$ are the building blocks of $\overline{\mathrm{cd}}\,x$ in the same way how cd, $k^{-1}\,\mathrm{dc}$, nd', $j\,\mathrm{sc}'$, $-\,\mathrm{nd}'$ and $-j\,\mathrm{sc}'$ are the building blocks of $\mathrm{cd}\,x$. Fig. 9.50 illustrates. Notice that we don't need non-unity scaling by k^{-1} anymore, only shifts, rotations and scaling by $\pm j$ are required to convert between the respective functions. Fig. 9.51 provides a similar illustration for $\overline{\mathrm{nd}}$. The diagrams Fig. 9.43 and 9.45 are transformed in a similar way.

For normalized elliptic functions the reciprocal symmetries of (9.67) take the form

$$\overline{\mathrm{sc}}\,x\;\overline{\mathrm{sc}}(K-x) = 1 \tag{9.75a}$$
$$\overline{\mathrm{nd}}\,x\;\overline{\mathrm{nd}}(K-x) = 1 \tag{9.75b}$$
$$\overline{\mathrm{sn}}\,x\;\overline{\mathrm{sn}}(jK'-x) = -1 \tag{9.75c}$$
$$\overline{\mathrm{cd}}\,x\;\overline{\mathrm{cd}}(jK'-x) = 1 \tag{9.75d}$$

with the analogous shift properties (9.65) taking the form

$$\overline{\mathrm{sc}}\,x\;\overline{\mathrm{sc}}(x\pm K) = -1 \tag{9.76a}$$
$$\overline{\mathrm{nd}}\,x\;\overline{\mathrm{nd}}(x\pm K) = 1 \tag{9.76b}$$
$$\overline{\mathrm{sn}}\,x\;\overline{\mathrm{sn}}(x\pm jK') = 1 \tag{9.76c}$$
$$\overline{\mathrm{cd}}\,x\;\overline{\mathrm{cd}}(x\pm jK') = 1 \tag{9.76d}$$

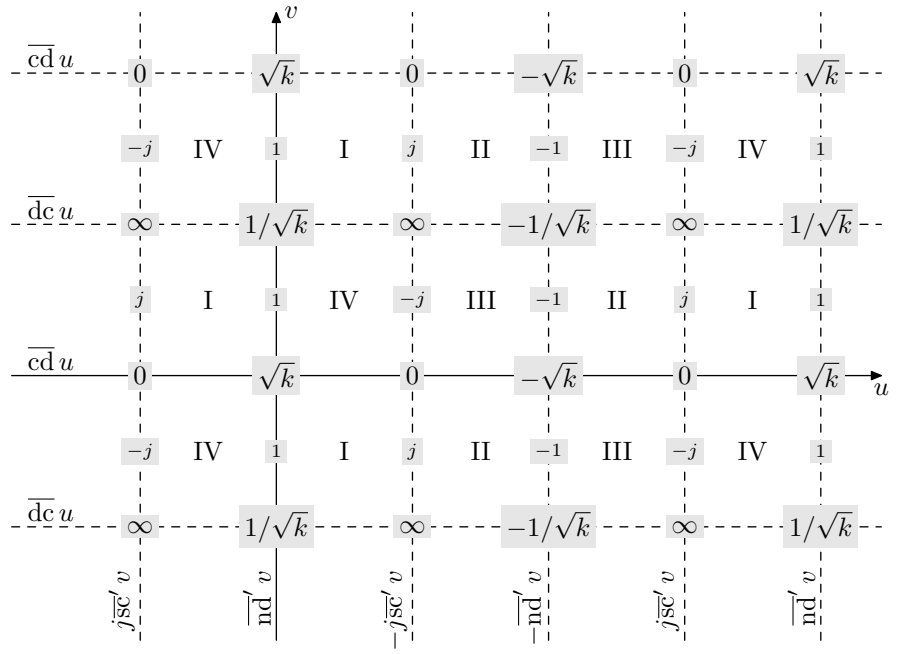


Figure 9.50: Values of $\overline{\mathrm{cd}}\, x = \overline{\mathrm{cd}}(u + jv)$ on the quarter-period grid (compare to Fig. 9.42).

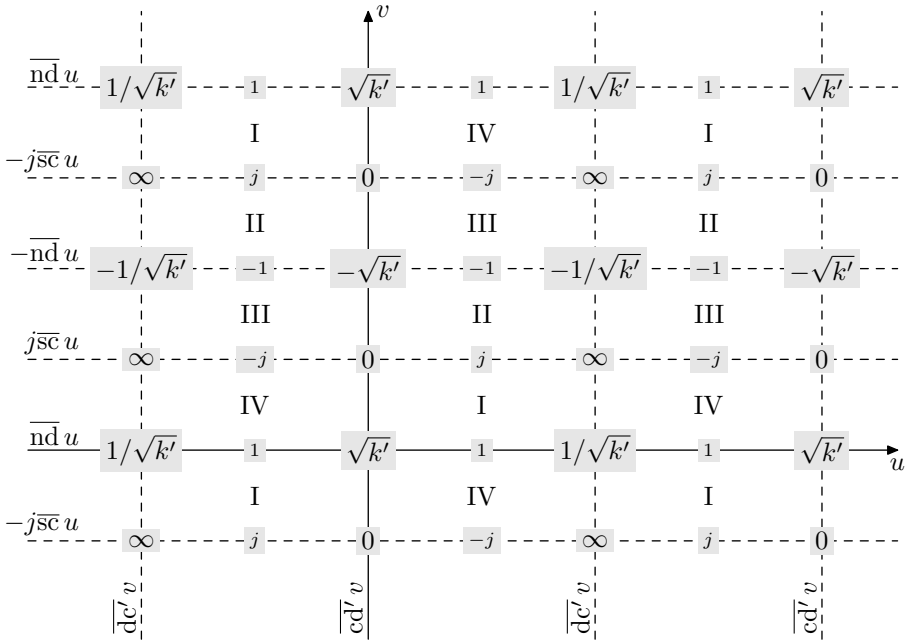


Figure 9.51: Values of $\overline{\mathrm{nd}}\, x = \overline{\mathrm{nd}}(u + jv)$ on the quarter-period grid (compare to Fig. 9.44).

Derivatives

In terms of normalized functions the logarithmic derivative formulas (9.72) take the form

$$\frac{\mathrm{d}}{\mathrm{d}x}\ln\overline{\mathrm{cd}}\,x = -k'\,\overline{\mathrm{sc}}\,x\,\overline{\mathrm{nd}}\,x \tag{9.77a}$$

$$\frac{\mathrm{d}^2}{\mathrm{d}x^2}\ln\overline{\mathrm{cd}}\,x = k\left(\overline{\mathrm{cd}}^2\,x - \frac{1}{\overline{\mathrm{cd}}^2\,x}\right) \tag{9.77b}$$

By checking the complex quadrants of the values of sc and nd in Figs. 9.44 and 9.45, one could establish that the first logarithmic derivative is lying in the lower complex semiplane on even imaginary periods and in the upper complex semiplane on odd imaginary periods:

$$\operatorname{Im}\frac{\mathrm{d}}{\mathrm{d}x}\ln\overline{\mathrm{cd}}\,x < 0 \qquad \text{for } \operatorname{Im}x \in (2n'K', (2n'+1)K')$$

$$\operatorname{Im}\frac{\mathrm{d}}{\mathrm{d}x}\ln\overline{\mathrm{cd}}\,x > 0 \qquad \text{for } \operatorname{Im}x \in ((2n'+1)K', (2n'+2)K')$$

or simply

$$\operatorname{sgn}\operatorname{Im}\frac{\mathrm{d}}{\mathrm{d}x}\ln\overline{\mathrm{cd}}\,x = (-1)^{n'+1} \qquad \text{for } \operatorname{Im}x \in (2n'K', (2n'+1)K') \tag{9.78}$$

where n' is the imaginary quarter period index.

The second logarithmic derivative is apparently lying in the upper complex semiplane if $\overline{\mathrm{cd}}\,x$ is in the I or III complex quadrant and in the lower complex semiplane if $\overline{\mathrm{cd}}\,x$ is in the II or IV complex quadrant:

$$\operatorname{Im}\frac{\mathrm{d}}{\mathrm{d}x^2}\ln\overline{\mathrm{cd}}\,x > 0 \qquad \text{if } \mathrm{cd}\,x \in \text{I or III}$$

$$\operatorname{Im}\frac{\mathrm{d}}{\mathrm{d}x^2}\ln\overline{\mathrm{cd}}\,x < 0 \qquad \text{if } \mathrm{cd}\,x \in \text{II or IV}$$

or, using Fig. 9.50,

$$\operatorname{sgn}\operatorname{Im}\frac{\mathrm{d}}{\mathrm{d}x^2}\ln\overline{\mathrm{cd}}\,x = (-1)^{n+n'+1} \tag{9.79}$$

where n and n' are respectively the real and imaginary quarter period indices.

Horizontal and vertical preimage lines of $\overline{\mathrm{cd}}\,x$

The formulas (9.77) can be used to obtain more information about the behavior of $\overline{\mathrm{cd}}\,x$ (and respectively $\mathrm{cd}\,x$) on its quarter periods. For $\cos x$ this kind of information can be directly obtained from the complex argument formula (9.29a). For $\mathrm{cd}\,x$ the same formula (9.71) is a bit more complicated and cannot be as easily used for analysis.

Given $u = \operatorname{Re}x$, $v = \operatorname{Im}x$ we obtain:

$$\frac{\mathrm{d}}{\mathrm{d}u}\arg\overline{\mathrm{cd}}(u+jv) = \frac{\mathrm{d}}{\mathrm{d}u}\operatorname{Im}\ln\overline{\mathrm{cd}}(u+jv) = \operatorname{Im}\frac{\mathrm{d}}{\mathrm{d}u}\ln\overline{\mathrm{cd}}(u+jv) =$$

$$= \operatorname{Im}\frac{\mathrm{d}}{\mathrm{d}x}\ln\overline{\mathrm{cd}}\,x \tag{9.80a}$$

$$\frac{\mathrm{d}}{\mathrm{d}v}\ln|\overline{\mathrm{cd}}(u+jv)| = \frac{\mathrm{d}}{\mathrm{d}v}\operatorname{Re}\ln\overline{\mathrm{cd}}(u+jv) = \operatorname{Re}\frac{\mathrm{d}}{\mathrm{d}v}\ln\overline{\mathrm{cd}}(u+jv) =$$
$$= \operatorname{Re} j\frac{\mathrm{d}}{j\mathrm{d}v}\ln\overline{\mathrm{cd}}(u+jv) = \operatorname{Re} j\frac{\mathrm{d}}{\mathrm{d}x}\ln\overline{\mathrm{cd}}(u+jv) =$$
$$= -\operatorname{Im}\frac{\mathrm{d}}{\mathrm{d}x}\ln\overline{\mathrm{cd}}\,x \tag{9.80b}$$
$$\frac{\mathrm{d}^2}{\mathrm{d}u^2}\arg\overline{\mathrm{cd}}(u+jv) = \frac{\mathrm{d}^2}{\mathrm{d}u^2}\operatorname{Im}\ln\overline{\mathrm{cd}}(u+jv) = \operatorname{Im}\frac{\mathrm{d}^2}{\mathrm{d}u^2}\ln\overline{\mathrm{cd}}(u+jv) =$$
$$= \operatorname{Im}\frac{\mathrm{d}^2}{\mathrm{d}x^2}\ln\overline{\mathrm{cd}}\,x \tag{9.80c}$$
$$\frac{\mathrm{d}^2}{\mathrm{d}v^2}\arg\overline{\mathrm{cd}}(u+jv) = \frac{\mathrm{d}^2}{\mathrm{d}v^2}\operatorname{Im}\ln\overline{\mathrm{cd}}(u+jv) = \operatorname{Im}\frac{\mathrm{d}^2}{\mathrm{d}v^2}\ln\overline{\mathrm{cd}}(u+jv) =$$
$$= -\operatorname{Im}\frac{\mathrm{d}^2}{\mathrm{d}x^2}\ln\overline{\mathrm{cd}}\,x \tag{9.80d}$$

Suppose the point $x = u+jv$ is moving horizontally to the right within the n'-th imaginary quarter period, that is $\dot{u} > 0$ and $v = \text{const} \in (2n'K', (2n'+1)K')$. Then we have the following.

- By (9.80a) and (9.78)

$$\operatorname{sgn}\frac{\mathrm{d}}{\mathrm{d}u}\arg\overline{\mathrm{cd}}(u+jv) = \operatorname{sgn}\operatorname{Im}\frac{\mathrm{d}}{\mathrm{d}x}\ln\overline{\mathrm{cd}}\,x = (-1)^{n'+1} \tag{9.81a}$$

  therefore the value of $\mathrm{cd}\,x$ is moving clockwise on even imaginary quarter periods and counterclockwise on odd imaginary quarter periods. By (9.71) and using the complex quadrants in Fig. 9.42 or 9.50 as a reference, we additionally find that

$$\arg\overline{\mathrm{cd}}(Kn+jv) = (-1)^{n'+1}\cdot\frac{\pi}{2}n \tag{9.81b}$$

  that is at integer multiples of K ($u = Kn$) the value of $\mathrm{cd}\,x$ is crossing the real and imaginary axes, starting with the real axis at $u = 0$. Fig. 9.52 illustrates. The family of curves generated by such horizontal preimage lines in shown in Fig. 9.53.

- By (9.80c) and (9.79)

$$\operatorname{sgn}\frac{\mathrm{d}}{\mathrm{d}u^2}\arg\overline{\mathrm{cd}}(u+jv) = \operatorname{sgn}\operatorname{Im}\frac{\mathrm{d}}{\mathrm{d}x^2}\ln\overline{\mathrm{cd}}\,x = (-1)^{n+n'+1} \tag{9.81c}$$

  Comparing (9.81c) to (9.81a) and (9.81b) we find that, given $\dot{u} = \text{const}$, the trajectories in Fig. 9.53 are speeding up when going away from the real axis and slowing down when going towards the real axis.

Now suppose the point $x = u + jv$ is moving in a vertical line towards the top: $\dot{v} > 0$, $u = \text{const} \in (2nK, (2n+1)K)$.

- By (9.80b) and taking into account (9.78)

$$\operatorname{sgn}\frac{\mathrm{d}}{\mathrm{d}v}|\overline{\mathrm{cd}}(u+jv)| = -\operatorname{sgn}\operatorname{Im}\frac{\mathrm{d}}{\mathrm{d}x}\ln\overline{\mathrm{cd}}\,x = (-1)^{n'} \tag{9.82a}$$

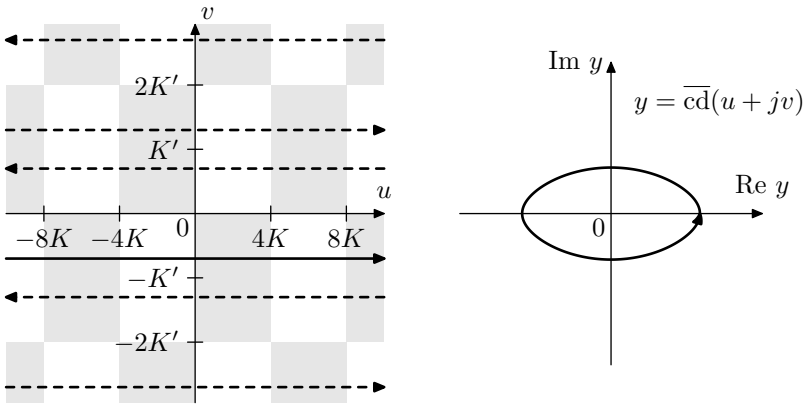


Figure 9.52: A quasielliptic trajectory and its preimages. The picture is qualitative. Particularly, the principal preimage line, shown by the solid arrow line, is actually closer to the real axis (it must be closer that $K'/2$).

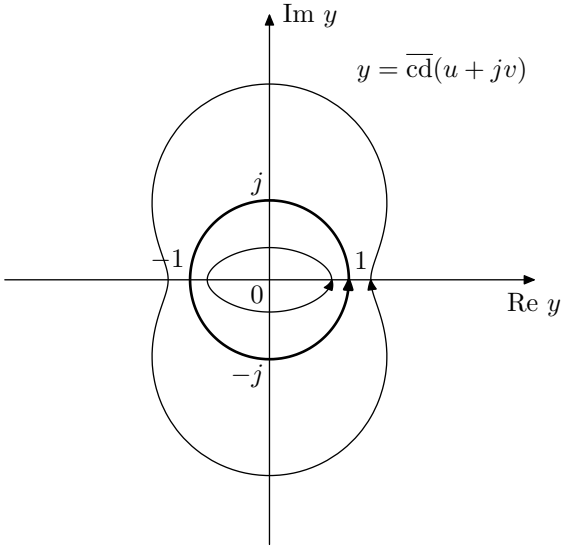


Figure 9.53: A family of quasielliptic trajectories generated from horizontal preimages $v = \text{const} \in [-K', 0]$. The unit circle trajectory occurs at $v = -K'/2$.

where n' is the imaginary quarter period index corresponding to the current value of v. That is $|\operatorname{cd} x|$ will be increasing on even imaginary quarter periods and decreasing on odd imaginary quarter periods.

In fact, the movement trajectories will be as shown in Fig. 9.54, where the movement around $y = 1$ will be occurring on even real quarter-periods and the movement around $y = -1$ will be occurring on odd real quarter-periods. At the even boundaries $u = 2nK$ the movement will be oscillating along the real line between $(-1)^n\sqrt{k}$ and $(-1)^n/\sqrt{k}$. At the odd bound-

aries $u = (2n+1)K$ the movement will be occurring along the entire imaginary axis looping through the ∞, going downwards all the time if n is even and going upwards all the time if n is odd (referring to Fig. 9.50 is recommended for understanding these boundary cases). The trajectories in Fig. 9.54 complete a full cycle over one imaginary period $2K'$ of $\overline{\mathrm{cd}}\, x$.

- By (9.80d) and (9.79)

$$\operatorname{sgn} \frac{\mathrm{d}}{\mathrm{d}v^2} \arg \overline{\mathrm{cd}}(u+jv) = -\operatorname{sgn} \operatorname{Im} \frac{\mathrm{d}}{\mathrm{d}x^2} \ln \overline{\mathrm{cd}}\, x = (-1)^{n+n'} \qquad (9.82b)$$

Equation (9.82b) means that the second derivative of $\arg \overline{\mathrm{cd}}\, x$ doesn't change sign during vertical motion within a single imaginary quarter period. This doesn't seem much, but it will be a quite useful property.

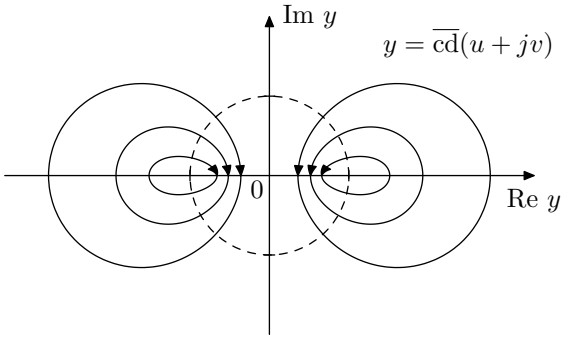


Figure 9.54: A family of trajectories generated from vertical preimages $u = \text{const}$. Notice that the trajectories intersect the unit circle (shown by the dashed line) at right angles.

Since $|\overline{\mathrm{cd}}\, x|$ monotonic in the vertical direction on a single quarter period and $\arg \overline{\mathrm{cd}}\, x$ is monotonic in the horizontal direction, it follows that within a single quarter-period grid cell the function $\overline{\mathrm{cd}}\, x$ is taking each value no more than once. Respectively, the quasielliptic curves in Fig. 9.53 are all distinct (that is they don't intersect or overlap) within a single imaginary quarter-period of the domain of $\overline{\mathrm{cd}}\, x$. Conversely, each imaginary quarter period of the domain of $\overline{\mathrm{cd}}\, x$ contains exactly one preimage of any given such curve, as shown in Fig. 9.52.

In a similar way one can argue the distinctness of the curves in Fig. 9.54.

Unit circle symmetries

In Figs. 9.53 and 9.54 we have specifically highlighted the unit circle, which is related some of the properties of $\overline{\mathrm{cd}}\, x$. It turns out that $\overline{\mathrm{cd}}\, x$ has some symmetries in respect to the unit circle and its preimage.

Let's take two points $jK'/2 + x$ and $jK'/2 + x^*$, which are located symmetrically to the line $\operatorname{Im} x = jK'/2$ on the complex plane, and consider the product

$$\overline{\mathrm{cd}}(jK'/2+x)\left(\overline{\mathrm{cd}}(jK'/2+x^*)\right)^* = \overline{\mathrm{cd}}(jK'/2+x)\left(\overline{\mathrm{cd}}(x-jK'/2)^*\right)^* =$$
$$= \overline{\mathrm{cd}}(x+jK'/2)\,\overline{\mathrm{cd}}(x-jK'/2) =$$

$$= \overline{\mathrm{cd}}(x + jK'/2)\,\overline{\mathrm{cd}}\big((x + jK'2) - jK'\big) = 1$$

where the latest is by (9.76d). That is the corresponding values of the Jacobian elliptic cosine are conjugate-reciprocal:

$$\overline{\mathrm{cd}}(jK'/2 + x)\left(\overline{\mathrm{cd}}(jK'/2 + x^*)\right)^* = 1 \tag{9.83}$$

and the line $\mathrm{Im}\, x = jK'/2$ is the axis of the conjugate-reciprocal symmetry of $\overline{\mathrm{cd}}\, x$. From the evenness property of the elliptic cosine (and the fact that x in (9.83) is arbitrary) it follows that

$$\overline{\mathrm{cd}}(-jK'/2 + x)\left(\overline{\mathrm{cd}}(-jK'/2 + x^*)\right)^* = 1$$

that is the line $\mathrm{Im}\, x = -jK'/2$ is also the axis of the conjugate-reciprocal symmetry of $\overline{\mathrm{cd}}\, x$. Since $\overline{\mathrm{cd}}\, x$ is $2K'$-periodic along the imaginary axis, any other lines of the form $\mathrm{Im}\, x = jK'/2 + K'n'$ are also the axes of the conjugate-reciprocal symmetry of $\overline{\mathrm{cd}}\, x$:

$$\overline{\mathrm{cd}}(jK'/2 + jK'n' + x)\left(\overline{\mathrm{cd}}(jK'/2 + jK'n' + x^*)\right)^* = 1 \tag{9.84}$$

Taking the absolute value of both sides of (9.84) we obtain

$$|\,\overline{\mathrm{cd}}(jK'/2 + jK'n' + x)| \cdot |\,\overline{\mathrm{cd}}(jK'/2 + jK'n' + x^*)| = 1$$

Further, assuming a purely real x (so that $x = x^*$) the above turns into

$$\left|\,\overline{\mathrm{cd}}(jK'/2 + jK'n' + x)\right|^2 = 1$$

or simply

$$\left|\,\overline{\mathrm{cd}}(jK'/2 + jK'n' + x)\right| = 1 \tag{9.85}$$

that is the absolute magnitude of $\overline{\mathrm{cd}}\, x$ is unity on the line $\mathrm{Im}\, x = jK'/2 + jK'n'$, exactly corresponding to the unit circle trajectory in Fig. 9.53. As another illustration, in Fig. 9.50 one could notice that $\overline{\mathrm{cd}}\, x$ is taking the values ± 1 and $\pm j$ at the intesections of vertical grid lines with the line $\mathrm{Im}\, x = jK'/2 + jK'n'$. Since we showed that the quasielliptic trajectories in Fig. 9.53 are all distinct, there are no other points within the imaginary quarter period where $|\,\overline{\mathrm{cd}}\, x| = 1$, and respectively the lines $\mathrm{Im}\, x = jK'/2 + jK'n$ are the only preimages of the unit circle.

Taking the complex argument of both parts of (9.84) we have

$$\arg\overline{\mathrm{cd}}(jK'/2 + jK'n' + x) - \arg\overline{\mathrm{cd}}(jK'/2 + jK'n' + x^*) = 0$$

or

$$\arg\overline{\mathrm{cd}}(jK'/2 + jK'n' + x) = \arg\overline{\mathrm{cd}}(jK'/2 + jK'n' + x^*) \tag{9.86}$$

That is the complex arguments of $\overline{\mathrm{cd}}\, x$ taken at the points symmetric relatively to the line $\mathrm{Im}\, x = jK'/2 + jK'n'$ are equal.

Fig. 9.55 provides an illustration for the range $0 \le \mathrm{Im}\, x \le K'$. Apparently on the ends of that range the elliptic cosine has purely real values (which can

be seen from the properties of $\mathrm{cd}\,x$ and from Fig. 9.50), corresponding to the complex argument being equal to 0 or π. Inside that range the value is becoming complex, where it is "maximally complex" (in the sense of $\arg \mathrm{cd}\,x$ having the maximal deviation from 0 or π) exactly in the middle, that is at $\mathrm{Im}\,x = K'/2$. This corresponds to the trajectories in Fig. 9.54 crossing the unit circle at right angles, so that the tangent lines of the trajectories taken at the intersection points are going through the origin, and therefore the angular deviation from the real line is attaining a maximum at these intersection points.

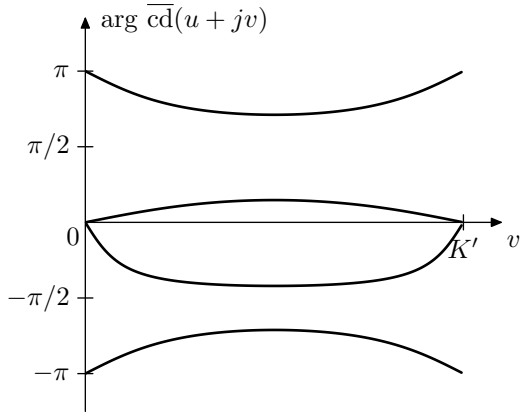


Figure 9.55: Deviation of Jacobian cosine's value from the real axis as a function of the imaginary part v of its argument, plotted for various real parts u and various elliptic moduli k.

In order to explain this maximum angular deviation at $\mathrm{Im}\,x = K'/2$ consider the following. The symmetry of the graphs in Fig. 9.55 is directly following from (9.86). Therefore there must be an extremum at the point in the middle of the range $\mathrm{Im}\,x \in [0, K']$. By (9.82b) this is the only extremum on that range and therefore this is the point of the maximum deviation.

Since the functions $\overline{\mathrm{sn}}$, $\overline{\mathrm{sc}}$ and $\overline{\mathrm{nd}}$ can be obtained from $\overline{\mathrm{cd}}$ by shifts and/or rotations of the complex plane (and a multiplication by j or by $-j$ for $\overline{\mathrm{sc}}$), they also exhibit similar symmetries. We won't go into detail of these. The functions $\overline{\mathrm{dc}}$ and $\overline{\mathrm{ns}}$ being the reciprocals of $\overline{\mathrm{cd}}$ and $\overline{\mathrm{sn}}$ are having similar symmetries as well.

Normalized argument

It will be also often convenient to use the following notation:

$$\begin{aligned} \mathrm{cd}_K\, x &= \mathrm{cd}\, Kx \\ \mathrm{sc}_{K'}\, x &= \mathrm{sc}\, K'x \\ \overline{\mathrm{cd}}_K\, x &= \overline{\mathrm{cd}}\, Kx \\ &\text{etc.} \end{aligned}$$

that is we write the quarter-period multiplier of the argument as a subscript of the function's name. In this notation e.g. the real quarter-period of $\mathrm{cd}_K\, x$ be-

comes equal to 1, therefore we will refer to this notation as *normalized-argument* Jacobian elliptic functions.

Note that we can't normalize the argument simultaneously for real and imaginary quarter periods, that is we need to choose between e.g. $\mathrm{cd}_K x$ and $\mathrm{cd}_{K'} x$, depending on our needs. Noticing that

$$\mathrm{cd}(Ku + jK'v) = \mathrm{cd}\, K\left(u + j\frac{K'}{K}v\right) = \mathrm{cd}_K\left(u + j\frac{K'}{K}v\right)$$
$$\mathrm{cd}(Ku + jK'v) = \mathrm{cd}\, K'\left(\frac{K}{K'}u + jv\right) = \mathrm{cd}_{K'}\left(\frac{K}{K'}u + jv\right)$$

we can see that the imaginary quarter period of $\mathrm{cd}_K x$ is K'/K and the real quarter period of $\mathrm{cd}_{K'} x$ is K/K'. The same obviously holds for other Jacobian elliptic functions.

Notice that Figs. 9.35 through 9.38 are effectively plotting sn_K, cd_K, sc_K and nd_K, since the argument scale is scaled by K.

In Figs. 9.35 through 9.38 one could notice that the values of sn_K, cd_K, sc_K and nd_K seem to be growing (in absolute magnitude) with k. Let's see if this is always the case.

In the beginning we are going to establish the fact that the argument-normalized amplitude $\varphi(x,k) = \mathrm{am}_K(x,k) = \mathrm{am}(K(k)x,k)$ grows with k on $x \in (0,1)$, that is

$$\frac{\partial\varphi}{\partial k} = \frac{\partial}{\partial k}\mathrm{am}_K(x,k) > 0 \qquad (0 < x < 1,\ 0 < k < 1) \tag{9.87}$$

Before analysing the partial derivative of $\mathrm{am}_K(x,k)$ with respect to k, we need to note the range in which $\mathrm{am}_K(x,k)$ is varying for $x \in (0,1)$:

$$\mathrm{am}_K(x,k) \in (0,\pi/2) \qquad \forall x \in (0,1) \quad (0 \le k < 1) \tag{9.88}$$

Indeed, by (9.55) $F(\varphi,k)$ is strictly increasing for $0 \le k < 1$, therefore, since $F(0,k) = 0$ and $F(\pi/2,k) = K(k)$, the range $\varphi \in (0,\pi/2)$ is mapped to $F(\varphi,k) \in (0,K)$ and vice versa. Therefore $\mathrm{am}(x,k)$ is monotonically changing from 0 to $\pi/2$ for x changing from 0 to K, and respectively $\mathrm{am}_K(x,k)$ is monotonically changing from 0 to $\pi/2$ for x changing from 0 to 1.

Now, given $\varphi(x,k) = \mathrm{am}_K(x,k)$, by (9.57)

$$F(\varphi,k) = K(k)x$$

Since we are interested in the partial derivative of φ with respect to k, we will consider x to be fixed and φ and k varying in the above equation. Then, taking the logarithm, we have

$$\ln F(\varphi,k) = \ln K(k)x$$

or

$$\ln F(\varphi,k) = \ln K(k) + \ln x$$

Let's take a full derivative in respect to k of both sides, where, since $x = \mathrm{const}$, the respective term fully disappears:

$$\frac{\mathrm{d}}{\mathrm{d}k}\ln F(\varphi,k) = \frac{\mathrm{d}}{\mathrm{d}k}\ln K(k)$$

$$\frac{\partial \ln F}{\partial \varphi}\frac{\mathrm{d}\varphi}{\mathrm{d}k}+\frac{\partial \ln F}{\partial k}=\frac{\mathrm{d}\ln K}{\mathrm{d}k}$$

Since $x=\text{const}$, we have $\partial\varphi/\partial k=\mathrm{d}\varphi/\mathrm{d}k$ and thus

$$\frac{\partial \ln F}{\partial \varphi}\frac{\partial\varphi}{\partial k}+\frac{\partial \ln F}{\partial k}=\frac{\mathrm{d}\ln K}{\mathrm{d}k}$$

Since by (9.55)

$$\frac{\partial \ln F}{\partial \varphi}=\frac{1}{F}\frac{\partial F}{\partial \varphi}=\frac{1}{F\sqrt{1-k^2\sin^2\varphi}}>0$$

it is suffcient to show that

$$\frac{\mathrm{d}\ln K}{\mathrm{d}k}>\frac{\partial \ln F}{\partial k}$$

and then $\partial\varphi/\partial k>0$ will automatically follow.

The previous inequality can be equivalently rewritten as

$$\frac{\mathrm{d}K}{\mathrm{d}k}>\frac{\partial F}{\partial k}$$

or, noticing that $K(k)=F(\pi/2,k)$ and reintroducing the explicit argument notation $F=F(\varphi,k)$,

$$\frac{\partial}{\partial k}F(\pi/2,k)>\frac{\partial}{\partial k}F(\varphi,k)$$

$$\frac{\partial}{\partial k}\big(F(\pi/2,k)-F(\varphi,k)\big)>0$$

$$\frac{\partial}{\partial k}\int_\varphi^{\pi/2}\frac{\mathrm{d}\theta}{\sqrt{1-k^2\sin^2\theta}}>0$$

$$\int_\varphi^{\pi/2}\left(\frac{\mathrm{d}}{\mathrm{d}k}\frac{1}{\sqrt{1-k^2\sin^2\theta}}\right)\mathrm{d}\theta>0$$

$$\int_\varphi^{\pi/2}\frac{k\sin^2\theta}{\big(1-k^2\sin^2\theta\big)^{3/2}}\,\mathrm{d}\theta>0 \tag{9.89}$$

Obviously the integral in (9.89) is positive for any $0<k<1$ and $0<\varphi<\pi/2$ and therefore (9.87) holds. It follows that

$$\mathrm{am}_K(x,k_1)<\mathrm{am}_K(x,k_2)\qquad\forall x\in(0,1)\quad(0\le k_1<k_2<1) \tag{9.90}$$

Notice that we have allowed $k_1=0$ in (9.90). Strictly speaking, at $k=0$ the integral in (9.89) turns to zero, respectively $\partial\varphi/\partial k=0$. However it doesn't matter much: since $\partial\varphi/\partial k>0$ starting with arbirarily small k, we have $\mathrm{am}_K(x,k)>\mathrm{am}_K(x,0)\ \forall x\in(0,1)$ and respectively (9.90) also holds for $k_1=0$.

Using (9.90), (9.88) and (9.58) we obtain

$$\mathrm{sn}_K(x,k_1)<\mathrm{sn}_K(x,k_2)\qquad\forall x\in(0,1)\quad(0\le k_1<k_2<1)$$

Using shift and symmetry properties of sn we can extend the above to the entire real axis (with the exception of purely integer points where $\operatorname{sn}_K x$ has the same values independently of k):

$$|\operatorname{sn}_K(x,k_1)| < |\operatorname{sn}_K(x,k_2)| \qquad \forall x \notin \mathbb{Z} \quad (0 \le k_1 < k_2 < 1,\ x \in \mathbb{R})$$

The same property for cd_K follows from the fact that cd_K can be obtained from sn_K by a quarter-period shift, and we have

$$|\operatorname{cd}_K(x,k_1)| < |\operatorname{cd}_K(x,k_2)| \qquad \forall x \notin \mathbb{Z} \quad (0 \le k_1 < k_2 < 1,\ x \in \mathbb{R}) \tag{9.91}$$

The same property for sc_K follows from (9.90), (9.88) and (9.60). The same property for nd_K follows from (9.90), (9.88) and (9.61).

Notably, the same property doesn't hold if the argument is not normalized by the real period K. Indeed, it is easily noticed that $F(\varphi,k)$ grows with both φ and k, therefore, given $F(\varphi,k) = \text{const}$, the value of φ will be decreasing for growing k, which means that

$$\frac{\partial}{\partial k} \operatorname{am}(x,k) < 0$$

9.11 Landen transformations

Given an elliptic modulus k and the associated quarter periods K and K' we could desire to find another elliptic modulus, such that the period ratio[13] K'/K is increased or decreased by an integer factor (compared to the original ratio K'/K). We will specifically focus on the transformation which changes the period ratio by a factor of 2. It will be particularly (but not only) useful as a means of evaluation of Jacobian elliptic functions and their inverses.

Given an elliptic modulus k_0 and the corresponding period ratio K_0'/K_0, let k_1 denote the elliptic modulus such that the corresponding period ratio is halved: $K_1'/K_1 = K_0'/2K_0$. It turns out that k_1 can be found by a simple formula: $k_1 = 2\sqrt{k_0}/(1+k_0)$. We define the *ascending Landen transformation*:

$$\mathcal{L}(k) = \frac{2\sqrt{k}}{1+k} \tag{9.92a}$$

It is easily verified that $\mathcal{L}(k) > k\ \forall k \in (0,1)$, which explains the name "ascending". Intuitively, an increase of the elliptic module k increases the real period K and reduces the imaginary period K', therefore the ratio K'/K is also reduced.

Inverting the ascending Landen transformation we obtain the *descending Landen transformation*:

$$\mathcal{L}^{-1}(k) = \frac{1-k'}{1+k'} = \left(\frac{k}{1+k'}\right)^2 \tag{9.92b}$$

where $k' = \sqrt{1+k^2}$ is the corresponding complementary modulus.[14] The readers are encouraged to check that $\mathcal{L}^{-1}(\mathcal{L}(k)) = \mathcal{L}(\mathcal{L}^{-1}(k)) = k$. Obviously, the descending Landen transformation doubles the period ratio K'/K.

[13] The ratio of the periods is of course formally speaking not K'/K but $4K'/4K$. However, obviously these values are equal.

[14] The second expression in (9.92b), in comparison to the first one, reduces the computation precision losses at small k.

It is easily found that the ascending and descending transformations are dual with respect to swapping k and k' (or, which is the same, K and K'):

$$\mathcal{L}'(k) = L^{-1}(k') \tag{9.93}$$

where $\mathcal{L}'(k) = \sqrt{1-(\mathcal{L}(k))^2}$ denotes the elliptic modulus complementary to $\mathcal{L}(k)$. Another property which follows from (9.92) is

$$(1+k)(1+\mathcal{L}'(k)) = 2 \tag{9.94a}$$

which also can be equivalently written as

$$(1+k')(1+\mathcal{L}^{-1}(k)) = 2 \tag{9.94b}$$

Landen sequences of elliptic moduli

Given some elliptic modulus k_0, Landen transformation establishes a bilateral sequence of elliptic moduli:

$$\ldots < k_{-2} < k_{-1} < k_0 < k_1 < k_2 < \ldots \tag{9.95a}$$

where $k_{n+1} = \mathcal{L}(k_n)$. Due to (9.93) this also automatically establishes a sequence of complementary moduli

$$\ldots > k'_{-2} > k'_{-1} > k'_0 > k'_1 > k'_2 > \ldots \tag{9.95b}$$

where $k'_{n+1} = \mathcal{L}^{-1}(k'_n)$. Note that by (9.94) we have

$$(1+k_n)(1+k'_{n+1}) = 2 \tag{9.96a}$$
$$(1+k'_n)(1+k_{n-1}) = 2 \tag{9.96b}$$

At small k (9.92b) turns to

$$\mathcal{L}^{-1}(k) \approx \frac{k^2}{4} \qquad (\text{for } k \approx 0) \tag{9.97}$$

Thus, as n grows, the moduli k_{-n} quickly decrease to zero. Conversely, k_n quickly grows to 1. E.g. starting at $k = 0.999$ we have a sequence

$$\begin{aligned} k_0 &= 0.999 \\ k_{-1} &\approx 0.914 \\ k_{-2} &\approx 0.424 \\ k_{-3} &\approx 0.0494 \\ k_{-4} &\approx 6 \cdot 10^{-4} \\ k_{-5} &\approx 1 \cdot 10^{-7} \end{aligned}$$

At this point the "trigonometric" elliptic functions become practically equal to their trigonometric counterparts (recall the property (9.62)), while the real quarter period becomes practically equal to $\pi/2$. Thus we almost exactly know the value of the real quarter period and we also can evaluate the respective trigonometric functions instead of an elliptic ones. Using the relationships that we are about to establish below, one can relate the elliptic function values at $k \approx 0$ to the values at larger k, which then provides a way to evaluate the elliptic functions for arbitrary k. Obviously, the same applies to the "hyperbolic" elliptic functions at $k \to 1$.

Ascending recursion for quarter period K

Landen transformation changes the real quarter period as

$$K(\mathcal{L}(k)) = (1+k)K(k) \tag{9.98}$$

(where $K(k)$ is the complete elliptic integral of the first kind).

Considering the sequence (9.95), let $K_n = K(k_n)$, $K'_n = K(k'_n)$ denote the real and imaginary quarter periods corresponding to moduli k_n. By (9.98)

$$K_{n+1} = (1+k_n)K_n \tag{9.99a}$$
$$K'_{n-1} = (1+k'_n)K'_n \tag{9.99b}$$

One can verify that (9.99) are in agreement with the fact that the period ratio is changed by a factor of 2:

$$\frac{K'_{n+1}}{K_{n+1}} = \frac{K'_n}{(1+k'_{n+1})\cdot(1+k_n)K_n} = \frac{K'_n}{2K_n}$$

where we have used (9.99) and (9.96).

The formula (9.99a) can be used as a means to compute $K(k)$ (for $0 < k < 1$). Notice that as k_n is getting small, the factors $(1+k_n)$ are becoming very close to zero. Therefore

$$K_{-n} = \frac{K_0}{\prod_{\nu=1}^{n}(1+k_{-\nu})}$$

by (9.56) should converge to $K_{-\infty} = K(0) = \pi/2$. In practical computations, starting from some n the factor $(1+k_{-n})$ will be indistinguishable from zero within the available computation precision, and so (by (9.97)) will be the subsequent factors. At this point the computations may be stopped and we can assume that $K_{-n} = \pi/2$ within the computation precision. Respectively

$$K_0 = K_{-n}\cdot\prod_{\nu=1}^{n}(1+k_{-\nu}) = \frac{\pi}{2}\cdot\prod_{\nu=1}^{n}(1+k_{-\nu}) \tag{9.100}$$

Thus we arrive at the following algorithm.

Given k_0 we wish to evaluate $K(k_0)$. Use descending Landen transformation to build a sequence of decreasing moduli k_0, k_{-1}, k_{-2}, ..., until at some step n the values k_{-n} becomes sufficiently small so that $1 + k_{-n} = 1$ within the available computation precision. Then ascend back to k_0 using (9.99a), thereby computing K_{-n+1}, K_{-n+2}, ..., K_{-1}, K_0.

Using (9.100) the same algorithm can be expressed iteratively rather than recursively:

```
// compute K from k
K := pi/2;
for i:=1 to 5 do
    k' := sqrt(1+k^2);
    k := (k/(1+k'))^2; // descending Landen transformation
    K := K*(1+k);
endfor;
```

Ascending recursion[15] **for** $\operatorname{sn} x$ **and** $\operatorname{cd} x$

Let's introduce the notation $\operatorname{sn}_n x = \operatorname{sn}_{K_n}(x, k_n) = \operatorname{sn}(K_n x, k_n)$, $\overline{\operatorname{sn}}_n x = \overline{\operatorname{sn}}_{K_n}(x, k_n) = \overline{\operatorname{sn}}(K_n x, k_n)$ etc. Note that thereby the imaginary period of sn_{n+1} is halved compared to sn_n. Let's also introduce the notation for the arithmetic average of x and its reciprocal $1/x$:

$$\mathcal{A}(x) = \frac{x + \frac{1}{x}}{2}$$

Then

$$\overline{\operatorname{sn}}_{n+1} x = \frac{1}{\sqrt{k_{n+1}}\mathcal{A}(\overline{\operatorname{sn}}_n x)} \tag{9.101a}$$

For the purposes of numeric evaluation it is usually more practical to rewrite (9.101a) in the form:

$$\operatorname{sn}_{n+1} x = \frac{(1 + k_n)\operatorname{sn}_n x}{1 + k_n \operatorname{sn}_n^2 x} \tag{9.101b}$$

which particularly avoids the division by zero if $\operatorname{sn}_n x = 0$.

Substituting $x + 1$ for x in (9.101) and using the shift property (9.66) we obtain

$$\overline{\operatorname{cd}}_{n+1} x = \frac{1}{\sqrt{k_{n+1}}\mathcal{A}\left(\overline{\operatorname{cd}}_n x\right)} \tag{9.102a}$$

or the version for numeric evaluation:

$$\operatorname{cd}_{n+1} x = \frac{(1 + k_n)\operatorname{cd}_n x}{1 + k_n \operatorname{cd}_n^2 x} \tag{9.102b}$$

The formulas (9.101), (9.102) can be used to compute the elliptic sine and cosine for $k \in (0, 1)$ by using a similar approach to how we used (9.99) to evaluate $K(k)$. Let's start with the elliptic cosine. The idea is that by (9.62)

$$\lim_{n\to+\infty} \operatorname{cd}_{-n} x = \lim_{n\to+\infty} \operatorname{cd}(K_{-n}x, k_{-n}) = \cos\frac{\pi}{2}x$$

Now notice that in (9.102b) we have $\operatorname{cd}_{n+1} x = \operatorname{cd}_n x$ within the available computation precision, provided

$$1 + k_n = 1 \tag{9.103a}$$

$$1 + k_n \operatorname{cd}_n^2 x = 1 \tag{9.103b}$$

within the same computation precision. Apparently, at this moment the sequence $\operatorname{cd}_n x$ (where $n \to -\infty$) converges to $\cos(\pi x/2)$.

The condition (9.103a) is the same that we had in the evaluation of $K(k)$. However additionally we have the requirement (9.103b) which is redundant if x is real (since then $0 \le \operatorname{cd}_n^2 x \le 1$), but becomes essential if $\operatorname{Im} x \ne 0$.

[15] This technique is commonly referred to as *descending Landen transformation*, since it expresses elliptic functions with higher values of the modulus k via elliptic functions with lower values of the modulus. However the recursion formula itself is applied to compute elliptic functions with higher k from elliptic functions with lower k, thus the recursion itself is ascending.

Since we don't know the value of $\operatorname{cd}_n x$ in advance, we can't directly estimate at which n (9.103b) begins to hold. Simply assuming that (9.103a) will suffice is not the best idea, since $\operatorname{cd}_n x$ can easily have values comparable to or exceeding $1/\sqrt{k_n}$ in absolute magnitude. Suppose however that

$$|\operatorname{Im} x| \le \frac{K'_n}{2K_n} \tag{9.104}$$

that is the imaginary part of the argument of cd_n doesn't exceed half of the imaginary quarter period.[16] From our previous discussion of the behavior of cd and $\overline{\operatorname{cd}}$ we should remember that $\overline{\operatorname{cd}}$ attains unit values in the middle of the imaginary quarter period and that its absolute magnitude grows away from the real axis (within the first imaginary quarter period). That is

$$|\overline{\operatorname{cd}}\, x| \le 1 \qquad \text{for } |\operatorname{Im} x| \le K'/2$$

Respectively

$$|\operatorname{cd} x| \le \frac{1}{\sqrt{k}} \qquad \text{for } |\operatorname{Im} x| \le K'/2$$

and

$$|\operatorname{cd}_n x| \le \frac{1}{\sqrt{k_n}} \qquad \text{for } |\operatorname{Im} x| \le \frac{K'_n}{2K_n} \tag{9.105}$$

Thus we have established that under the condition (9.104) the values of $\operatorname{cd}_n x$ do not exceed $1/\sqrt{k_n}$ in absolute magnitude. Apparently this is by far not good enough for (9.103b) to hold, since we only guarantee that $|k_n \operatorname{cd}_n^2 x| \le 1$, however the situation will improve if we decrease n by one or more steps.

First notice that if (9.104) holds at some n_0, then it will hold $\forall n \le n_0$ and so will (9.105), therefore $|k_n \operatorname{cd}_n^2 x| \le 1$ and $|1 + k_n \operatorname{cd}_n^2 x| \le 2$. Under further assumption of (9.103a), from (9.102b) we have

$$|\operatorname{cd}_n x| \le 2 \cdot |\operatorname{cd}_{n+1} x|$$

However by (9.97) we have $k_n = k_{n+1}^2/4$ and therefore

$$|k_n \operatorname{cd}_n^2 x| \le \frac{k_{n+1}^2}{4} \cdot 4 \cdot |\operatorname{cd}_{n+1}^2 x| = k_{n+1} \cdot |k_{n+1} \operatorname{cd}_{n+1}^2 x|$$

That is $k_n \operatorname{cd}_n^2 x$ will turn essentially to zero after just decreasing n by one step and respectively the sequence $\operatorname{cd}_n x$ (for $n \to -\infty$) will immediately converge.

In principle, we could now allow $|\operatorname{Im} x|$ to be arbitrarily large. As we decrease n step by step, the imaginary period K'_n/K_n of the function cd_n is doubling each time, thus sooner or later (9.104) will hold. However we don't want to do unnecessarily many iterations, not only for performance reasons, but also because the precision losses will accumulate. Therefore it might be more straightforward to simply wrap the argument of cd using the imaginary periodicity property.

Thus we arrive at the following algorithm. Suppose we want to evaluate $\operatorname{cd}(x, k)$. If $|\operatorname{Im} x| > K'$, we should use the periodicity property to get x into the range $|\operatorname{Im} x| \le K'$. Then introduce $u = x/K$ and $k_0 = k$, so that we

[16] Notice that we needed to divide by K_n in (9.104) because the notation cd_n includes the automatic multiplication of the argument by K_n.

have $\operatorname{cd}(x,k) = \operatorname{cd}_0 u$. Then we use the descending Landen transformation to decrease k_{-n} to almost zero,[17] where

$$\operatorname{cd}_{-n} u = \operatorname{cd} K_{-n} u \approx \operatorname{cd} \frac{\pi}{2} u \approx \cos \frac{\pi}{2} u$$

At this point we compute $\cos(\pi u/2)$ instead of $\operatorname{cd}_{-n} u$ and ascend back using (9.102b). In pseudocode this could be expressed as:

```
// compute cd(x/K,k), assuming |Im x|<=K'
function cdK(u,k,steps=5)
    if steps=0 then return cos(pi/2*u) endif;
    k' := sqrt(1+k^2);
    k := (k/(1+k'))^2; // descending Landen transformation
    y := cdK(u,k,steps-1);
    return ((1+k)*y)/(1+k*y^2);
endfunction;
```

Notice that u may be complex in the above, where we would need a cosine routine supporting a complex argument, which, if missing, could be implemented by (9.29a).

Evaluation of $\operatorname{sn} x$ is done in the same way, except that we have to compute $\sin(\pi u/2)$ as the approximation of $\operatorname{sn}_{-n} u$. The evaluation routines for sn and cd can be also reused for evaluation of sc and nd using (9.63).

Descending recursion for $\operatorname{sn} x$ and $\operatorname{cd} x$

We could invert the formulas (9.101) and (9.102) to express sn_{n-1} and cd_{n-1} in terms of respectively sn_n or cd_n:

$$\overline{\operatorname{sn}}_{n-1} x = \mathcal{A}^{-1}\left(\frac{1}{\sqrt{k_n}\,\overline{\operatorname{sn}}_n x}\right) \tag{9.106a}$$

or its "numerical" version, avoiding the divisions by zero for $\operatorname{sn}_n x = 0$

$$\operatorname{sn}_{n-1} x = \frac{1}{1+k_{n-1}} \cdot \frac{2\operatorname{sn}_n x}{1 \pm \sqrt{1 - k_n^2 \operatorname{sn}_n^2 x}} \tag{9.106b}$$

and

$$\overline{\operatorname{cd}}_{n-1} x = \mathcal{A}^{-1}\left(\frac{1}{\sqrt{k_n}\,\overline{\operatorname{cd}}_n x}\right) \tag{9.107a}$$

$$\operatorname{cd}_{n-1} x = \frac{1}{1+k_{n-1}} \cdot \frac{2\operatorname{cd}_n x}{1 \pm \sqrt{1 - k_n^2 \operatorname{cd}_n^2 x}} \tag{9.107b}$$

The ambiguity in formulas (9.106) and (9.107)[18] is apparently due to the fact that the imaginary periods of $\overline{\operatorname{sn}}_{n-1}$ and $\overline{\operatorname{cd}}_{n-1}$ are doubled compared to $\overline{\operatorname{sn}}_n$ and $\overline{\operatorname{sn}}_n$, therefore the formulas "do not know which of the two imaginary half-periods to choose".

[17] Note that thereby after one iteration we are guaratneed that $|\operatorname{Im} u| \le K'_n/2K_n$.

[18] Note that the inverse of $\mathcal{A}$ gives two different values.

The descending recursion can be used to evaluate the inverses of sn and cd. Given an equation of the form $\mathrm{cd}(x,k) = y$ (where we want to find $x = \mathrm{cd}^{-1}(y,k)$), we introduce $u = x/K$, $k_0 = k$ and $y_0 = y$, so that $y_0 = \mathrm{cd}_0\, u$. Then we use (9.107) to descend to $k_{-n} \approx 0$, where we have

$$y_{-n} = \mathrm{cd}_{-n}\, u \approx \cos\frac{\pi}{2}u$$

with high precision and therefore we can simply find u by $u = (2/\pi)\cos^{-1} y_{-n}$, thereby obtaining $\mathrm{cd}^{-1}(y,k) = x = K_0 u$.

Note that, even though (9.107) gives ambiguous results, any of those results will give a correct answer in the sense that we will get one of the possible solutions of $\mathrm{cd}(x,k) = y$ at the end of the recursion procedure. However, in order to avoid getting too far away from the origin (and in order to keep u within the real numbers range if x is a real number not exceeding 1 in magnitude), it is recommended to choose the value with the smaller absolute magnitude from the two values of $\mathcal{A}^{-1}$ in (9.107a), or, equivalently choose the "+" sign in the denominator of (9.107b). In case of complex values "choosing the + sign" also means that the complex square root operation should yield a value with a nonnegative real part, that is we should use the principal value (9.30).

Computing the inverse of sn is done in the same way using (9.106) (where it is preferable to choose the smaller-magnitude one from the two values of A^{-1} in (9.106a) and to use the "+" sign in the denominator of (9.106b)), finally computing u by $u = (2/\pi)\sin^{-1} y_{-n}$, thereby obtaining $\mathrm{sn}^{-1}(y,k) = x = K_0 u$. The respective pseudocode routine could be e.g.:

```
// compute x=sn^-1(y,k)
Kbypi2:=1; // accumulate ratio K/(pi/2)
for i:=1 to 5 do
    k' := sqrt(1+k^2);
    k_1 := (k/(1+k'))^2; // descending Landen transformation
    y := 2/(1+k_1) * y/(1+sqrt(1-k^2*y^2));
    k := k_1; Kbypi2 := Kbypi2*(1+k);
endfor;
x := Kbypi2*arcsin(y);
```

The routine for cd^{-1} is identical, except that it should use arccos instead of arcsin. Alternatively notice that cd^{-1} and sn^{-1} are related via shift and symmetry properties of cd and sn,, e.g. $\mathrm{cd}^{-1}\, x = K - \mathrm{sn}^{-1}\, x$, so that one function can be expressed in terms of the other. The functions sc^{-1} and nd^{-1} can be expressed via sn^{-1} and cd^{-1} using (9.63).

If the argument of sn^{-1} and cd^{-1} is restricted to real values, all respective computations will be real. Otherwise we need sqrt, arcsin and arccos functions to support complex argument, where sqrt must return the principal value (with nonnegative real part). These functions, if missing, can be implemented using (9.30) and (9.32).

We mentioned that the ambiguity of (9.106) and (9.107) is due to the doubling of the imaginary period of $\overline{\mathrm{sn}}$ and $\overline{\mathrm{cd}}$ on each step. Instead of that, we could have had the imaginary period fixed and the real period halved on each step, resulting in the same change of the period ratio. E.g. for the elliptic cosine,

introducing $\mathrm{cd}_{n'} = \mathrm{cd}(K'_n x, k_n)$, we have another relationship:

$$\mathrm{cd}_{n-1'}\, x = \frac{(1+k'_n)\,\mathrm{cd}^2_{n'}\, x - 1}{1-(1-k'_n)\,\mathrm{cd}^2_{n'}\, x} \tag{9.108}$$

(where $\mathrm{cd}_{n-1'}\, x = \mathrm{cd}(K'_{n-1} x, k_{n-1})$).

Unfortunately, while (9.108) avoids the ambiguity of (9.106) and (9.107), it is not useful for evaluation of the inverses of sn and cd, as there is another ambiguity popping up. Due to periodicity and symmetries of $\cos x$ along the real axis, we won't know which of the possible values of the inverse of $\cos x$ to take. When using (9.106) and (9.107) the real period of $\overline{\mathrm{sn}}$ and $\overline{\mathrm{cd}}$ was always exactly preserved by the transformation, therefore this ambiguity didn't matter as any of the values of $\cos^{-1}$ and $\sin^{-1}$ would do. If however the real period is not kept intact, the value returned by $\cos^{-1}$ might result in a wrong value after rescaling back to the original periods K_0 and K'_0.

One further issue is related to the preservation of the imaginary period. Particularly the range $y_{-n} \in [1, 1/k_{-n}]$ is mapped to the range $y_{-n-1} \in [1, 1/k_{-n-1}]$, respectively for a real y_{-n} above that range (that is $y_{-n} > 1/k_{-n}$) we obtain a real $y_{-n-1} > 1/k_{-n-1}$. Respectively $\cos^{-1}$ will return a purely imaginary result (while what we expect from cd^{-1} is clearly not purely imaginary, as one can see e.g. from Fig. 9.42) no matter how many times we apply the recursion (9.108) before evaluating the inverse cosine.

Ascending recursion for $\mathrm{nd}\, x$

Using the imaginary argument property (9.63) of the elliptic cosine and the Landen transformation's duality (9.93) we can convert the descending recursion formula (9.108) for $\mathrm{cd}\, x$ into an ascending recursion for $\mathrm{nd}\, x$, which takes the form

$$\mathrm{nd}_{n+1}\, x = \frac{(1+k_n)\,\mathrm{nd}^2_n\, x - 1}{1-(1-k_n)\,\mathrm{nd}^2_n\, x} \tag{9.109}$$

The main value of this recursion formula for us will be that we'll use it to derive another transformation.

Double Landen transformation

Consider two subsequent Landen transformation steps occurring from k_{n-1} to k_{n+1}. Inverting (9.108) we obtain

$$\mathrm{cd}^2_{n'}\, x = \frac{\mathrm{cd}_{n-1'}\, x + 1}{(1+k'_n) + (1-k'_n)\,\mathrm{cd}_{n-1'}\, x} = \frac{1}{1+k'_n} \cdot \frac{\mathrm{cd}_{n-1'}\, x + 1}{1 + k_{n-1}\,\mathrm{cd}_{n-1'}\, x}$$

Now we switch to the real period-based notation by substituting $K'_n x \leftarrow K_n x$. This is also equivalent to $K'_{n-1} x \leftarrow 2K_{n-1} x$ since $K'_{n-1}/K_{n-1} = 2K'_n/K_n$ and thus $K'_{n-1}/K'_n = 2K_{n-1}/K_n$. Therefore the substitution replaces $\mathrm{cd}_{n'}\, x$ with $\mathrm{cd}_n\, x$ and $\mathrm{cd}_{n-1'}\, x$ with $\mathrm{cd}_{n-1}\, 2x$, resulting in

$$\mathrm{cd}^2_n\, x = \frac{\mathrm{cd}_{n-1}\, 2x + 1}{(1+k'_n) + (1-k'_n)\,\mathrm{cd}_{n-1}\, 2x} = \frac{1}{1+k'_n} \cdot \frac{\mathrm{cd}_{n-1}\, 2x + 1}{1 + k_{n-1}\,\mathrm{cd}_{n-1}\, 2x}$$

Inverting (9.109) we obtain

$$\mathrm{nd}^2_n\, x = \frac{\mathrm{nd}_{n+1}\, x + 1}{(1+k_n) + (1-k_n)\,\mathrm{nd}_{n+1}\, x} = \frac{1}{1+k_n} \cdot \frac{\mathrm{nd}_{n+1}\, x + 1}{1 + k'_{n+1}\,\mathrm{nd}_{n+1}\, x}$$

By (9.69)

$$
\begin{aligned}
k_n^2 \operatorname{cd}_{n'}^2 x + k_n'^2 \operatorname{nd}_n^2 x = \\
&= \frac{k_n^2}{1+k_n'} \cdot \frac{\operatorname{cd}_{n-1} 2x + 1}{1 + k_{n-1} \operatorname{cd}_{n-1} 2x} + \frac{k_n'^2}{1+k_n} \cdot \frac{\operatorname{nd}_{n+1} x + 1}{1 + k_{n+1}' \operatorname{nd}_{n+1} x} = \\
&= (1+k_n')k_{n-1} \cdot \frac{\operatorname{cd}_{n-1} 2x + 1}{1 + k_{n-1} \operatorname{cd}_{n-1} 2x} + (1+k_n)k_{n+1}' \cdot \frac{\operatorname{nd}_{n+1} x + 1}{1 + k_{n+1}' \operatorname{nd}_{n+1} x} = \\
&= (1+k_n') \left(1 - \frac{1-k_{n-1}}{1 + k_{n-1} \operatorname{cd}_{n-1} 2x}\right) + \\
&\qquad + (1+k_n) \left(1 - \frac{1-k_{n+1}'}{1 + k_{n+1}' \operatorname{nd}_{n+1} x}\right) = \\
&= (1+k_n') \left(1 - \frac{2k_n'/(1+k_n')}{1 + k_{n-1} \operatorname{cd}_{n-1} 2x}\right) + \\
&\qquad + (1+k_n) \left(1 - \frac{2k_n/(1+k_n)}{1 + k_{n+1}' \operatorname{nd}_{n+1} x}\right) = \\
&= (1+k_n') - \frac{2k_n'}{1 + k_{n-1} \operatorname{cd}_{n-1} 2x} + (1+k_n) - \frac{2k_n}{1 + k_{n+1}' \operatorname{nd}_{n+1} x} = 1
\end{aligned}
$$

Solving for $k_{n+1}' \operatorname{nd}_{n+1} x$:

$$
\begin{aligned}
\frac{2k_n}{1 + k_{n+1}' \operatorname{nd}_{n+1} x} = 1 + k_n + k_n' - \frac{2k_n'}{1 + k_{n-1} \operatorname{cd}_{n-1} 2x} = \\
= \frac{(1+k_n+k_n')(1 + k_{n-1} \operatorname{cd}_{n-1} 2x) - 2k_n'}{1 + k_{n-1} \operatorname{cd}_{n-1} 2x}
\end{aligned}
$$

$$
1 + k_{n+1}' \operatorname{nd}_{n+1} x = \frac{2k_n(1 + k_{n-1} \operatorname{cd}_{n-1} 2x)}{(1+k_n+k_n')(1 + k_{n-1} \operatorname{cd}_{n-1} 2x) - 2k_n'}
$$

$$
\begin{aligned}
k_{n+1}' \operatorname{nd}_{n+1} x = \\
&= \frac{2k_n(1 + k_{n-1} \operatorname{cd}_{n-1} 2x) + 2k_n' - (1+k_n+k_n')(1 + k_{n-1} \operatorname{cd}_{n-1} 2x)}{(1+k_n+k_n')(1 + k_{n-1} \operatorname{cd}_{n-1} 2x) - 2k_n'} = \\
&= \frac{(k_n + k_n' - 1) - (1+k_n+k_n')k_{n-1} \operatorname{cd}_{n-1} 2x}{(1+k_n-k_n') + (1+k_n+k_n')k_{n-1} \operatorname{cd}_{n-1} 2x}
\end{aligned}
$$

By (9.92) and (9.93)

$$
\begin{aligned}
k_n &= \frac{2\sqrt{k_{n-1}}}{1+k_{n-1}} \\
k_n' &= \frac{1-k_{n-1}}{1+k_{n-1}} \\
k_{n+1}' = \frac{1-k_n}{1+k_n} = \frac{1+k_{n-1}-2\sqrt{k_{n-1}}}{1+k_{n-1}+2\sqrt{k_{n-1}}} &= \left(\frac{1-\sqrt{k_{n-1}}}{1+\sqrt{k_{n-1}}}\right)^2 = \\
&= \left(-\rho_{-1}\left(\sqrt{k_{n-1}}\right)\right)^2
\end{aligned}
$$

where we have noticed that the obtained expression can be conveniently written in terms of the Riemann sphere rotation ρ_{-1}. Therefore

$$\sqrt{k'_{n+1}} = -\rho_{-1}\left(\sqrt{k_{n-1}}\right) = \frac{1-\sqrt{k_{n-1}}}{1+\sqrt{k_{n-1}}} \tag{9.110}$$

Continuing the transformation of $k'_{n+1}\,\mathrm{nd}_{n+1}\,x$ we obtain

$$k'_{n+1}\,\mathrm{nd}_{n+1}\,x = \frac{(2\sqrt{k_{n-1}}-2k_{n-1})-(2-2\sqrt{k_{n-1}})k_{n-1}\,\mathrm{cd}_{n-1}\,2x}{(2k_{n-1}+2\sqrt{k_{n-1}})+(2+2\sqrt{k_{n-1}})k_{n-1}\,\mathrm{cd}_{n-1}\,2x} =$$
$$= \frac{1-\sqrt{k_{n-1}}}{1+\sqrt{k_{n-1}}}\cdot\frac{1-\sqrt{k_{n-1}}\,\mathrm{cd}_{n-1}\,2x}{1+\sqrt{k_{n-1}}\,\mathrm{cd}_{n-1}\,2x} = \sqrt{k'_{n+1}}\cdot\frac{1-\overline{\mathrm{cd}}_{n-1}\,2x}{1+\overline{\mathrm{cd}}_{n-1}\,2x}$$

and thus

$$\overline{\mathrm{nd}}_{n+1}\,x = \frac{1-\overline{\mathrm{cd}}_{n-1}\,2x}{1+\overline{\mathrm{cd}}_{n-1}\,2x} = -\rho_{-1}\left(\,\overline{\mathrm{cd}}_{n-1}\,2x\right)$$

or

$$\overline{\mathrm{nd}}_{n+1}\,\frac{x}{2} = \frac{1-\overline{\mathrm{cd}}_{n-1}\,x}{1+\overline{\mathrm{cd}}_{n-1}\,x} = -\rho_{-1}\left(\,\overline{\mathrm{cd}}_{n-1}\,x\right) \tag{9.111}$$

where the respective elliptic modulus is found from (9.110).

Notice that the halving of the argument in (9.111) is matched by the fact that the period ratio K'/K is changed by a factor of 4, That is $K'_{n-1}/K_{n-1} = 4K'_{n+1}/K_{n+1}$. At the same time the real and imaginary periods of $\overline{\mathrm{cd}}_{n-1}\,x$ are 4 and $2K'_{n-1}/K_{n-1}$, while the real and imaginary periods of $\overline{\mathrm{nd}}_{n+1}(x/2)$ are 4 and $8K'_{n+1}/K_{n+1} = 2K'_{n-1}/K_{n-1}$. Thus we have identically periodic functions in the left- and right-hand sides of (9.111).

9.12 Elliptic rational functions

Landen transformation was changing the period ratio by a factor of 2, which resulted in various elliptic functions after the transformation being expressed as a rational function of the same elliptic function prior to the transformation. There is a generalization of Landen transformation where the period ratio is changed by an arbitrary positive integer factor N. Such transformation is referred to as *N-th degree transformation* and the factor N is referred to as the *degree* of the transformation.

Suppose we are having an elliptic modulus k with respective quarter periods K' and K. Let $\tilde{k}$ be another elliptic modulus with respective quarter periods $\tilde{K}'$ and $\tilde{K}$, such that the quarter period ratio is increased N times:

$$\frac{\tilde{K}'}{\tilde{K}} = N\frac{K'}{K} \tag{9.112}$$

(the equation (9.112) is referred to as *degree equation*). Notice that since the period ratio is increased, the modulus is decreased: $\tilde{k} < k$.

Apparently k and $\tilde{k}$ are interdependent, where from (9.112) we obtain that increasing k decreases the ratios K'/K and $\tilde{K}'/\tilde{K}$, and thus increases $\tilde{k}$ as well. Thus $\tilde{k} = \tilde{k}(k)$ is an increasing function, where at N equal to a power of 2 we

obtain a $\log_2 N$ times repeated Landen transformation. The way to compute $\tilde{k}$ from a given k (and back) for arbitrary N will be discussed later.

In the transformation from k to $\tilde{k}$ we wish to obtain the relationship for cd, such that the imaginary period (in terms of normalized argument) is fixed. It turns out that such relationship always has the form:

$$\mathrm{cd}_{\tilde{K}'}\, u = R_N\big(\mathrm{cd}_{K'}\, u\big) \tag{9.113}$$

where $\mathrm{cd}_{K'}\, u = \mathrm{cd}(K'u, k)$, $\mathrm{cd}_{\tilde{K}'}\, u = \mathrm{cd}(\tilde{K}'u, \tilde{k})$ and $R_N(x)$ is some real rational function of order N. We already had a particular case of this formula for $N = 2$ in (9.108) where

$$R_2(x) = \frac{(1 + k_n')x^2 - 1}{1 - (1 - k_n')x^2}$$

The function $R_N(x)$ is referred to as *elliptic rational function of order N*. Notice that $R_N(x)$ depends on the elliptic modulus k, even though we don't explicitly notate it as function's parameter. Example graphs of $R_N(x)$ are given in Fig. 9.56.

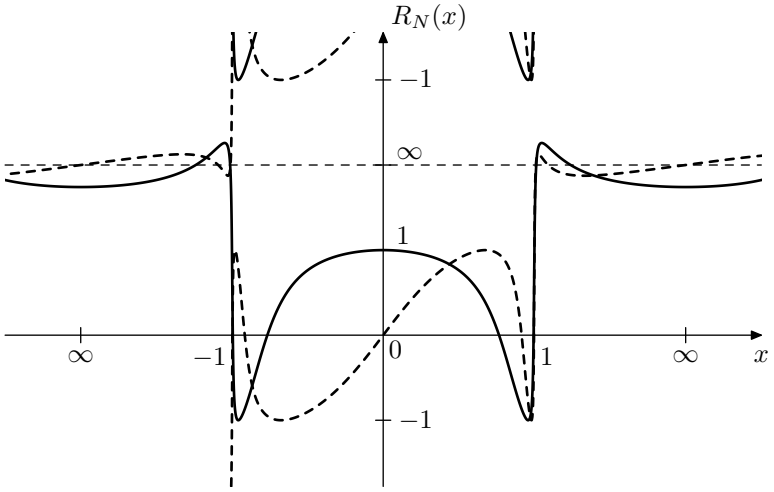


Figure 9.56: Elliptic rational functions of even (solid) and odd (dashed) orders for $k = 0.99$. The graphs do not cross the horizontal axis at $x = 1$, rather $R_N(1) = 1\ \forall N$, however the resolution of the figure is insufficient to see that. The poles of R_N are occurring at the intersections of the respective graph with the thin horizontal dashed line at ∞.

By using (9.112) we could rewrite (9.113) in terms of the real periods:

$$\mathrm{cd}_{\tilde{K}}\, Nu = R_N\big(\mathrm{cd}_K\, u\big) \tag{9.114}$$

where $\mathrm{cd}_K\, u = \mathrm{cd}(Ku, k)$, $\mathrm{cd}_{\tilde{K}}\, u = \mathrm{cd}(\tilde{K}u, \tilde{k})$. We could also rewrite (9.113) and (9.114) in a form without argument normalization, giving:

$$\mathrm{cd}(N\tilde{K}u, \tilde{k}) = R_N\big(\mathrm{cd}(Ku, k)\big) \tag{9.115}$$

Notice that any of the formulas (9.113), (9.114), (9.115) implies that $R_{N\cdot M}(x) = R_N(R_M(x)) = R_M(R_N(x))$ (with the properly chosen elliptic moduli for each of $R_{N\cdot M}$, R_N and R_M), as we are effectively simply chaining an N-th and an M-th degree transformations.

$R_N(x)$ as representation of linear scaling

In an obvious way, equation (9.113) can be expressed in terms of the preimage domain:

$$x = \mathrm{cd}_{K'} u \tag{9.116a}$$
$$R_N(x) = \mathrm{cd}_{\tilde{K}'} u \tag{9.116b}$$

Alternatively, (9.114) can be expressed as

$$x = \mathrm{cd}_K u \tag{9.117a}$$
$$v = Nu \tag{9.117b}$$
$$R_N(x) = \mathrm{cd}_{\tilde{K}} v \tag{9.117c}$$

Differently from x^N, $T_N(x)$ and $T_N^{-1}(x^{-1})$, this time there are two different mappings from the preimage to the representation domain in each case, corresponding to the two different moduli k and $\tilde{k}$. The linear scaling is explicitly present only in (9.117), however this is purely due to the implicit scaling contained in the period-normalized notation. The explicit notation form is the same for both (9.116) and (9.117) and contains the linear scaling:

$$x = \mathrm{cd}(u, k) \tag{9.118a}$$
$$v = N\frac{\tilde{K}}{K}u = \frac{\tilde{K}'}{K'}u \tag{9.118b}$$
$$R_N(x) = \mathrm{cd}(v, \tilde{k}) \tag{9.118c}$$

The mappings are however still different, since $k \neq \tilde{k}$.

By (9.112) the scaling (9.118b) exactly matches the imaginary periods and expands a single real period to exactly N real periods. For that reason the shifts of u by an integer number of real and/or imaginary periods do not affect the values of x and $R_N(x)$. By the even symmetry of cd a change of sign of u doesn't affect the values of x and $R_N(x)$ either. This however exhausts the set of possible preimages of a given x, since, as we know, $\mathrm{cd}\, x$ takes each value only once per quater-period grid cell (where the complex quadrants in Fig. 9.42 provide additional reference). Thus we can pick any preimage of x as the value of u and therefore can rewrite (9.116), (9.117) and (9.118) in their respective explicit forms:

$$R_N(x) = \mathrm{cd}_{\tilde{K}'}\left(\mathrm{cd}_{K'}^{-1} x\right) \tag{9.119a}$$
$$R_N(x) = \mathrm{cd}_{\tilde{K}}\left(N\,\mathrm{cd}_K^{-1} x\right) \tag{9.119b}$$
$$R_N(x) = \mathrm{cd}\left(N\frac{\tilde{K}}{K}\,\mathrm{cd}^{-1}(x,k),\ \tilde{k}\right) \tag{9.119c}$$

where cd^{-1} denote the inverse functions of the respective cd functions. Equation (9.119c) is the commonly known explicit expression for elliptic rational functions.

As with $T_N(x)$ and $\mathit{L}_N(x)$, an important class of preimages will be the horizontal lines in the complex planes u and v. From our discussion of cd and $\overline{\mathrm{cd}}$ we should recall that these lines produce distinct quasielliptic curves as their respective images, the full cycle of these curves corresponding to a single real period of u or v respectively (Figs. 9.52 and 9.53 serve as reference). Therefore x moving in such quasielliptic curve will be mapped to $R_N(x)$ moving in a similar curve, each cycle of x producing N cycles of $R_N(x)$.

Recall that with cosine-based preimages we were preferring the preimages in the lower complex semiplane, so that preimage movement towards the right was corresponding to counterclockwise rotation in the representation domain. Similarly, we are going to choose the elliptic cosine-based preimages within the imaginary quarter period strip located immediately below the real axis, as shown in Fig. 9.52, therefore preimage movement towards the right will correspond to counterclockwise rotation in the representation domain.

Given a preimage u located in the imaginary quarter period immediately below the real axis, the preimage v will also be located in the imaginary quarter period immediately below the real axis, since the imaginary quarter periods of u are mapped exactly onto the respective imaginary quarter periods of v. Also, apparently, u and v either both move simultaneously to the right or both to the left. Thus x and $R_N(x)$ move either both counterclockwise or both clockwise.[19]

Bands of $R_N(x)$

The four different parts of the principal preimage of the real line in Fig. 9.47 will correspond to the bands of elliptic filters which we are going to construct later. It is convenient to introduce the respective terminology at this point already.

In terms of (9.118) the principal preimage of the real axis $x \in \mathbb{R}$ is

$$\begin{aligned}
&u \in [0, 2K] && \Longleftrightarrow x \in [-1, 1] && \text{(a)}\\
&u \in [0, jK'] && \Longleftrightarrow x \in [1, 1/k] && \text{(b)}\\
&u \in [jK', jK' + 2K] && \Longleftrightarrow x \in [1/k, -1/k] && \text{(c)}\\
&u \in [2K, 2K + jK'] && \Longleftrightarrow x \in [-1/k, 1] && \text{(d)}
\end{aligned}$$

Respectively the principal preimage of the real axis $R_N(x) \in \mathbb{R}$ is:

$$\begin{aligned}
&v \in [0, 2\tilde{K}] && \Longleftrightarrow R_N(x) \in [-1, 1] && \text{(}\tilde{\text{a}}\text{)}\\
&v \in [0, j\tilde{K}'] && \Longleftrightarrow R_N(x) \in [1, 1/\tilde{k}] && \text{(}\tilde{\text{b}}\text{)}\\
&v \in [j\tilde{K}', j\tilde{K}' + 2\tilde{K}] && \Longleftrightarrow R_N(x) \in [1/\tilde{k}, -1/\tilde{k}] && \text{(}\tilde{\text{c}}\text{)}\\
&v \in [2\tilde{K}, 2\tilde{K} + j\tilde{K}'] && \Longleftrightarrow R_N(x) \in [-1/\tilde{k}, 1] && \text{(}\tilde{\text{d}}\text{)}
\end{aligned}$$

The linear scaling (9.118b) maps (b) to ($\tilde{\text{b}}$) one-to-one (imaginary period is preserved). The mapping from (a) to ($\tilde{\text{a}}$) and from (c) to ($\tilde{\text{c}}$) is one-to-N (real period

[19] Obviously, we could have chosen any other imaginary quarter period preimage strip. We have chosen the one right below the real axis simply to have a better defined reference in the preimage domain.

is multiplied by N). This is responsible for the appearance of the equiripples for $x \in [-1,1]$ and $x \in [1/k,-1/k]$ in Fig. 9.56. The mapping from (c) results either in some (non necessarily principal) preimage ($\tilde{\text{d}}$) if N is odd or in a non-principal preimage ($\tilde{\text{b}}$) if N is even.

Naming these four bands of $R_N(x)$ after the respective bands of the elliptic filters, we have:

Passband:	$x \in [-1,1]$	$\lvert R_N(x)\rvert \le 1$
Two transition bands:	$x \in [-1/k,-1] \cup [1,1/k]$	$1 \le \lvert R_N(x)\rvert \le 1/\tilde{k}$
Stopband:	$x \in [1/k,-1/k]$	$\lvert R_N(x)\rvert \ge 1/\tilde{k}$

The readers are advised to compare the above results to Fig. 9.56, identifying the equiripples of amplitudes 1 and $1/\tilde{k}$ in the pass- and stop-bands respectively. Since k is very close to 1, the transition bands are very narrow and aren't visible in Fig. 9.56, however at smaller k the stopband equiripples would become too small to be visible in the same figure.

The value $1/k$ is determining the width of the transition band(s) and is therefore referred to as the *selectivity factor*. The value $1/\tilde{k}$ determines the ratio of the equiripple amplitudes in the pass- and stop-bands and is referred to as the *discrimination factor*. Since k and $\tilde{k}$ increase or decrease simultaneously, so do $1/k$ and $1/\tilde{k}$. Therefore decreasing the transition band width (which is the same as decreasing the selectivity factor $1/k$) decreases the discrimination factor of $1/\tilde{k}$, thereby making the stop-band equiripples larger. Thus there is a tradeoff between the transition band width (which we, generally speaking, want to be small) and the discrimination factor (which we, generally speaking, want to be large). Fig. 9.57 illustrates.

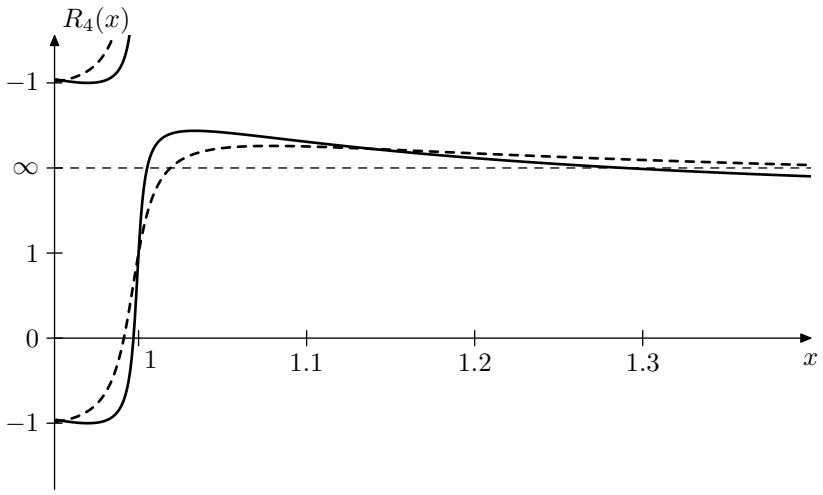


Figure 9.57: Transition region of $R_4(x)$ for $k = 0.998$ (solid) and $k = 0.99$ (dashed). The horizontal axis is linear, the vertical axis is using the arctangent scale.

Even/odd property

Since $\operatorname{cd}(u \pm 2K) = -\operatorname{cd} u$, a negation of x corresponds to a shift of its preimage u by $2K$. Respectively v is shifted by $2N\tilde{K}$, which will result in a negation of $R_N(x)$ if N is odd and will not change $R_N(x)$ is N is even. Therefore $R_N(x)$ is even/odd if N is even/odd:

$$R_N(-x) = (-1)^N R_N(x) \tag{9.120}$$

Values at special points

The principal preimage of $x = 1$ is $u = 0$. Therefore $v = 0$ and $R_N(x) = 1$. Therefore

$$R_N(1) = 1$$

By (9.120)

$$R_N(-1) = (-1)^N$$

The principal preimage of $x = 1/k$ is $u = jK'$. By (9.112) $v = j\tilde{K}'$ and $R_N(x) = 1/\tilde{k}$. Therefore

$$R_N(1/k) = 1/\tilde{k}$$

By (9.120)

$$R_N(-1/k) = (-1)^N/\tilde{k}$$

Since equiripples begin exactly at the boundaries of the respective bands of $R_N(x)$, the values at $x = \pm 1$ and $x = \pm 1/k$ also give the amplitudes of the equiripples of $R_N(x)$, which are thereby 1 in the passband and $1/\tilde{k}$ in the stopband.

The principal preimage of $x = 0$ is $u = K$. By (9.112) $v = N\tilde{K}$ and

$$R_N(0) = \begin{cases} 0 & \text{if } N \text{ is odd} \\ (-1)^{N/2} & \text{if } N \text{ is even} \end{cases}$$

The principal preimage of $x = \infty$ is $u = K + jK'$. By (9.112) $v = N\tilde{K} + jK'$. With the help of (9.65) we reuse the result for $R_N(0)$, obtaining

$$R_N(\infty) = \begin{cases} \infty & \text{if } N \text{ is odd} \\ (-1)^{N/2}/\tilde{k} & \text{if } N \text{ is even} \end{cases}$$

Two other interesting points are logarithmic midpoints of the transition band occurring at $x = \pm 1/\sqrt{k}$. The princial preimage of $x = 1/\sqrt{k}$ is the transition band's preimage midpoint $u = jK'/2$ and respectively $v = j\tilde{K}'/2$. Thus

$$R_N(1/\sqrt{k}) = 1/\sqrt{\tilde{k}}$$

That is the logarithmic midpoint of the transition band $[1, 1/k]$ is mapped to the logarithmic midpoint of the respective value range $[1, 1/\tilde{k}]$. By (9.120)

$$R_N(-1/\sqrt{k}) = (-1)^N/\sqrt{\tilde{k}}$$

Normalized elliptic rational functions

The graphs of $R_N(x)$ in Fig. 9.56 look somewhat asymmetric regarding the boundaries of the bands, which are at ± 1 and $\pm 1/k$ and the amplitudes of the ripples which are 1 and $1/\tilde{k}$ respectively. This can be addressed by switching to normalized elliptic cosine. The equations (9.113), (9.114) and (9.115) thereby respectively turn into

$$\overline{\mathrm{cd}}_{\tilde{K}'}\, u = \bar{R}_N\big(\overline{\mathrm{cd}}_{K'}\, u\big) \tag{9.121a}$$

$$\overline{\mathrm{cd}}_{\tilde{K}}\, Nu = \bar{R}_N\big(\overline{\mathrm{cd}}_{K}\, u\big) \tag{9.121b}$$

$$\overline{\mathrm{cd}}(N\tilde{K}u, \tilde{k}) = \bar{R}_N\big(\overline{\mathrm{cd}}(Ku, k)\big) \tag{9.121c}$$

while (9.119) turn into

$$\bar{R}_N(x) = \overline{\mathrm{cd}}_{\tilde{K}'}\left(\overline{\mathrm{cd}}_{K'}^{-1}\, x\right) \tag{9.121d}$$

$$\bar{R}_N(x) = \overline{\mathrm{cd}}_{\tilde{K}}\left(N\,\overline{\mathrm{cd}}_{K}^{-1}\, x\right) \tag{9.121e}$$

$$\bar{R}_N(x) = \overline{\mathrm{cd}}\left(N\frac{\tilde{K}}{K}\,\overline{\mathrm{cd}}^{\,-1}(x,k),\ \tilde{k}\right) \tag{9.121f}$$

where

$$\bar{R}_N(x) = \sqrt{\tilde{k}} R_N\left(\frac{x}{\sqrt{k}}\right) \tag{9.122}$$

is the *normalized* elliptic rational function. Note that $\bar{R}_N(x)$ is essentially simply a notational shortcut for the right-hand side of (9.122), however due to the more pure symmetries, it is often more convenient to work in terms of $\bar{R}_N(x)$ than $R_N(x)$.

The bands of $\bar{R}_N(x)$ are threrefore

Passband:	$\lvert x\rvert \le \sqrt{k}$	$\lvert R_N(x)\rvert \le \sqrt{\tilde{k}}$
Transition bands:	$\sqrt{k} \le \lvert x\rvert \le 1/\sqrt{k}$	$\sqrt{\tilde{k}} \le \lvert R_N(x)\rvert \le 1/\sqrt{\tilde{k}}$
Stopband:	$\lvert x\rvert \ge 1/\sqrt{k}$	$\lvert R_N(x)\rvert \ge 1/\sqrt{\tilde{k}}$

while the special points of $\bar{R}_N(x)$ respectively are:

$$\begin{aligned}
\bar{R}_N(\sqrt{k}) &= \sqrt{\tilde{k}} \\
\bar{R}_N(-\sqrt{k}) &= (-1)^N\sqrt{\tilde{k}} \\
\bar{R}_N(1/\sqrt{k}) &= 1/\sqrt{\tilde{k}} \\
\bar{R}_N(-1/\sqrt{k}) &= (-1)^N/\sqrt{\tilde{k}} \\
\bar{R}_N(0) &= \begin{cases} 0 & \text{if } N \text{ is odd} \\ (-1)^N\sqrt{\tilde{k}} & \text{if } N \text{ is even} \end{cases} \\
\bar{R}_N(\infty) &= \begin{cases} \infty & \text{if } N \text{ is odd} \\ (-1)^N/\sqrt{\tilde{k}} & \text{if } N \text{ is even} \end{cases} \\
\bar{R}_N(1) &= 1
\end{aligned}$$

$$\bar{R}_N(-1) = (-1)^N$$

The graphs of $\bar{R}_N(x)$ are plotted in Fig. 9.58.

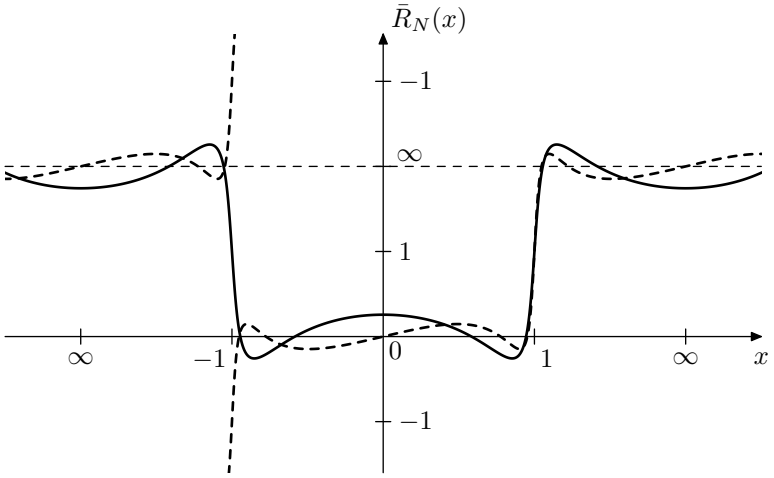


Figure 9.58: Normalized elliptic rational functions of even (solid) and odd (dashed) orders for $k = 0.9$.

The interpretation of $\bar{R}_N(x)$ as a linear scaling representation is essentially the same as in (9.116), (9.117) and (9.118) respectively, except that cd must be replaced with $\overline{\text{cd}}$. E.g. (9.118) becomes

$$x = \overline{\text{cd}}(u, k) \tag{9.123a}$$

$$v = N\frac{\tilde{K}}{K}u = \frac{\tilde{K}'}{K'}u \tag{9.123b}$$

$$\bar{R}_N(x) = \overline{\text{cd}}(v, \tilde{k}) \tag{9.123c}$$

Symmetries of $\bar{R}_N(x)$

In Fig. 9.58 one can notice that the graphs of $\bar{R}_N(x)$ have certain symmetries. One symmetry is relatively to the origin:

$$\bar{R}_N(-x) = (-1)^N \bar{R}_N(x) \tag{9.124}$$

(which is simultaneously the symmetry relative to $x = \infty$). Apparently this is simply the even/odd property of $R_N(x)$ which is preserved in the $\bar{R}_N(x)$ form.

The other symmetry is relatively to the point $\bar{R}_N(1) = 1$:

$$\bar{R}_N(1/x) = 1/\bar{R}_N(x) \tag{9.125}$$

with a similar symmetry around the point at $x = -1$ which follows from the first two symmetries. The proof of (9.125) follows from (9.75d). Given x and its preimage u, the preimage of $1/x$ is $jK' - u$. Similarly, the preimage of $1/\bar{R}_N(x)$ is $j\tilde{K}' - v$, however simultaneously

$$j\tilde{K}' - v = \frac{\tilde{K}'}{K'}\,(jK' - u)$$

therefore $j\tilde{K}' - v$ is also the preimage of $\bar{R}_N(1/x)$ and $\bar{R}_N(1/x) = 1/\bar{R}_N(x)$.

In terms of $R_N(x)$ the symmetry (9.125) takes the form

$$R_N(1/kx) = 1\Big/\tilde{k}R_N(x) \tag{9.126}$$

Similarly to $R_N(x)$, the normalized elliptic rational function $\bar{R}_N(x)$ maps the quasielliptic curves Fig. 9.53 to other quasielliptic curves from the same family. By Fig. 9.53 and (9.85) the unit circle will be mapped to the unit circle, since the preimage line $\operatorname{Im} u = jK'/2 + jK'n'$ will be mapped to the preimage line $\operatorname{Im} v = j\tilde{K}'/2 + j\tilde{K}'n'$. There won't be other preimages of the unit circle, since all trajectories in Fig. 9.53 are distinct and occur for different imaginary parts of the preimage. Thus

$$|x| = 1 \iff |\bar{R}_N(x)| = 1 \tag{9.127}$$

Poles and zeros of $R_N(x)$

Letting $R_N(x) = 0$ and using the representation form (9.117) we obtain

$$v = 2n + 1 = 2\left(\frac{1}{2} + n\right)$$

(where the second form is given for comparison with the respective derivations of the zeros of x^N and $T_N(x)$). Respectively

$$u = \frac{2n+1}{N} = 2\frac{\frac{1}{2}+n}{N}$$

and

$$x = \operatorname{cd}_K \frac{2n+1}{N} = \operatorname{cd}_K\left(2\frac{\frac{1}{2}+n}{N}\right)$$

which means that the zeros of $R_N(x)$ are

$$z_n = \operatorname{cd}_K \frac{2n+1}{N} = \operatorname{cd}_K\left(2\frac{\frac{1}{2}+n}{N}\right) \tag{9.128}$$

where there are N distinct values corresponding to $0 < u < 2$. Notice that the zeros are all real and lie within $(-1, 1)$. Also notice that $z_n = -z_{N-1-n}$, therefore the zeros are positioned symmetrically around the origin. Consequently, if N is odd, one of z_n will be at the origin.

By (9.126) the poles can be obtained from zeros as

$$p_n = 1/kz_n$$

Note that if N is odd, then so is $R_N(x)$ and one of the zeros will be at $x = 0$. In this case one of the poles will be at the infinity and there will be only $N - 1$ finite poles.

Given z_n and taking into account that $p_n = 1/kz_n$, we can write $R_N(x)$ in the form

$$R_N(x) = g \cdot \frac{\prod(x - z_n)}{\prod\limits_{z_n \neq 0}(x - 1/z_n)}$$

which we could also write as

$$R_N(x) = g \cdot x^{N\wedge 1} \cdot \prod_{z_n \neq 0} \frac{x - z_n}{x - 1/kz_n} = g \cdot x^{N\wedge 1} \cdot \prod_{z_n > 0} \frac{x^2 - z_n^2}{x^2 - 1/k^2 z_n^2}$$

The value of the gain coefficient g can be obtained from the fact that $R_N(x)$ must satisfy $R_N(1) = 1$. Alternatively we can satisfy $R_N(1) = 1$ by simply forcing each of the factors of $R_N(x)$ to be equal to unity at $x = 1$ (where prior to the factor normalization we also have multiplied each of the factors by $-k^2 z_n^2$)

$$R_N(x) = x^{N\wedge 1} \cdot \prod_{z_n > 0} \left(\frac{1 - k^2 z_n^2}{1 - z_n^2} \cdot \frac{x^2 - z_n^2}{1 - k^2 z_n^2 x^2} \right) \tag{9.129}$$

Poles and zeros of $\bar{R}_N(x)$

By (9.122) the zeros of $\bar{R}_N(x)$ can be obtained from the zeros of $R_N(x)$ as

$$\bar{z}_n = \sqrt{k} z_n = \overline{\mathrm{cd}}_K \frac{2n+1}{N} = \overline{\mathrm{cd}}_K \left(2\frac{\frac{1}{2} + n}{N} \right) \tag{9.130}$$

Notice that thereby $\bar{z}_n \in (-\sqrt{k}, \sqrt{k})$. By (9.125) the poles are related to the zeros as

$$\bar{p}_n \bar{z}_n = 1$$

The factored form (9.129) respectively becomes

$$\bar{R}_N(x) = x^{N\wedge 1} \cdot \prod_{\bar{z}_n > 0} \frac{x^2 - \bar{z}_n^2}{1 - \bar{z}_n^2 x^2} \tag{9.131}$$

where we don't have explicit normalization factors anymore, since the factors under the product sign are already all equal to unity at $x = 1$, thereby giving $\bar{R}_N(1) = 1$ as the transition band's midpoint (which is exactly what it should be according to the previously discussed special point values of $\bar{R}_N(x)$).

By obtaining equations (9.129) and (9.131) we have constructed $R_N(x)$ and $\bar{R}_N$ in the explicit rational function form.

Relationship between k and $\tilde{k}$

Note that the explicit rational function forms (9.129) and (9.131) were obtained without using the yet unknown to us $\tilde{k}$ (or $\tilde{K}$ or $\tilde{K}'$). On the other hand, having constructed $R_N(x)$ and/or $\bar{R}_N(x)$, we can obtain $\tilde{k}$ from the condition $R_N(1/k) = 1/\tilde{k}$ or $\bar{R}_N(\sqrt{k}) = \sqrt{\tilde{k}}$.

We can also obtain an explicit expression for $\tilde{k}$ in terms of k. Substituting (9.129) into $R_N(1/k) = 1/\tilde{k}$ we obtain

$$1/\tilde{k} = k^{-(N\wedge 1)} \cdot \prod_{z_n > 0} \left(\frac{1 - k^2 z_n^2}{1 - z_n^2} \cdot \frac{1/k^2 - z_n^2}{1 - z_n^2} \right) =$$

$$= k^{-N} \cdot \prod_{z_n > 0} \left(\frac{1 - k^2 z_n^2}{1 - z_n^2} \cdot \frac{1 - k^2 z_n^2}{1 - z_n^2} \right) = k^{-N} \cdot \prod_{z_n > 0} \left(\frac{1 - k^2 z_n^2}{1 - z_n^2} \right)^2$$

By (9.128) $z_n = \mathrm{cd}_K\, u_n$ where

$$u_n = \frac{2n+1}{N}$$

Therefore

$$\frac{1-k^2 z_n^2}{1-z_n^2} = \frac{1-k^2\,\mathrm{cd}_K^2\, u_n}{1-\mathrm{cd}_K^2\, u_n} = \frac{1}{\mathrm{sn}_K^2\, u_n}$$

where the latter transformation is by (9.68). Noticing that

$$z_n > 0 \iff 0 < u_n < 1$$

we obtain

$$\tilde{k} = k^N \cdot \prod_{0<u_n<1} \mathrm{sn}_K^4\, u_n \tag{9.132}$$

(where the number of factors under the product sign is equal to the integer part of $N/2$) which formally gives an explicit expression for $\tilde{k}$. However practically this expression is exactly the same as $\tilde{k} = 1/R_N(1/k)$ and thus we can simply find $\tilde{k}$ (and respectively $\tilde{K}$) from the latter condition.

Finding k from $\tilde{k}$ can be done by using the duality of the N-th degree transformation in respect to k and k' (which is pretty much the same as the respective duality of the Landen transformation). Let $\tilde{k} = \mathcal{N}(k)$ denote the N-th degree transformation of k defined by (9.132) (or by explicit usage of $R_N(1/k) = 1/\tilde{k}$ or $\bar{R}_N(\sqrt{k}) = \sqrt{\tilde{k}}$). If the ratio K'/K is decreased N times by the N-th degree transformation from k to $\tilde{k}$, then the ratio K/K' is increased N times, which means we are performing an N^{-1}-th degree transformation from k' to $\tilde{k}'$, that is $\tilde{k}' = \mathcal{N}^{-1}(k')$. Conversely, $\tilde{k}'$ and k' are related via an N-th degree transformation: $k' = \mathcal{N}(\tilde{k}')$, which by (9.132) means

$$k' = \tilde{k}'^N \cdot \prod_{0<u_n<1} \mathrm{sn}_{\tilde{K}'}^4\, u_n \tag{9.133}$$

Renormalized elliptic rational functions

Similarly to renormalized Chebyshev polynomials we introduce renormalized elliptic rational functions where we will renormalize only $\bar{R}_N(x)$ (although we could take similar steps to renormalize $R_N(x)$ as well):

$$\bar{R}_N(x,\lambda) = \frac{\bar{R}_N(x/\lambda)}{\bar{R}_N(1/\lambda)} \tag{9.134}$$

As with renormalized Chebyshev polynomials, the parameter λ affects the bands of the elliptic rational functions and the equiripple amplitudes. Apparently the passband of $\bar{R}_N(x)$ is $|x| \le \lambda\sqrt{k}$, while the stopband is $|x| \ge \lambda/\sqrt{k}$. Thus if λ becomes smaller, the passband shrinks while the stopband simultaneously expands and vice versa. The equiripple amplitudes in the pass and stop-bands are becoming equal to $\sqrt{\tilde{k}}/\bar{R}_N(1/\lambda)$ and $1/\sqrt{\tilde{k}}\bar{R}_N(1/\lambda)$ respectively. Therefore both values increase if λ grows and decrease if λ becomes smaller, which means that the equiripple sizes in the pass- and stop-bands are traded against each

other[20] (Fig. 9.59). Notice that thereby a smaller bandiwith of the pass- or stop-band corresponds to a smaller equiripple amplitude in the same band and vice versa.

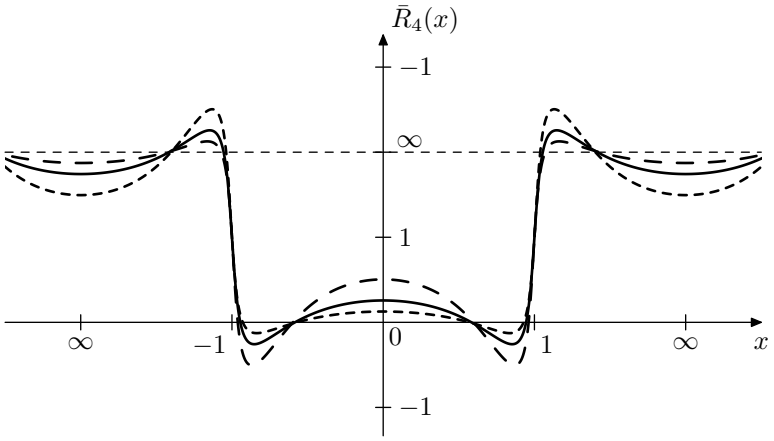


Figure 9.59: Renormalized elliptic rational function for $\lambda = 1$ (solid), $\lambda = k^{1/4}$ (small dashes) and $\lambda = k^{-1/4}$ (large dashes). As the used elliptic modulus $k = 0.9$ is pretty close to 1, the corresponding variations of the transition band width are not well visible.

The reasonable range of λ is equal to the transition band $[\sqrt{k}, 1/\sqrt{k}]$. For λ within this range the equiripples (both in the pass- and stopbands) do not grow further than to unit amplitude. As a somewhat excessive range of λ we could take $(\max\{\bar{z}_n\}, 1/\max\{\bar{z}_n\})$, limiting λ between the zeros and poles of $\bar{R}_N(x)$. In this case the equiripples may become arbitrarily large.

As with renormalized Chebyshev polynomials, we may omit the parameter λ, understanding it implicitly, and simply write $\hat{\bar{R}}_N(x)$.

Relation to x^N, $T_N(x)$ and $L_N(x)$

In (9.131) it is easily noticed that at $\bar{z}_n \to 0$ the right-hand side turns into x^N. On the other hand $|\bar{z}_n| \le \sqrt{k}$, therefore, if $k \to 0$, then $\bar{z}_n \to 0$ and respectively by (9.131)

$$\lim_{k\to 0} \bar{R}_N(x) = x^N \tag{9.135}$$

or simply $\bar{R}_N(x) = x^N$ for $k = 0$ (Fig. 9.60).

At $k \to 0$ we have $\mathrm{cd}\, x \to \cos x$ and $\mathrm{cd}_K x \to \cos \pi x/2$, therefore (9.114) turns into

$$\cos \frac{\pi}{2} N u = R_N \left(\cos \frac{\pi}{2} u \right)$$

By replacing $\pi u/2$ with u it can be equivalently written as

$$\cos Nu = R_N\big(\cos u\big)$$

[20] By the earlier given definition of the amplitude of oscillations around infinity, the size of the stop-band equiripples is smaller if the formal amplitude of the equiripples is larger.

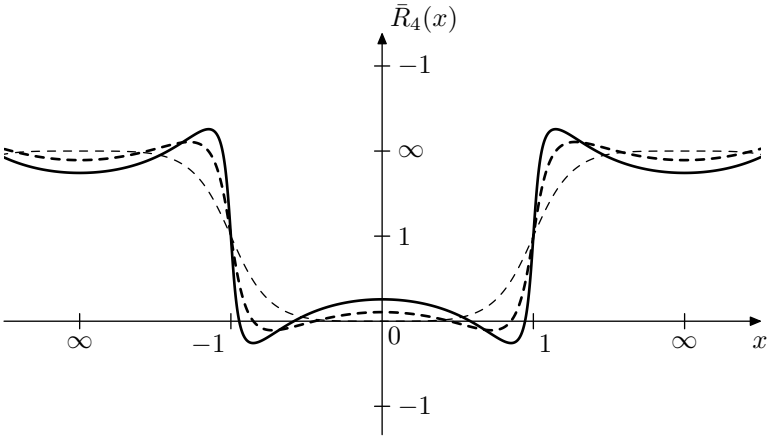


Figure 9.60: Normalized elliptic rational function for $k = 0.9$ (solid), $k = 0.7$ (dashed) and $k = 0$ (thin dashed).

which is identical to (9.35). Therefore $R_N(x)$ becomes identical to T_N and thus we have

$$\lim_{k\to 0} R_N(x) = T_N(x) \tag{9.136}$$

or simply $R_N(x) = T_N(x)$ for $k = 0$. Notice that as $K' \to \infty$ and $1/k \to \infty$, the transition band of $R_N(x)$ becomes infinitely large in both preimage and representation domains, while the stopband $|x| \geq 1/k$ disappears into infinity.

Using (9.122) and (9.134) we can express the approaching to $T_N(x)$ in terms of $\bar{R}_N(x)$ and $\hat{\bar{R}}_N(x)$:

$$\lim_{k\to 0} \frac{\bar{R}_N\left(x\sqrt{k}\right)}{\sqrt{\tilde{k}}} = \lim_{k\to 0} \hat{\bar{R}}_N\left(x, 1/\sqrt{k}\right) = T_N(x) \tag{9.137}$$

where we had to take the limit to avoid divisions and multiplications by zero. By the definition of $\mathcal{L}_N(x)$ equation (9.137) can be rewritten as

$$\lim_{k\to 0} \frac{\bar{R}_N\left(x\sqrt{k}\right)}{\sqrt{\tilde{k}}} = \frac{1}{\mathcal{L}_N(1/x)}$$

Substituting $1/x$ for x and reciprocating both sides

$$\lim_{k\to 0} \frac{\sqrt{\tilde{k}}}{\bar{R}_N\left(\sqrt{k}/x\right)} = \mathcal{L}_N(1/x)$$

By (9.125) and (9.134)

$$\lim_{k\to 0} \sqrt{\tilde{k}}\bar{R}_N\left(x/\sqrt{k}\right) = \lim_{k\to 0} \hat{\bar{R}}_N\left(x, \sqrt{k}\right) = \mathcal{L}_N(1/x) \tag{9.138}$$

Transition band slope of $\bar{R}_N(x)$

By (9.135) $\bar{R}_N(x)$ turns into x^N at $k = 0$. On the other hand at the transition band's midpoint $x = 1$ we have $\bar{R}_N(x) = x^N = 1\ \forall k$. It would be therefore

interesting to compare the slopes of $\bar{R}_N(x)$ and x^N within the transition band. In order to do that we are going to take similar steps to what we did in the comparison of $T_N(x)$ and x^N. Comparing (9.131) to (9.40) we compare their individual factors by computing their respective differences:

$$\frac{x^2-\bar{z}_n^2}{1-\bar{z}_n^2x^2}-x^2=\frac{x^2-\bar{z}_n^2-x^2+\bar{z}_n^2x^4}{1-\bar{z}_n^2x^2}=\frac{\bar{z}_n^2(x^4-1)}{1-\bar{z}_n^2x^2} \tag{9.139}$$

Assuming $k>0$, we have $0<\bar{z}_n<\sqrt{k}<1$ in the above. Thus in the range $1<|x|<1/\max\{\bar{z}_n\}$ the differences (9.139) are strictly positive and the factors of (9.131) are larger than those of (9.40). Since for $1<x<1/\max\{\bar{z}_n\}$ all factors of (9.131) and (9.40) are positive, we have

$$\bar{R}_N(x)>x^N \qquad (1<x<1/\max\{\bar{z}_n\},\ N>1)$$

By the even/odd symmetries of $\bar{R}_N(x)$ and x^N:

$$|\bar{R}_N(x)|>|x^N| \qquad (1<|x|<1/\max\{\bar{z}_n\},\ N>1)$$

By the symmetry (9.125) and by the same symmetry of x^N

$$|\bar{R}_N(x)|<|x^N| \qquad (\max\{\bar{z}_n\}<|x|<1,\ N>1)$$

Note that thereby our discussion has completely covered the band $|x|\in(\max\{\bar{z}_n\},1/\max\{\bar{z}_n\})$ which also includes the entire transition band $|x|\in[\sqrt{k},1/\sqrt{k}]$. From (9.139) we can also notice that the difference grows in magnitude as $\bar{z}_n$ grow in magnitude. However by (9.130) and (9.91) the absolute magnitudes of $\bar{z}_n$ should simply grow with k. Thus the differences (9.139) grow with k and respectively $\bar{R}_N(x)$ deviates stronger for x^N within the transition band as k grows (which can be seen in Fig. 9.59).

Transition band slope of $R_N(x)$

By (9.136) the elliptic rational function $R_N(x)$ turns into a Chebyshev polynomial of the same order N. We would be now in the position to compare their slopes within the transition band of $R_N(x)$.

Comparing the factors of (9.129) and (9.42) (where in (9.42) we let $\lambda=1$ to turn $\hat{T}_N(x)$ into $T_N(x)$) we first pretend that the zeros of $R_N(x)$ and $T_N(x)$ are identical. In that case the difference of the respective factors is

$$\begin{aligned}&\frac{1-k^2z_n^2}{1-z_n^2}\cdot\frac{x^2-z_n^2}{1-k^2z_n^2x^2}-\frac{x^2-z_n^2}{1-z_n^2}=\frac{x^2-z_n^2}{1-z_n^2}\cdot\left(\frac{1-k^2z_n^2}{1-k^2z_n^2x^2}-1\right)=\\&=\frac{x^2-z_n^2}{1-z_n^2}\cdot\frac{1-k^2z_n^2-1+k^2z_n^2x^2}{1-k^2z_n^2x^2}=\frac{x^2-z_n^2}{1-z_n^2}\cdot\frac{k^2z_n^2(x^2-1)}{1-k^2z_n^2x^2}\end{aligned} \tag{9.140}$$

Since within the transition band $[1,1/k]$ of $R_N(x)$ we have $1-k^2z_n^2x^2>0$ and $x^2-z_n^2>0$, the difference (9.140) is positive for $x\in(1,1/k]$ (for $0<k<1$).[21]

By (9.128) and (9.91) the zeros z_n grow with k. On the other hand, the zeros z_n of $R_N(x)$ become equal to Chebyshev polynomial's zeros at $k=0$.

[21] Actually the same holds a bit further than within the tranistion band, namely within $[1,1/k\max\{z_n\}]$ but for simplicity we'll talk of transition band.

Thus the zeros of $T_N(x)$ are smaller in absolute magnitude than those of R_N. Temporarily notating Chebyshev polynomial's zeros as $\lambda_n z_n$ where $0 < \lambda_n < 1$ (assuming $k > 0$), we notice that

$$\frac{x^2 - z_n^2}{1 - z_n^2} - \frac{x^2 - (\lambda_n z_n)^2}{1 - (\lambda_n z_n)^2} = \left(\frac{x^2 - z_n^2}{1 - z_n^2} - x^2\right) - \left(\frac{x^2 - (\lambda_n z_n)^2}{1 - (\lambda_n z_n)^2} - x^2\right) = \\ = \frac{z_n^2(x^2 - 1)}{1 - z_n^2} - \frac{(\lambda_n z_n)^2(x^2 - 1)}{1 - (\lambda_n z_n)^2} \tag{9.141}$$

where we have used (9.43). That is the difference (9.141) is again positive for $x > 1$ and it is growing with $\lambda_n \to 0$ (which corresponds to the growing k). Adding (9.141) and (9.140) we obtain

$$\frac{1 - k^2 z_n^2}{1 - z_n^2} \cdot \frac{x^2 - z_n^2}{1 - k^2 z_n^2 x^2} - \frac{x^2 - \lambda_n z_n^2}{1 - \lambda_n z_n^2} = \\ = \frac{x^2 - z_n^2}{1 - z_n^2} \cdot \frac{k^2 z_n^2 (x^2 - 1)}{1 - k^2 z_n^2 x^2} + \left(\frac{z_n^2(x^2 - 1)}{1 - z_n^2} - \frac{(\lambda_n z_n)^2(x^2 - 1)}{1 - (\lambda_n z_n)^2}\right) > 0 \tag{9.142}$$

where $x \in (1, 1/k]$. By our preceding discussion (9.142) becomes larger at larger k.

Since all involved factors are greater than unity in the transition band of $R_N(x)$, it follows that $R_N(x) > T_N(x)$ in that range and the difference grows with k. By even/odd symmetries of the respective functions we have

$$|R_N(x)| > |T_N(x)| \qquad (|x| \in (1, 1/k],\ k > 0,\ N > 1)$$

and the difference becomes larger with k.

9.13 Elliptic filters

Elliptic filters are obtained by using renormalized elliptic rational functions $\bar{R}_N(\omega)$ as $f(\omega)$ in (9.18).[22] The main motivation to use renormalized elliptic rational functions instead of ω^N and $T_N(\omega)$ is that, as we already know they grow faster than ω^N and $T_N(\omega)$ within their transition bands, which results in a steeper transition band's slope. The tradeoff is that in order to achieve a steeper transition band we need to allow ripples in the pass- and stop-bands.

Thus, in (9.18) we let

$$f(\omega) = \bar{R}_N(\omega)$$

that is

$$|H(j\omega)|^2 = \frac{1}{1 + \bar{R}_N^2(\omega)}$$

[22] Classically, elliptic filters are obtained from elliptic rational functions $R_N(\omega)$ by letting $f(\omega) = \varepsilon R_N(\omega)$ where $\varepsilon > 0$ is some small value. This way however usually requires some cutoff correction afterwards. The way how we introduce Chebyshev filters is essentially the same, but directly results in a better cutoff positioning and better symmetries. Particularly EMQF filters directly arise at $\lambda = 1$. One way is related to the other via $\varepsilon = 1/R_N(1/\sqrt{k}\lambda)$ combined with a cutoff adjustment by the factor $\sqrt{k}\lambda$.

The λ parameter of $\bar{\hat{R}}_N(\omega)$ is affecting the equiripple amplitudes of $\bar{\hat{R}}_N$ and thereby the equiripple amplitude in the pass- and stop-bands of $|H(j\omega)|$. It is convenient to introduce the additional variable

$$\varepsilon = \frac{1}{\bar{R}_N(1/\lambda)} \tag{9.143}$$

Using (9.143) we particularly may write

$$f(\omega) = \bar{\hat{R}}_N(\omega) = \varepsilon \bar{R}_N(\omega/\lambda)$$

Given a desired filter order N, we still have two further freedom degrees to play with, corresponding to the parameters k and λ, where, as we should recall from the discussion of elliptic rational functions, k defines the tradeoff between the transition bandwidth and the ripple amplitudes, and λ defines the tradeoff between the ripple amplitudes in the pass- and stop-bands. One of the possible scenarios to compute the elliptic filter parameters can be therefore the following. Suppose we are given the desired filter order N and the desired pass- and stop-band boundaries (where the passband boundary must be to the left of $\omega = 1$ and the stopband boundary must be to the right of $\omega = 1$). Recalling that the pass- and stop-bands of $\bar{\hat{R}}_N$ are (in the positive frequency range) $[0, \lambda\sqrt{k}]$ and $[\lambda/\sqrt{k}, +\infty)$ respectively, we can find λ and k.

Another scenario occurs if we are given the discrimination factor $1/\tilde{k}$. By (9.132) this defines selectivity factor $1/k$ and respectively the transition band width, but we can still play with λ to control the tradeoff between the pass- and stop-band ripples.

Since the passband amplitude of $\bar{R}_N$ is $\sqrt{k}$, the amplitude response $|H(j\omega)|$ is varying within $[1/\sqrt{1+k\varepsilon^2}, 1]$ in the passband. Since the stopband amplitude of $\bar{R}_N$ is $1/\sqrt{k}$, the amplitude response $|H(j\omega)|$ is varying within $[0, 1/\sqrt{1+\varepsilon^2/k}]$ in the stopband. The value of ε therefore affects the tradeoff between the equiripple amplitudes of $|H(j\omega)|$ in the pass- and stop-bands. Since ε depends on λ, this is consistent with our previous conclusion that λ affects the tradeoff between the equiripple amplitudes of $\bar{\hat{R}}_N$, where a smaller bandwidth of the pass- or stop-band corresponds to smaller equiripples within the respective band. Remember that the range of λ is generally restricted to $[\sqrt{k}, 1/\sqrt{k}]$ (or just a little bit wider). Fig. 9.61 illustrates.

Poles of elliptic filters

Since $\bar{\hat{R}}_N(\omega)$ is a rational function, the transfer function defined by (9.18) will have poles and zeros. The equation for the poles of $|H(s)|^2 = H(s)H(-s)$ is

$$1 + \bar{\hat{R}}_N^2(\omega) = 0$$

or

$$\bar{\hat{R}}_N(\omega) = \pm j$$

or

$$\varepsilon \bar{R}_N(\omega/\lambda) = \pm j \tag{9.144}$$

or, introducing $\bar{\omega} = \omega/\lambda$

$$\bar{R}_N(\bar{\omega}) = \pm \frac{j}{\varepsilon} \tag{9.145}$$

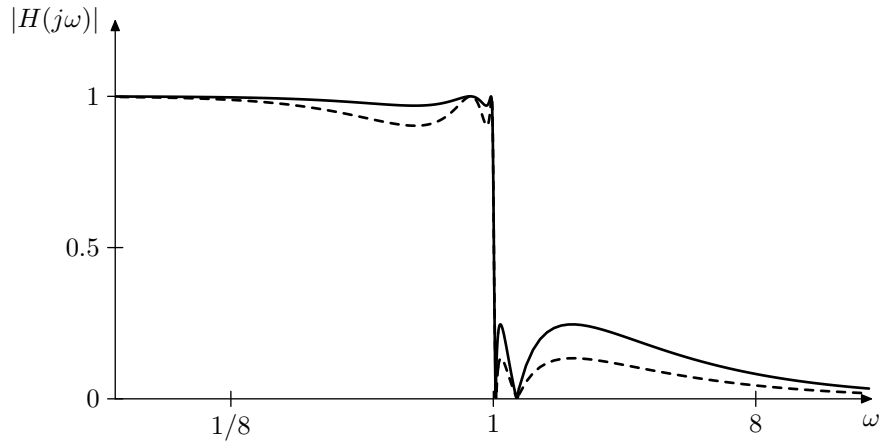


Figure 9.61: Elliptic filter's amplitude responses for $N = 5$, $k = 0.98$ and $\lambda = 1$ (solid) and $\lambda = k^{-1/4}$ (dashed). Notice the usage of the linear amplitude scale, which is chosen in order to be able to show the amplitude response zeros.

where the "+" sign corresponds to the even poles and the "−" sign to odd poles.

Recall the interpretation of $\bar{R}_N$ as a representation of linear scaling of the preimage, which is given by (9.123). Suppose $\bar{\omega}$ is moving in a counterclockwise direction in a quasielliptic curve which is a representation of some preimage line $\operatorname{Im} u = \beta$ (Figs. 9.52, 9.53). Earlier we have agreed to chose the preimages within the imaginary quarter period right below the real axis, that is $\beta \in (-jK', 0)$. In this case the counterclockwise movement in the representation domain assumes that u is moving towards the right.

Since $v = u\tilde{K}'/K'$, the corresponding line in the preimage of $\bar{R}_N(\bar{\omega})$ is $\operatorname{Im} v = \tilde{\beta}$, where $\tilde{\beta} = \beta\tilde{K}'/K'$. Therefore $\operatorname{Im} v \in (-j\tilde{K}', 0)$, that is the line also goes within the imaginary quarter period right below the real axis. Obviously, since u moves towards the right, so does v, and $\bar{R}_N(\bar{\omega})$ moves in a counterclockwise direction.

We wish $\bar{R}_N(\bar{\omega})$ to pass through the points $\pm j/\varepsilon$ going counterclockwise. By (9.71) the intersections of the quasielliptic curve $\bar{R}_N(\bar{\omega})$ with the imaginary axis are occuring at $\pm j\,\overline{\mathrm{sc}}(\tilde{\beta}, \tilde{k}')$, therefore we choose

$$\tilde{\beta} = -\,\overline{\mathrm{sc}}^{\,-1}(1/\varepsilon, \tilde{k}')$$

(which thereby belongs to $(-\tilde{K}', 0)$)[23] and

$$\beta = \frac{K'}{\tilde{K}'}\tilde{\beta} = -\frac{K'}{\tilde{K}'}\,\overline{\mathrm{sc}}^{\,-1}(1/\varepsilon, \tilde{k}') = -\frac{K}{N\tilde{K}}\,\overline{\mathrm{sc}}^{\,-1}(1/\varepsilon, \tilde{k}')$$

According to (9.71) the purely imaginary values of $\mathrm{cd}(v, \tilde{k})$ are attained when the real part of the elliptic cosine's argument is equal to $(2n+1)\tilde{K}$, where $n \in \mathbb{Z}$.

[23] Instead of $-\,\overline{\mathrm{sc}}^{\,-1}(1/\varepsilon, \tilde{k}')$, which can be expected to have an unambiguous principal value, one can equivalently compute $-\,\overline{\mathrm{cd}}^{\,-1}(j/\varepsilon, \tilde{k}')$, however, as there are multiple solutions to the equation $\overline{\mathrm{cd}}(x, \tilde{k}') = j/\varepsilon$, one needs to be careful to be sure that the $\overline{\mathrm{cd}}^{\,-1}$ routine returns the same value as would have been obtained by using $\overline{\mathrm{sc}}^{\,-1}$. In principle any solution of $\overline{\mathrm{cd}}(x, \tilde{k}') = j/\varepsilon$ would do, but may result in a different order of iteration of elliptic filter's poles, so that the unstable poles will be obtained first.

Thus, the values $\pm j/\varepsilon$ will be attained by $\bar{R}_N(\bar{\omega})$ at

$$v = j\tilde{\beta} + (2n+1)\tilde{K} \tag{9.146}$$

where, since $\tilde{\beta} < 0$, the value $\bar{R}_N(\bar{\omega}) = j/\varepsilon$ is attained at $n = 0$ and other even values of n. Thus, the solutions of the even pole equation $f = j$ will occur at even values of n. Fig. 9.62 illustrates.

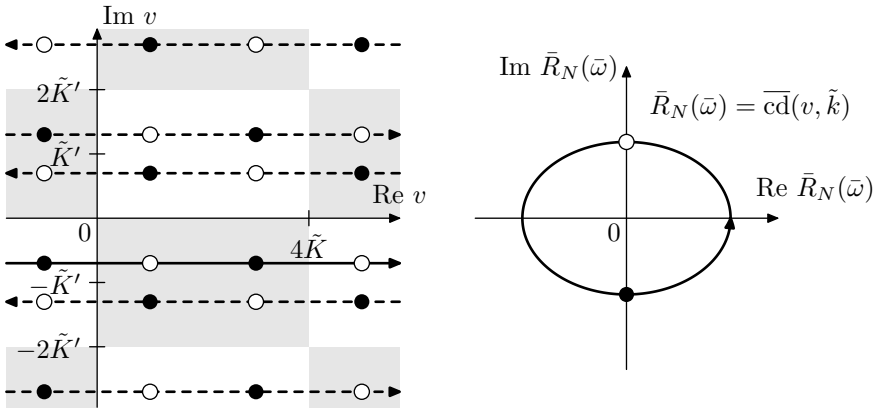


Figure 9.62: Preimages of $\bar{R}_N(\bar{\omega}) = \pm j/\varepsilon$ (qualitatively).

From (9.146) we obtain

$$u = j\beta + \frac{K}{N\tilde{K}}(2n+1)\tilde{K} = j\beta + K\frac{2n+1}{N} = j\beta + 2K\frac{\frac{1}{2}+n}{N}$$

where there are $2N$ essentially different preimages of $\bar{\omega}$ occuring at $2N$ consecutive values of n all lying on the line $\operatorname{Im} u = \beta$. Going back to the representation domain we obtain $\bar{\omega}$ lying on the respective quasiellipse:

$$\bar{\omega} = \mathrm{cd}\left(j\beta + K\frac{2n+1}{N}, k\right)$$

Fig. 9.63 illustrates.

Switching to $\omega = \lambda\bar{\omega}$:

$$\omega = \lambda\,\overline{\mathrm{cd}}\left(j\beta + K\frac{2n+1}{N}, k\right)$$

It is easily checked that the values of ω are moving counterclockwise starting from the positive real semiaxis, where the values occurring at even/odd n correspond to even/odd poles respectively.

Switching from ω to $s = j\omega$ we obtain the expression for the poles:

$$s = j\lambda\,\overline{\mathrm{cd}}\left(j\beta + K\frac{2n+1}{N}, k\right) \tag{9.147}$$

Since the values of ω are moving counterclockwise starting from the real positive semiaxis, the values of s are moving counterclockwise starting from the imaginary "positive" semiaxis, which means that starting at $n = 0$ we first obtain the stable poles at $n = 0, \ldots, N-1$. The next N values of n will give the unstable poles.

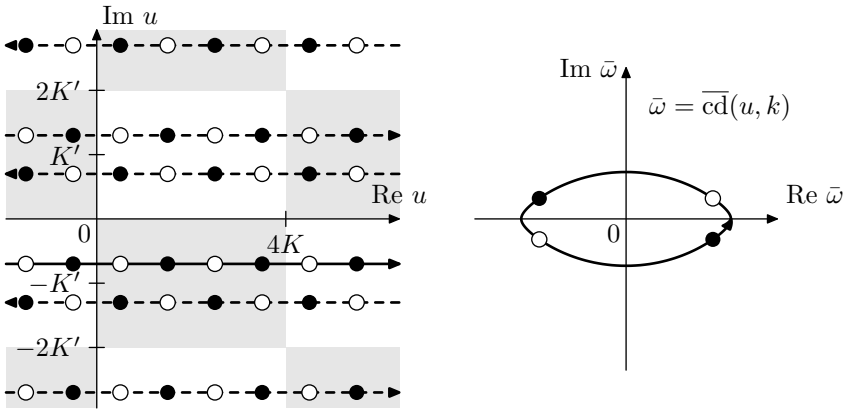


Figure 9.63: Transformation of Fig. 9.62 by $u = Kv/N\tilde{K}$ (for $N = 2$). The white and black dots on the quasielliptic curve are even/odd elliptic poles in terms of $\bar\omega$. (The picture is qualitative.)

Zeros of elliptic filters

The zeros of $H(s)$, if expressed in terms of ω coincide with the poles of $f(\omega) = \bar{R}_N(\omega,\lambda)$, which can be found from the poles $\bar{p}_n = 1/\bar{z}_n$ of $\bar{R}_N(\omega)$ as $\hat{\bar{p}}_n = \lambda\bar{p}_n = \lambda/\bar{z}_n$, where $\bar{z}_n$ are given by (9.130). By $s = j\omega$ the zeros can be reexpressed in terms of s.

One should remember that for odd N the function $\bar{R}_N$ has a zero at the origin which doesn't have a corresponding finite pole of $\bar{R}_N$, respectively there is no corresponding finite zero of $H(s)$ and no corresponding factor in the numerator of $H(s)$. Respectively the order of the numerator of $H(s)$ is $N-1$ rather than N, and the zero at the inifinity occurs automatically due to the order of the numerator being less than the order of the denominator. Fig. 9.64 provides an example.

In Fig. 9.64 one can notice that the poles are condensed closer to the imaginary axis. Apparently, this is due to (9.81c) and the related explanation.

Gain adjustments

The default normalization of the elliptic filter's gain is according to (9.18):

$$H(0) = \frac{1}{\sqrt{1+\hat{\bar{R}}_N^2(0)}} = \frac{1}{\sqrt{1+\varepsilon^2\bar{R}_N^2(0)}} = \frac{1}{\sqrt{1+\varepsilon^2\,(\mathrm{Re}\,j^N)^2\,\tilde{k}}} \tag{9.148}$$

which thereby defines the leading gain coefficient of the cascade form implementation (8.1). We could also find the leading gain from the requirement $|H(j)|^2 = 1/2$, but we should mind the possibility of accidentally obtaining a 180° phase response at $\omega = 0$.

With the leading gain defined this way the amplitude response varies within $[1/\sqrt{1+\tilde{k}\varepsilon^2}, 1]$ in the passband. We could choose some other normalizations, though. E.g. we could require $H(0) = 1$. Or we could require the passband ripples to be symmetric relatively to the zero decibel level, which is achieved by

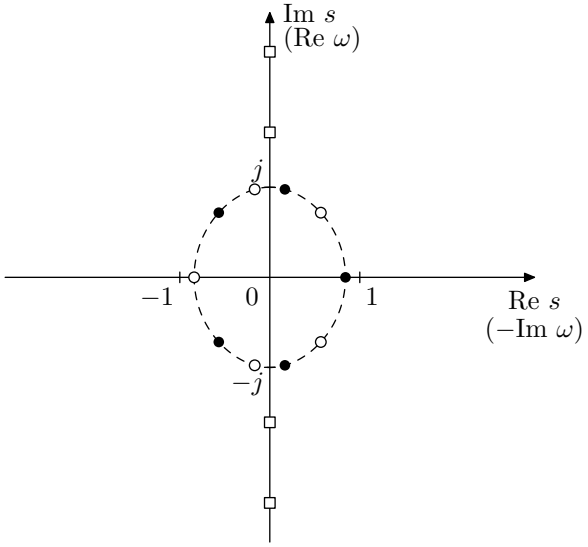


Figure 9.64: Poles (white and black dots) and zeros (white squares) of an elliptic filter of order $N = 5$. Each of the zeros is duplicated, but the duplicates are dropped together with the unstable poles.

multiplying (9.148) by $(1 + \tilde{k}\varepsilon^2)^{1/4}$:

$$H(0) = \sqrt{\frac{\sqrt{1 + \tilde{k}\varepsilon^2}}{1 + \varepsilon^2 \bar{R}_N^2(0)}}$$

so that $|H(j\omega)|$ varies within $[1/(1 + \tilde{k}\varepsilon^2)^{1/4}, (1 + \tilde{k}\varepsilon^2)^{1/4}]$ within the passband.

Elliptic minimum Q filters

At $\lambda = 1$ we have $\bar{R}_N(\omega) = \bar{R}_N(\omega)$, thus $f(\omega)$ attains the reciprocal symmetry (9.125). Simultaneously, by (9.143) $\varepsilon = 1$, respectively

$$\tilde{\beta} = -\,\overline{\mathrm{sc}}^{\,-1}(1, \tilde{k}') = -\tilde{K}'/2$$

$$\beta = \frac{K'}{\tilde{K}'}\tilde{\beta} = -K/2$$

while (9.147) turns into

$$s = j\,\overline{\mathrm{cd}}\left(j\frac{K}{2} + K\frac{2n+1}{N}, k\right) \tag{9.149}$$

By Fig. 9.53 and (9.85) the poles of $H(s)$ are all lying on the unit circle.[24] Further, by Figs. 9.54, 9.55 and the associated discussion, the values of $\overline{\mathrm{cd}}$ in (9.149) are having the maximum possible angular deviation (among the ones arising from different values for k) from the real axis. Respectively, the poles

[24] From a slightly different angle, since $\lambda = 1$ and $\varepsilon = 1$, the pole equation turns into $\bar{R}_N(\omega) = \pm j$. By (9.127) the solutions are lying on the unit circle.

given by (9.149) are having the maximum possible angular deviation from the imaginary axis, which means that the corresponding cascade 2-pole sections will have the minimum possible resonances. Therefore elliptic filters arising at $\lambda = 1$ are referred to as *elliptic minimum Q filters*, or shortly EMQF.

Butterworth and Chebyshev limits

By (9.135) at $k = 0$ EMQF filters turn into Butterworth filters. By (9.137) at $k \to 0$ and $\lambda = 1/\sqrt{k}$ elliptic filters turn into Chebyshev type I filters. By (9.138) at $k \to 0$ and $\lambda = \sqrt{k}$ elliptic filters turn into Chebyshev type II filters.

SUMMARY

Classical signal processing filters are defined in terms of the squared amplitude response equation (9.18). By choosing different function types as $f(\omega)$ in (9.18) one obtains the respective filter types:

Butterworth	$f(x) = x^N$
Chebyshev type I	$f(x) = \hat{T}_N(x)$
Chebyshev type II	$f(x) = \hat{\underline{L}}_N(x)$
Elliptic	$f(x) = \hat{\bar{R}}_N(x)$

Chapter 10

Special filter types

Butterworth filters of the 1st and 2nd kind as well as elliptic filters can serve as a basis to contruct other filter types of a more specialzed nature, which are going to be the subject of this chapter.

10.1 Reciprocally symmetric functions

The reciprocal symmetry (9.125) seems to be responsible for the special properties of EMQF filters. There is indeed a strong relationship between those, which is worth a dedicated discussion, because we will have more uses of such functions throughout this text. Let's therefore suppose that $f(x)$ (used in (9.18)) satisfies

$$f(1/x) = 1/f(x) \tag{10.1}$$

Reciprocal symmetry of the poles

An obvious conjecture which might appear from the discussion of EMQF filters is that (10.1) implies the poles on the unit circle. This, however, is not exactly true, although there is some relation.

The symmetry (10.1) actually implies the reciprocal symmetry of the filter's poles. That is, if s is a pole of $H(s)$, then so is $1/s$. Indeed, suppose $f^2(-js) = -1$, which means s is a pole of $H(s)$. Then $f^2(-j/s) = f^2(1/js) = 1/f^2(js) = 1/f^2(-js) = -1$ (where the latter transformation is by the fact that f is required to be odd or even) and thus $1/s$ is also a pole of $H(s)$.

The reciprocal symmetry of the filter's poles manifests itself nicely for the poles on the unit circle, where the reciprocation turns into simply conjugation, and as poles of the real filters must be conjugate symmetric, they are also reciprocally symmetric. But the poles do not really have to lie on the unit circle.

Image of the unit circle

$f(x)$ maps unit circle to the unit circle.[1] Indeed, first notice that $|x| = 1 \iff 1/x = x^*$. Suppose $|x| = 1$. Then, recalling that f is real,

$$f(1/x) = f(x^*) = f^*(x) = 1/f(x) \iff |f(x)| = 1$$

Therefore

$$|x| = 1 \implies |f(x)| = 1$$

The converse is however not necessarily true: it's possible that $\exists x\colon |x| \neq 1,\ |f(x)| = 1$. In other words, there may be other preimages of the unit circle points.

E.g. consider the function

$$f(x) = \rho_{+1}\left((\rho_{+1}(x))^3\right) = x\frac{x^2+3}{3x^2+1}$$

Apparently $f(x)$ satisfies (10.1). However, it has three different preimages of the unit circle:

$$x(t) = \rho_{+1}(e^{j\alpha}t)$$

where $t \in \mathbb{R}$ and α is one of the values $\pi/6$, $3\pi/6$, $5\pi/6$. At $\alpha = 3\pi/6 = \pi/2$ we obtain $|x| = 1$, however for other α this is not so.

Poles on the unit circle

Under the additional restriction that the zeros of $f(x)$ (including a possible zero at $x = \infty$) must be either all inside or all outside of the unit circle, the unit circle will be the only preimage of the unit circle, that is

$$|x| = 1 \iff |f(x)| = 1 \tag{10.2}$$

We have already shown that $|x| = 1 \implies |f(x)| = 1$, therefore it remains for us to show that $|x| = 1 \impliedby |f(x)| = 1$. First notice that (10.1) implies that the poles of $f(x)$ are reciprocals of the zeros. From (10.1) we also have $f(1) = \pm 1$. Then $f(x)$ can be written as

$$f(x) = \prod_n \frac{x - z_n}{1 - z_n x}$$

where z_n are the zeros of $f(x)$. Furthermore, since f is real, complex zeros must come in conjugate pairs and so must complex poles, and we can write

$$f(x) = \prod_n \frac{x - z_n}{1 - z_n^* x} \tag{10.3}$$

Suppose all zeros of $f(x)$ lie inside the unit circle. Let's show that in this case $|x| > 1 \implies |f(x)| > 1$. Suppose $|x| > 1$. In order to show that $|f(x)| > 1$ we are going to show that each of each of the factors of $f(x)$ has absolute magnitude greater that unity:

$$\left|\frac{x - z_n}{1 - z_n^* x}\right| > 1$$

[1] Notice that incidentally this implies that $f(x)$ is a discrete-time allpass transfer function, although not necessarily describing a stable allpass.

By equivalent transformations we are having

$$|x - z_n| > |1 - z_n^* x|$$

$$(x - z_n)(x^* - z_n^*) > (1 - z_n^* x)(1 - z_n x^*)$$

$$|x|^2 - z_n x^* - z_n^* x - |z_n|^2 > 1 - z_n x^* - z_n^* x - |z_n|^2 \cdot |x|^2$$

$$(|x|^2 - 1)(1 - |z_n|^2) > 0$$

which is obviously true, therefore each of the factors of $f(x)$ is larger than 1 in absolute magnitude and so is $f(x)$. In a similar way we can show that $|x| < 1 \implies |f(x)| < 1$. Therefore there are no other images of unit circle points and we have shown that $|x| = 1 \impliedby |f(x)| = 1$. The case of all zeros lying outside the unit circle is treated similarly, where we have $|x| > 1 \implies |f(x)| < 1$ and $|x| < 1 \implies |f(x)| > 1$.

From (10.2) it follows that the solutions of the pole equation $f^2(x) = -1$ are lying on the unit circle, and so do the poles of $H(s)$ obtained from $f(x)$.

Complementary symmetry of lowpass and highpass filters

The reciprocal symmetry of the poles implies that filters $H(s)$ and $H(1/s)$ (related by the LP to HP transformation) share the same poles. Furthermore, it turns out that there is a complementary symmetry of squared amplitude responses of these filters:

$$|H(j\omega)|^2 + |H(1/j\omega)|^2 = 1 \iff f(1/x) = 1/f(x) \tag{10.4}$$

Indeed, by the Hermitian property of $H(1/j\omega)$, the left-hand side of (10.4) can be equivalently written as

$$|H(j\omega)|^2 + |H(j/\omega)|^2 = 1$$

Using (9.18) we further rewrite it as

$$\frac{1}{1 + f^2(\omega)} + \frac{1}{1 + f^2(1/\omega)} = 1$$

Transforming the equation further, we obtain

$$\frac{1}{1 + f^2(1/\omega)} = 1 - \frac{1}{1 + f^2(\omega)} = \frac{f^2(\omega)}{1 + f^2(\omega)} = \frac{1}{1 + \dfrac{1}{f^2(\omega)}}$$

or equivalently

$$f^2(1/\omega) = 1/f^2(\omega)$$

or

$$f(1/\omega) = \pm 1/f(\omega)$$

Noticing that $f(1/\omega) = -1/f(\omega)$ implies $f^2(1) = -1$, which is impossible for a real $f(\omega)$, we conclude that $f(1/\omega) = -1/f(\omega)$ is not an option. Thus we simply have $f(1/\omega) = 1/f(\omega)$, which was obtained by an equivalent transformation from $|H(j\omega)|^2 + |H(1/j\omega)|^2 = 1$ and thus both conditions are equivalent.

10.2 Shelving and tilting filters

We have made some attempts to construct shelving filters in the discussions of 1- and 2-poles, but the results were lacking intuitively desired amplitude response symmetries shown in Fig. 10.1. The kind of symmetry shown in Fig. 10.1 is better expressed if we symmetrize the amplitude response further, obtaining the one in Fig. 10.2.

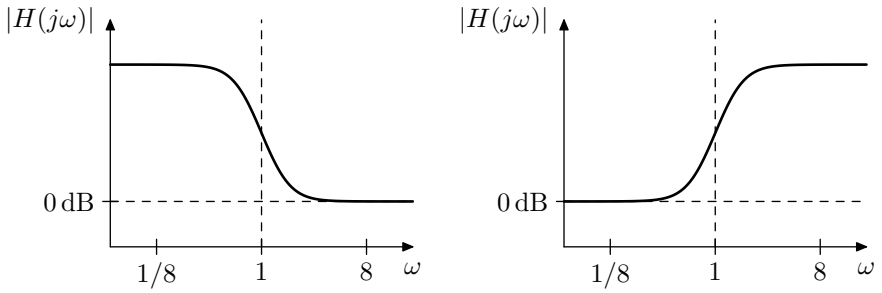


Figure 10.1: Shelving amplitude responses, ideally symmetric in fully logarithmic scale.

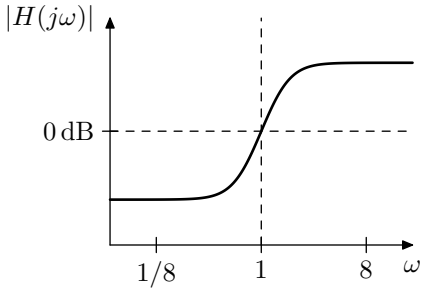


Figure 10.2: Tilting amplitude response, ideally symmetric in fully logarithmic scale.

Fig. 10.1 apparently shows low-shelving (left) and high-shelving (right) amplitude responses. The amplitude response in Fig. 10.2 can be referred to as *tilting* amplitude response. It is easy to notice that the low- and high-shelving responses can be obtained from the tilting one by vertical shifts. Vertical shifts in decibel scale are corresponding to multiplication of the signal by a constant. That is tilting and low- and high-shelving filters can be obtained from each other by a multiplication by a constant. We will therefore not make much distinction between these types, arbitrarily jumping from one type to the other, whenever the discussion requires so.

Reciprocal symmetry of poles and zeros

Treating $|H(1)| = 1$ as the logarithmic origin of an amplitude response graph we can express the desired symmetry of the tilting amplitude response in Fig. 10.2 as an odd logarithmic symmetry:

$$\log|H(j\exp(-x))| = -\log|H(j\exp x)|$$

which in linear scale becomes

$$|H(j/\omega)| = \frac{1}{|H(j\omega)|}$$

or

$$|H(j/\omega)|^2 = \frac{1}{|H(j\omega)|^2} \tag{10.5}$$

Writing $|H(j\omega)|^2$ as $H(j\omega)H(-j\omega)$ we introduce $G(s) = H(s)H(-s)$. Then (10.5) becomes

$$G(j/\omega) = \frac{1}{G(j\omega)}$$

Taking into account that $G(s)$ is even, we have

$$G(j\omega)G(1/j\omega) = 1$$

Apparently, the latter equality must be true not only for $\omega \in \mathbb{R}$ but also for any $\omega \in \mathbb{C}$, thus

$$G(s)G(1/s) = 1 \qquad (s \in \mathbb{C})$$

Therefore, $G(s) = 0 \iff G(1/s) = \infty$ and $G(s) = \infty \iff G(1/s) = 0$. That is the poles of $G(s)$ are reciprocals of the zeros of $G(s)$ and vice versa.

Conversely, given $G(s) = H(s)H(-s)$ such that its poles are reciprocals of its zeros and additionally requiring that $|G(j)| = 1$ we will have $G(s)G(1/s) = 1$ and (10.5) follows. Indeed, writing $G(s)$ in the factored form we have

$$G(s) = g \cdot \prod_n \frac{s - z_n}{1 - z_n s}$$

where z_n are the zeros of $G(s)$. Then

$$G(1/s) = g \cdot \prod_n \frac{1/s - z_n}{1 - z_n/s} = g \cdot \prod_n \frac{1 - z_n s}{s - z_n}$$

Therefore $G(s)G(1/s) = g^2$. Letting $s = j$ we have

$$g^2 = G(j)G(1/j) = G(j)G(-j) = G(j)G^*(j) = |G(j)|^2 = 1$$

and thus $G(s)G(1/s) = 1$.

Now the poles and zeros of $G(s)$ consist of those of $H(s)$ and their symmetric counterparts (with respect to the complex plane's origin). Under the assumption that $H(s)$ must be stable, all poles of $H(s)$ will be in the left complex semiplane. Under the additional assumption that $H(s)$ is minimum phase, so will be zeros of $H(s)$. Respectively $H(-s)$ will contain poles and zeros in the right semiplane. However, the reciprocation turns left-semiplane values into left-semiplane values and right-semiplane values into right-semiplane values. Therefore the poles of a minimum-phase stable $H(s)$ will be mutually reciprocal with the zeros of $H(s)$.

Thus, in order for a minimum phase $H(s)$ to have the tilting amplitude response symmetry of Fig. 10.2 its poles and zeros must be mutually reciprocal. Conversely, given $H(s)$ with mutually reciprocal poles and zeros (which is thereby minimum phase, assuming $H(s)$ is stable), (10.5) will hold under the additional requirement $|H(j)| = 1$. Relaxing the minimum phase requirement effectivlely means that some zeros will be flipped from the left semiplane into the right semiplane, which is pretty trivial and doesn't change the amplitude response, therefore we will concentrate on minimum phase tilting and shelving filters.

Construction as a lowpass ratio[2]

Let $G(s)$ be a filter (now this is a different $G(s)$ than $G(s) = H(s)H(-s)$ we have been using above) having the following properties: $G(s)$ doesn't have zeros, all its poles are lying on the unit circle, and $G(0) = 1$.

Apparently, $G(s)$ is a lowpass filter, which should be obvious by considering the factoring of $G(s)$ into a cascade of 1- and 2-poles. Such $G(s)$ also can be factored as

$$G(s) = \prod_{n=1}^{N} \frac{1}{s - p_n}$$

(where the leading coefficient is 1 due to $G(s)$ being real stable, $G(0) = 1$, $|p_n| = 1$ and $\operatorname{Re} p_n < 0$). The poles of $G(s)$ lying on the unit circle and being conjugate symmetric imply the reciprocal symmetry of the poles: if p_n is a pole of $G(s)$ then so is $1/p_n$.

Let's apply the cutoff substitution $s \leftarrow s/M$ ($M \in \mathbb{R}$, $M > 0$) to $G(s)$. We obtain

$$G(s/M) = \prod_{n=1}^{N} \frac{1}{s/M - p_n} = M^N \cdot \prod_{n=1}^{N} \frac{1}{s - Mp_n}$$

That is we obtain the filter with the poles Mp_n. Respectively, shifting the cutoff in the opposite direction by the same logarithmic amount, we have

$$G(Ms) = \prod_{n=1}^{N} \frac{1}{Ms - p_n} = M^{-N} \cdot \prod_{n=1}^{N} \frac{1}{s - M^{-1}p_n}$$

That is we obtain the filter with the poles $M^{-1}p_n$.

Since for each p_n there is $p_{n'} = 1/p_n$, for each Mp_n there is $M^{-1}p_{n'} = 1/Mp_n$. That is, the poles of $G(s/M)$ are mutually reciprocal with the poles of $G(Ms)$ and we can construct

$$H(s) = \frac{G(s/M)}{G(Ms)} \tag{10.6}$$

By construction the poles of $G(Ms)$ are the zeros of $H(s)$ and the poles of $G(s/M)$ are the poles of $H(s)$. Thus the poles of $H(s)$ are reciprocal to its zeros and vice versa. However generally $|H(j)| \neq 1$ (a little bit later we'll show that $|H(j)| = M^N$). This means that $H(s)$ is not a tilting filter, but is related to the tilting filter by some factor. Noticing that $H(0) = G(0)/G(0) = 1$, we conclude that $H(s)$ must be a kind of high-shelving filter. The conversion to the tilting filter is trivial: we can simply divide the result by $|H(j)|$.

The conversion to the low-shelving filter looks more complicated, since apparently $G(\infty) = 0$ and we have a $0/0$ uncertainty evaluating $H(\infty)$. However we can notice that at $s \to \infty$ we have $G(s) \sim s^{-N}$, $G(s/M) \sim M^N s^{-N}$ and $G(Ms) \sim M^{-N}s^{-N}$, thus $H(s) \sim M^{2N}$, that is simply $H(\infty) = M^{2N}$. We therefore obtain the low-shelving filter from the high-shelving one by dividing by M^{2N}.

We can also obtain the explicit expression for the value of $|H(j)|$, where we can simply use the symmetries of the amplitude response. Since $H(s)/|H(j)|$ is

[2]The author has learned the approach of constructing a shelving filter as a ratio of two lowpasses from Teemu Voipio.

a tilting filter, it must have mutually reciprocal amplitude responses at $\omega = 0$ and $\omega = \infty$, that is

$$\left|\frac{H(0)}{|H(j)|}\right| = \left|\frac{|H(j)|}{H(\infty)}\right|$$

from where $|H(j)|^2 = |H(0)| \cdot |H(\infty)| = M^{2N}$ and $|H(j)| = M^N$. Thus the tilting filter is obtained from the high-shelving one by dividing by M^N.

Therefore we have

$$H_{\mathrm{HS}}(s) = \frac{G(s/M)}{G(Ms)} \tag{10.7a}$$

$$H_{\mathrm{tilt}}(s) = M^{-N} \cdot \frac{G(s/M)}{G(Ms)} \tag{10.7b}$$

$$H_{\mathrm{LS}}(s) = M^{-2N} \cdot \frac{G(s/M)}{G(Ms)} \tag{10.7c}$$

for the high-shelving, tilting and low-shelving filter respectively. From the values $H(0)$, $|H(j)|$, $H(\infty)$ obtained earlier for the high-shelving filter we thus obtain:

$$\begin{array}{lll} H_{\mathrm{HS}}(0) = 1 & |H_{\mathrm{HS}}(j)| = M^N & H_{\mathrm{HS}}(\infty) = M^{2N} \\ H_{\mathrm{tilt}}(0) = M^{-N} & |H_{\mathrm{tilt}}(j)| = 1 & H_{\mathrm{tilt}}(\infty) = M^N \\ H_{\mathrm{LS}}(0) = M^{-2N} & |H_{\mathrm{LS}}(j)| = M^{-N} & H_{\mathrm{LS}}(\infty) = 1 \end{array} \tag{10.8}$$

Apparently M can be greater or smaller than 1, corresponding to increasing or decreasing of the signal level in the respective range. Since M and $1/M$ are filter cutoff factors, M must be positive.

The filters constructed by the lowpass ratio approach satisfy the symmetry (10.5), however we know little about the shapes of their amplitude responses. These shapes can be arbitrary odd functions (if seen in the logarithmic scale), whereas we would like to obtain the shapes at least resembling those in Figs. 10.1 and 10.2.

Also, apparently the lowpass ratio approach can be easily applied to a Butterworth $G(s)$, since Butterworth (unit-cutoff) filters have poles on the unit circle. On the other hand, while EMQF filters also have poles on the unit circle, they don't have only poles, but also zeros, therefore this method is not directly applicable to EMQF filters.[3] In order to have a better control of the amplitude responses and to be able to build tilting and shelving filters based on EMQF filter, we will need to address the problem from a different angle.

Construction by mixing

We have already made some attempts of constructing a low-shelving filters by mixing the lowpass signal with the input signal, which weren't too successful. Instead we could attempt the same mixing in terms of squared amplitude response, in which case we at least would not have the effects of the phase response

[3] It would have been okay, if all zeros of $G(s)$ were at the origin, since in this case the zeros of $G(s/M)$ and $G(Ms)$ would be also at the origin and therefore would cancel each other. Particularly, we could have used Butterworth highpass filters in (10.6), but this wouldn't have produced any new results compared to Butterworth lowpasses.

interfering. Also, rather that constructing a low-shelving filter, we shall attempt to construct a tilting filter, in which case it is easier to express the symmetry requirement (10.5).

Suppose $G(s)$ is defined by

$$|G(j\omega)|^2 = \frac{1}{1+f^2(\omega)}$$

We construct the tilting squared amplitude response by mixing $|G(j\omega)|^2$ with the squared "amplitude response of the input signal", which is simply 1:

$$|H(j\omega)|^2 = a^2 + \frac{b^2}{1+f^2(\omega)} = \frac{\alpha^2+\beta^2 f^2(\omega)}{1+f^2(\omega)}$$

where a^2 and b^2 (or, equivalently, α^2 and β^2) denote the unknown positive mixing coefficients. We wish $|H(j\omega)|^2$ to satisfy (10.5).

Assuming a lowpass $f(x)$, that is $f(x) \to 0$ for $x \to 0$ and $f(x) \to \infty$ for $x \to \infty$, we notice that $|H(0)|^2 = \alpha^2$ and $|H(\infty)|^2 = \beta^2$, therefore (10.5) can be attained only at $\alpha^2\beta^2 = 1$ and we can drop one of these variables obtaining:

$$|H(j\omega)|^2 = \frac{\beta^{-2}+\beta^2 f^2(\omega)}{1+f^2(\omega)} \tag{10.9}$$

However there apparently are additional restrictions on $f(x)$ which ensure that (10.5) holds for any ω and not just for $\omega = 0$ and $\omega = \infty$. To find these restrictions let's substitute (10.9) into (10.5):

$$\frac{\beta^{-2}+\beta^2 f^2(1/\omega)}{1+f^2(1/\omega)} = \frac{1+f^2(\omega)}{\beta^{-2}+\beta^2 f^2(\omega)}$$

$$\begin{aligned}\beta^{-4} + f^2(1/\omega) + f^2(\omega) + \beta^4 f^2(1/\omega) f(\omega) = \\ = 1 + f^2(1/\omega) + f^2(\omega) + f^2(1/\omega) f(\omega)\end{aligned}$$

$$1 - \beta^{-4} = (\beta^4 - 1) f^2(1/\omega) f^2(\omega)$$

$$(\beta^4 - 1) f^2(1/\omega) f^2(\omega) = \frac{\beta^4 - 1}{\beta^4}$$

and

$$f^2(1/\omega) f^2(\omega) = \beta^{-4} \tag{10.10}$$

The equation (10.10) thereby ensures that (10.5) will hold.

Let $\bar{f}(\omega) = \beta f(\omega)$, or $f(\omega) = \beta^{-1}\bar{f}(\omega)$. Then (10.10) becomes[4]

$$\bar{f}(1/\omega)\bar{f}(\omega) = 1 \tag{10.11}$$

while (10.9) becomes

$$|H(j\omega)|^2 = \frac{\beta^{-2}+\bar{f}^2(\omega)}{1+\beta^{-2}\bar{f}^2(\omega)} = \beta^{-2}\cdot\frac{1+\beta^2\bar{f}^2(\omega)}{1+\beta^{-2}\bar{f}^2(\omega)} \tag{10.12}$$

[4]The other option $\bar{f}(1/\omega)\bar{f}(\omega) = -1$ implied by (10.10) implies $\bar{f}^2(1) = -1$, therefore we ignore it.

That is, we simply want $\bar{f}(\omega)$ satisfying (10.11). Then $f(\omega) = \beta^{-1}\bar{f}(\omega)$ (for any arbitrarily picked β) will satisfy (10.10) and respectively $H(s)$ will satisfy (10.5).

Comparing the above to the lowpass-ratio approach to the construction of the tilting filters, that is comparing (10.12) to (10.6) we notice obvious similarities. Essentially (10.12) is a ratio of two lowpasses $\beta^{-2}H_1/H_2$:

$$|H_1(j\omega)|^2 = \frac{1}{1+\beta^{-2}\bar{f}^2(\omega)} \qquad |H_2(j\omega)|^2 = \frac{1}{1+\beta^2\bar{f}^2(\omega)}$$

with an additional gain of β^{-2}, which occurs since this is a tilting rather than high-shelving filter.

Given a Butterworth $f(\omega) = \omega^N$, we have $f(\omega/M) = M^{-N}f(\omega)$, that is the cutoff substitution $\omega \leftarrow \omega/M$ is equivalent to choosing $\beta = M^N$, in which case (10.12) means essentially the same as (10.6). The difference appears in the EMQF case where $f(\omega/M) \neq M^{-N}f(\omega)$.

The zeros of the EMQF filter would have been exactly the problem in the case of (10.6), since the cutoff substitution also shifts the zeros, and the zeros of $G(s/M)$ do not match the zeros of $G(Ms)$, respectively they cannot cancel each other in $H(s)$ and would have resulted in zero amplitude response at the zeros of the numerator and in infinite amplitude response at the zeros of the denominator. On the other hand, in (10.12), where the EMQF zeros correspond to the poles of $\bar{f}$, the poles of $\bar{f}$ will result in identical zeros of the numerator and of the denominator of $|H(j\omega)|^2$, thus they will cancel each other. In that sense, the approach of (10.12) is more general than the one of (10.6).

We can also notice that the right-hand side of (10.12) monotonically maps the range $[0, +\infty]$ of $\bar{f}^2$ onto $[\beta^{-2}, \beta^2]$ if $|\beta| > 1$ and onto $[\beta^2, \beta^{-2}]$ if $0 < |\beta| < 1$. Since negating β doesn't have any effect on (10.12), therefore we can restrict β to $\beta > 0$, in which case we can say $|H_{\text{tilt}}(j\omega)|$ is varying between β^{-1} and β.

Obviously, (10.12) results in

$$|H_{\text{HS}}(j\omega)|^2 = \frac{1+\beta^2\bar{f}^2(\omega)}{1+\beta^{-2}\bar{f}^2(\omega)} \tag{10.13a}$$

$$|H_{\text{tilt}}(j\omega)|^2 = \beta^{-2}\cdot\frac{1+\beta^2\bar{f}^2(\omega)}{1+\beta^{-2}\bar{f}^2(\omega)} \tag{10.13b}$$

$$|H_{\text{LS}}(j\omega)|^2 = \beta^{-4}\cdot\frac{1+\beta^2\bar{f}^2(\omega)}{1+\beta^{-2}\bar{f}^2(\omega)} \tag{10.13c}$$

The values at the key points respectively are (under the restriction $\beta > 0$)

$$\begin{array}{lll} |H_{\text{HS}}(0)| = 1 & |H_{\text{HS}}(j)| = \beta & |H_{\text{HS}}|(\infty) = \beta^2 \\ |H_{\text{tilt}}(0)| = \beta^{-1} & |H_{\text{tilt}}(j)| = 1 & |H_{\text{tilt}}|(\infty) = \beta \\ |H_{\text{LS}}(0)| = \beta^{-2} & |H_{\text{LS}}(j)| = \beta^{-1} & |H_{\text{LS}}|(\infty) = 1 \end{array} \tag{10.14}$$

where we also assume that $\bar{f}(0) = 0$ and $\bar{f}(\infty) = \infty$. In the EMQF case, where $\bar{f} = \bar{R}_N$ this is not true for even N, and we need to understand (10.14) as referring to the points where $\bar{f}(\omega) = 0$ and $\bar{f}(\omega) = \infty$ instead (which we didn't explicitly write in (10.14) for the sake of keeping the notation short).

As with M in the lowpass ratio approach, β can be greater than 1 or smaller than 1.

10.3 Fixed-slope shelving

Using Butterworth filters as a basis for a shelving/tilting filter is compatible with both lowpass ratio (10.6) and mixing (10.12) approaches, where both options are giving equivalent results. For now we will continue the discussion in terms of the lowpass ratio option (10.6).

We will be interested in shelving filters obtained from the Butterworth filters of the 1st kind by (10.6). Noticing that (10.6) commutes with the Butterworth transformation:

$$\mathcal{B}_N\left[\frac{G(s/M)}{G(Ms)}\right] = \frac{\mathcal{B}_N\left[G(s/M)\right]}{\mathcal{B}_N\left[G(Ms)\right]} \tag{10.15}$$

we can restrict our discussion to the shelving filters obtained by the application of (10.6) to the 1st-order Butterworth lowpass. By (10.15) higher order shelving filters will be simply Butterworth transformations of the 1st-order Butterworth shelving filters, which is going to be covered in Section 10.5.

Since the 1st-order Butterworth lowpass coincides with the ordinary 1-pole lowpass, we simply have

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

and respectively by (10.7)

$$\begin{aligned}
H_{\mathrm{HS}}(s) &= \frac{G(s/M)}{G(Ms)} = \frac{1+Ms}{1+s/M} = M^2 \cdot \frac{s+1/M}{s+M} \\
H_{\mathrm{tilt}}(s) &= M^{-1}H_{\mathrm{HS}}(s) = M^{-1} \cdot \frac{1+Ms}{1+s/M} = \frac{Ms+1}{s+M} = M\frac{s+1/M}{s+M} \\
H_{\mathrm{LS}}(s) &= M^{-1}H_{\mathrm{tilt}}(s) = M^{-2} \cdot \frac{1+Ms}{1+s/M} = \frac{s+1/M}{s+M}
\end{aligned}$$

By (10.8)

$$\begin{array}{lll}
H_{\mathrm{HS}}(0) = 1 & |H_{\mathrm{HS}}(j)| = M & H_{\mathrm{HS}}(\infty) = M^2 \\
H_{\mathrm{tilt}}(0) = M^{-1} & |H_{\mathrm{tilt}}(j)| = 1 & H_{\mathrm{tilt}}(\infty) = M \\
H_{\mathrm{LS}}(0) = M^{-2} & |H_{\mathrm{LS}}(j)| = M^{-1} & H_{\mathrm{LS}}(\infty) = 1
\end{array}$$

Implementation

The 1st-order tilting filter can be implemented as a linear combination of the lowpass and highpass signals:

$$\begin{aligned}
H_{\mathrm{tilt}}(s) = \frac{Ms+1}{s+M} = \frac{s+1/M}{s/M+1} = M\frac{s/M}{s/M+1} + M^{-1}\frac{1}{s/M+1} = \\
= M^{-1} \cdot H_{\mathrm{LP}}(s) + M \cdot H_{\mathrm{HP}}(s)
\end{aligned}$$

where the cutoff of the 1-pole multimode is at $\omega = M$. The low- and high-shelving filters can be obtained from the above mixture by a division a or multiplication by M:

$$\begin{aligned}
H_{\mathrm{HS}}(s) &= M \cdot H_{\mathrm{tilt}}(s) = H_{\mathrm{LP}}(s) + M^2 \cdot H_{\mathrm{HP}}(s) \\
H_{\mathrm{LS}}(s) &= M^{-1} \cdot H_{\mathrm{tilt}}(s) = M^{-2} \cdot H_{\mathrm{LP}}(s) + H_{\mathrm{HP}}(s)k
\end{aligned}$$

Amplitude response

The example amplitude responses of the 1-pole tilting filter are presented in Fig. 10.3, where the formal "cutoff" frequency is denoted as ω_{mid}, being the middle frequency of the tilting.

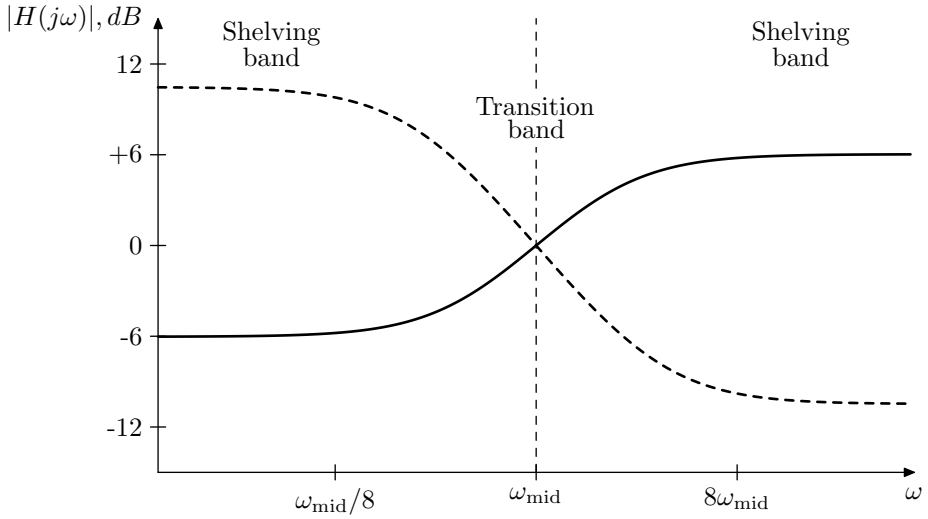


Figure 10.3: Amplitude responses of a 1-pole tilting filter for $M > 1$ (solid) and $M < 1$ (dashed).

Since the amplitude response of the tilting filter neither decreases to zero anywhere, nor does it have a range where it is approximately unity, we can't define pass- and stop-bands. Instead we can refer to the bands on the left and on the right, where the amplitude response is almost constant, as *shelving bands*. The band in the middle where the amplitude response is varying can be referred to as *transition band*, as usual.

On the other hand, for low- and high-shelving filters we can define one of the bands, where the amplitude response is approximately unity, as the passband (Figs. 10.4, 10.5).

Phase response

The representation of the shelving filter as a ratio of two lowpasses with cutoffs M and M^{-1} allows an intuitive derivation of the tilting filter's phase response, the latter being equal to the difference of the lowpass phase responses:

$$\begin{aligned}\arg H_{\text{HS}}(s) = \arg\frac{1+Ms}{1+s/M} = \arg\frac{1+s/M^{-1}}{1+s/M} = \\ = \arg\frac{1}{1+s/M} - \arg\frac{1}{1+s/M^{-1}} \qquad (s=j\omega)\end{aligned}$$

(since both shelving filters and the tilting filter all have identical phase responses, we picked the one with the most convenient transfer function).

Recalling how the phase response of a 1-pole lowpass looks (Fig. 2.5) we can conclude that the biggest deviation of the tilting filter's phase response from 0°

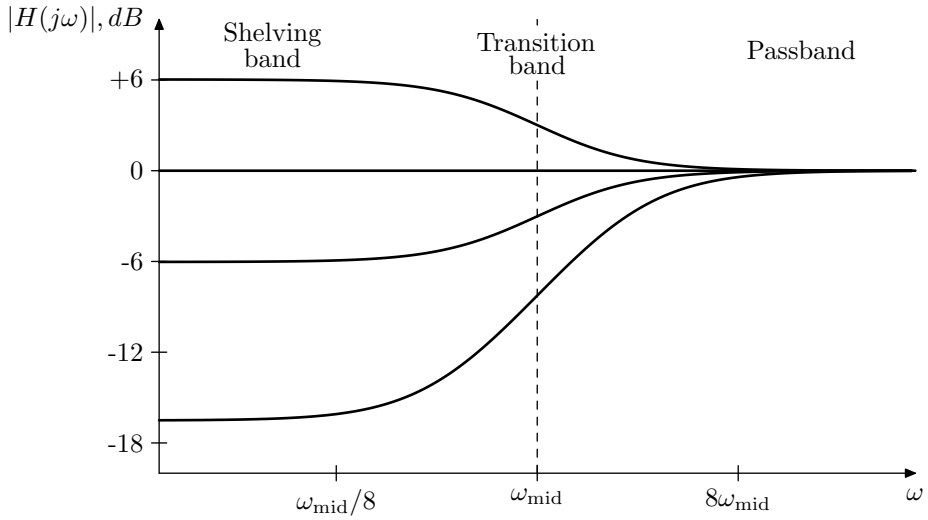


Figure 10.4: Amplitude responses of a 1-pole low-shelving filter.

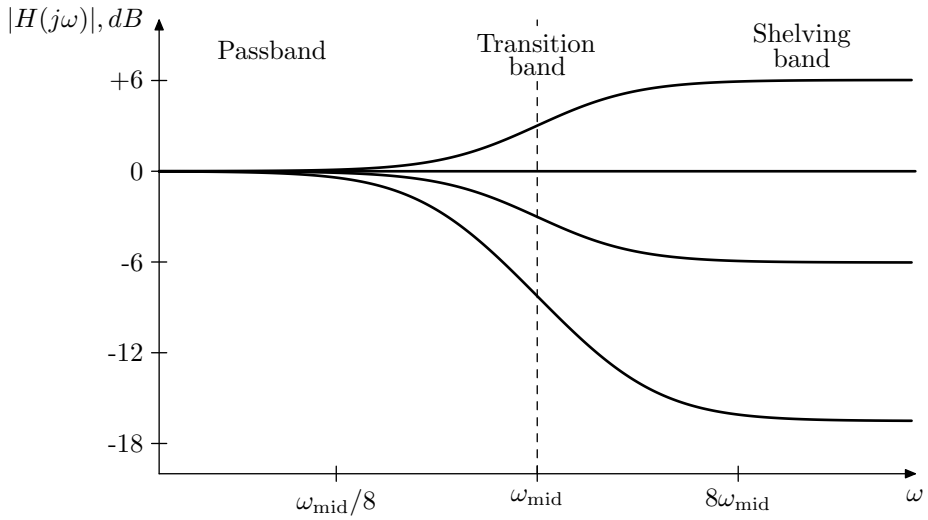


Figure 10.5: Amplitude responses of a 1-pole high-shelving filter.

(potentially reaching almost $\pm 90°$) should occur in the frequency band between M and M^{-1}, the deviation being positive if $M > 1 > M^{-1}$ and negative if $M < 1 < M^{-1}$. Outside of this band the phase response cannot exceed $\pm 45°$. Fig. 10.6 illustrates. Notice that therefore the phase response is close to zero outside of the transition region.

Transition band width

In order to roughly estimate the width of the transition band of the tilting filter's amplitude response we could divide the decibel difference in amplitude response levels at $\omega \to 0$ and $\omega \to \infty$ by the derivative of the amplitude response at the

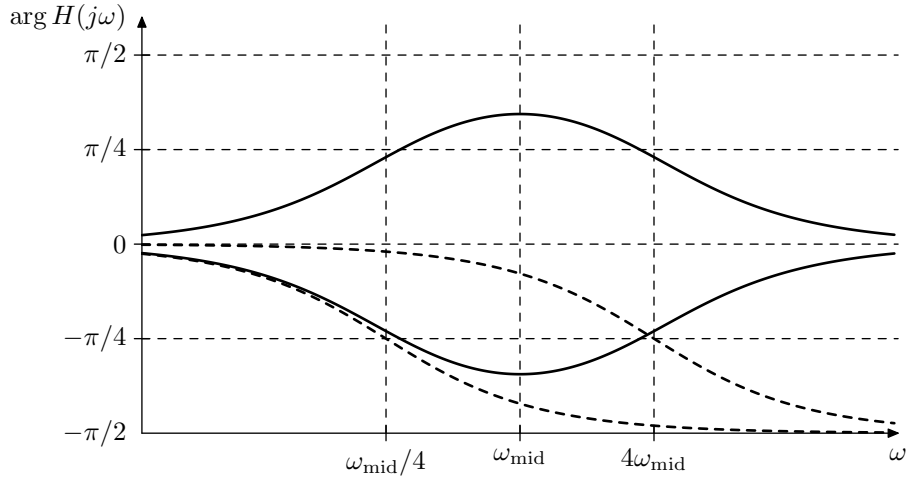


Figure 10.6: Phase response of the 1-pole shelving/tilting filters for $M = 4$ (positive) and for $M = 1/4$ (negative). Dashed curves represent phase responses of the underlying 1-pole lowpasses at cutoffs M and M^{-1}.

middle of the transition band (Fig. 10.7).

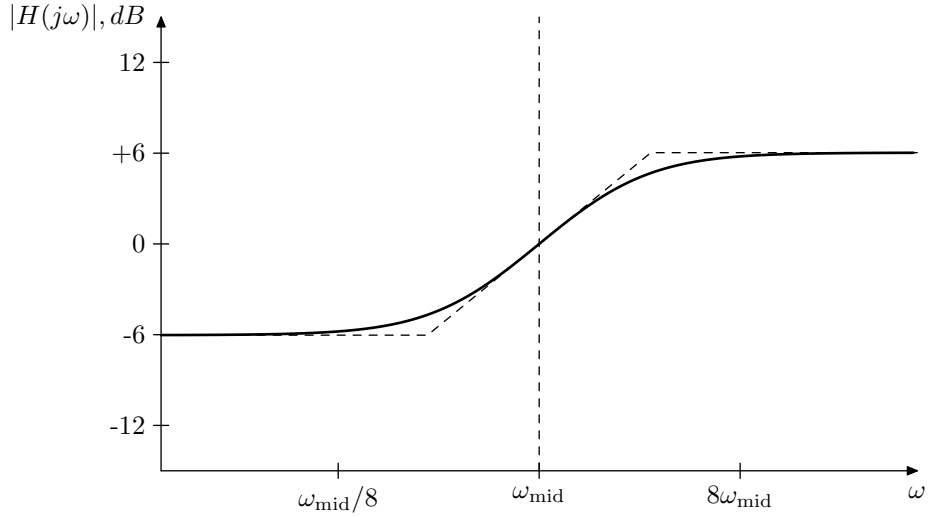


Figure 10.7: Estimation of the transition bandwidth of the 1-pole tilting filter by approximating the amplitude response by a broken line tangential to the amplitude response at the middle frequency.

Writing out the derivative of the amplitude response in the logarithmic frequency and amplitude scales (where we use natural logarithms to simplify the math), we obtain

$$\frac{\mathrm{d}}{\mathrm{d}x} \ln |H_{\rm tilt}(je^x)| \bigg|_{x=0} = |H_{\rm tilt}(je^x)|^{-1} \bigg|_{x=0} \cdot \frac{\mathrm{d}}{\mathrm{d}x} |H_{\rm tilt}(je^x)| \bigg|_{x=0} =$$

$$= \frac{\mathrm{d}}{\mathrm{d}x}|H_{\text{tilt}}(je^x)| = \frac{\mathrm{d}}{2\mathrm{d}x}|H_{\text{tilt}}(je^x)|^2 = \frac{\mathrm{d}}{2\mathrm{d}x}\left|\frac{jMe^x+1}{je^x+M}\right|^2 =$$
$$= \frac{\mathrm{d}}{2\mathrm{d}x}\left(\frac{M^2e^{2x}+1}{e^{2x}+M^2}\right) = \frac{2M^2e^{2x}(e^{2x}+M^2)-2e^{2x}(M^2e^{2x}+1)}{2(e^{2x}+M^2)^2}\bigg|_{x=0} =$$
$$= \frac{2M^2(1+M^2)-2(M^2+1)}{2(1+M^2)^2} = \frac{M^2-1}{M^2+1} = \frac{M-M^{-1}}{M+M^{-1}}$$

(where we have assumed $x = 0$ throughout the entire transformation chain). The logarithmic amplitude difference is $\ln M - \ln M^{-1} = 2\ln M$ and thus the bandwidth in terms of natural logarithmic scale is the ratio of that difference and the derivative at $x = 0$:

$$\Delta_{\ln} = 2\ln M \cdot \frac{M+M^{-1}}{M-M^{-1}}$$

Introducing the natural-logarithmic amplitude boost $m = \ln M$, we rewrite the above as

$$\Delta_{\ln} = 2m \cdot \frac{e^m+e^{-m}}{e^m-e^{-m}} = 2 \cdot \frac{m}{\tanh m} = \frac{2}{\operatorname{tanhc} m}$$

where

$$\operatorname{tanhc} m = \frac{\tanh m}{m}$$

is the "cardinal hyperbolic tangent" function, introduced similarly to the more commonly known cardinal sine function $\operatorname{sinc} x = \frac{\sin x}{x}$.

Introducing the decibel difference betwen the right and left "shelves"

$$G_{\text{dB}} = 20\log_{10} M^2$$

we can switch the amplitude scale from natural logarithmic to decibel:

$$M = e^m \qquad M^2 = 10^{G_{\text{dB}}/20}$$
$$2m = \ln 10^{G_{\text{dB}}/20} = G_{\text{dB}}/20 \cdot \ln 10$$
$$m = G_{\text{dB}}/40 \cdot \ln 10 \approx 0.0576 \cdot G_{\text{dB}}$$

Then

$$\Delta_{\ln} \approx \frac{2}{\operatorname{tanhc}(0.0576 \cdot G_{\text{dB}})}$$

Switching from the natural logarithmic bandwidth to the octave bandwidth we have

$$e^{\Delta_{\ln}} = 2^{\Delta_{\text{oct}}}$$
$$\Delta_{\ln} = \Delta_{\text{oct}} \cdot \ln 2$$

and we have obtained the octave bandwidth formula:

$$\Delta_{\text{oct}} \approx \frac{2}{\ln 2 \cdot \operatorname{tanhc}(0.0576 \cdot G_{\text{dB}})} \approx \frac{2.89}{\operatorname{tanhc}(0.0576 \cdot G_{\text{dB}})}$$

The graph of the dependency is plotted in Fig. 10.8. At $G_{\text{dB}} = 0$ we have $\Delta_{\text{oct}} \approx 2.89$. At $|G_{\text{dB}}| \to \infty$ we have $\Delta_{\text{oct}} \sim G_{\text{dB}}/6$, that is the bandwidth is growing proportionally to the decibel boost.[5] Since the shelving boosts are typically within the range of ±12dB, or maybe ±18dB, we could say that the typical transition band width of the titling filter is roughly 3 octaves.

[5]It's not difficult to realize that the 6 in the denominator is 1-pole lowpass filter's rolloff of 6dB/oct.

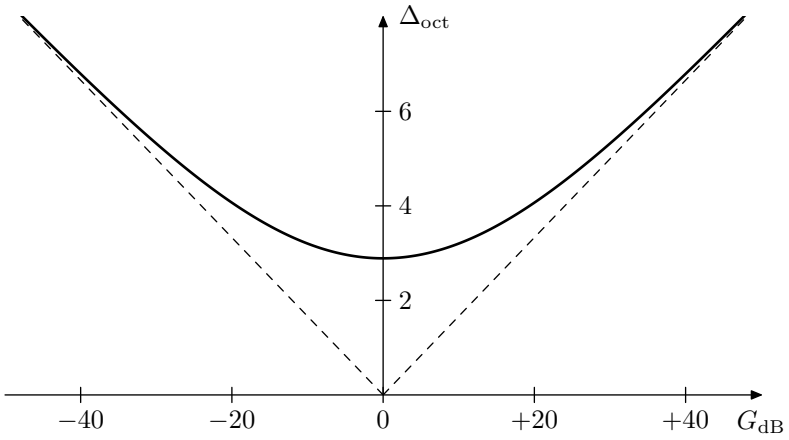


Figure 10.8: Transition bandwidth (estimated) as a function of the total decibel boost.

10.4 Variable-slope shelving

With shelving filters based on Butterworth filters of the 1st kind we didn't have any control over the steepness of the transition slope, or, respectively over the transition band width. In order to introduce that kind of control we can use Butterworth filters of the 2nd kind.

The commutativity relation (10.15) still applies, since it's independent of whether the involved filters are 1st or 2nd kind Butterworth, and we can restrict our discussion to the 2-pole shelving filters. Shelving filters of higher (even) orders can be obtained from those by Butterworth transformation, which is going to be covered in Section 10.5.

A generic unit-cutoff 2-pole lowpass filter

$$G(s) = \frac{1}{s^2 + 2Rs + 1} \tag{10.16}$$

has its poles on the unit circle, no zeros and unity gain at $\omega = 0$, thus the requirements of the lowpass ratio approach are fulfilled and we can obtain the respective shelving filters by (10.7):

$$H_{\text{HS}}(s) = \frac{G(s/M)}{G(Ms)} = \frac{M^2s^2 + 2RMs + 1}{s^2/M^2 + 2Rs/M + 1} = M^2 \cdot \frac{M^2s^2 + 2RMs + 1}{s^2 + 2RMs + M^2}$$

$$H_{\text{tilt}}(s) = M^{-2}H_{\text{HS}}(s) = \frac{M^2s^2 + 2RMs + 1}{s^2 + 2RMs + M^2}$$

$$H_{\text{LS}}(s) = M^{-2}H_{\text{tilt}}(s) = M^{-2} \cdot \frac{M^2s^2 + 2RMs + 1}{s^2 + 2RMs + M^2} = \frac{s^2 + 2Rs/M + 1/M^2}{s^2 + 2RMs + M^2}$$

where by (10.8)

$$\begin{array}{lll} H_{\text{HS}}(0) = 1 & |H_{\text{HS}}(j)| = M^2 & H_{\text{HS}}(\infty) = M^4 \\ H_{\text{tilt}}(0) = M^{-2} & |H_{\text{tilt}}(j)| = 1 & H_{\text{tilt}}(\infty) = M^2 \\ H_{\text{LS}}(0) = M^{-4} & |H_{\text{LS}}(j)| = M^{-2} & H_{\text{LS}}(\infty) = 1 \end{array}$$

Implementation

In order to construct an implementation of the tilting filter, we simply express its transfer function in terms of the SVF modes:

$$\begin{aligned} H_{\text{tilt}}(s) &= \frac{M^2 s^2 + 2RMs + 1}{s^2 + 2RMs + M^2} = \frac{s^2 + 2R(s/M) + 1/M^2}{(s/M)^2 + 2R(s/M) + 1} = \\ &= \frac{M^2 (s/M)^2 + 2R(s/M) + 1/M^2}{(s/M)^2 + 2R(s/M) + 1} = \\ &= M^{-2} H_{\text{LP}}(s) + H_{\text{BP1}}(s) + M^2 H_{\text{HP}}(s) \end{aligned}$$

where the cutoff of the multimode SVF is $\omega_c = M$ (notice that we used the normalized bandpass mode instead of the ordinary bandpass). Respectively

$$\begin{aligned} H_{\text{HS}}(s) &= M^2 H_{\text{tilt}}(s) = H_{\text{LP}}(s) + M^2 H_{\text{BP1}}(s) + M^4 H_{\text{HP}}(s) \\ H_{\text{LS}}(s) &= M^{-2} H_{\text{tilt}}(s) = M^{-4} H_{\text{LP}}(s) + M^{-2} H_{\text{BP1}}(s) + H_{\text{HP}}(s) \end{aligned}$$

Amplitude and phase response

Notably, $G(s)$ defined by (10.16) cannot be conveniently expressed in terms of (9.18), since

$$|G(j\omega)|^2 = \frac{1}{(\omega^2 - 1)^2 + 4R^2\omega^2} = \frac{1}{4R^2(1 - R^2)} \cdot \frac{1}{1 + \left(\dfrac{\omega^2 + 2R^2 - 1}{2R\sqrt{1 - R^2}} \right)^2}$$

Respectively, (10.12) doesn't apply and we cannot use the associated interpretation to reason about the shelving amplitude response shapes obtained from $G(s)$. However, we could notice that at $R = 1$ the filter $G(s)$ turns into a squared 1st-order Butterworth, while at $R = 1/\sqrt{2}$ it turns into a 2nd-order Butterworth of the 1st kind, therefore we can apply the results of Section 10.3 concluding that at least at these values of R we should expect to obtain a resonable shelving shape.

The family of amplitude responses of a 2-pole tilting filter for various R is shown in Fig. 10.9. One can see in the picture that R controls the slope, or equivalently, the width of the transition band, however only a small range of R generates "reasonable" tilting curves. We'll analyse this topic in detail a bit later.

The phase response expressed in terms of a lowpass ratio gives

$$\begin{aligned} \arg H_{\text{HS}}(s) &= \arg \frac{M^2 s^2 + 2RMs + 1}{s^2/M^2 + 2Rs/M + 1} = \arg \frac{s^2/M^{-2} + 2Rs/M^{-1} + 1}{s^2/M^2 + 2Rs/M + 1} = \\ &= \arg \frac{1}{(s/M)^2 + 2R(s/M) + 1} - \arg \frac{1}{(s/M^{-1})^2 + 2R(s/M^{-1}) + 1} \end{aligned}$$

(where $s = j\omega$). Recalling the 2-pole lowpass phase response (Fig. 4.6) we can conclude that the biggest deviation of the 2-pole tilting filter's phase response from 0° (potentially reaching almost $\pm 180^\circ$) should occur in the frequency band between M and M^{-1}, the deviation being positive if $M > 1 > M^{-1}$ and negative if $M < 1 < M^{-1}$. Outside of this band the phase response cannot exceed $\pm 90^\circ$. Fig. 10.10 illustrates. Notice that thus the phase response is close to zero outside of the transition region.

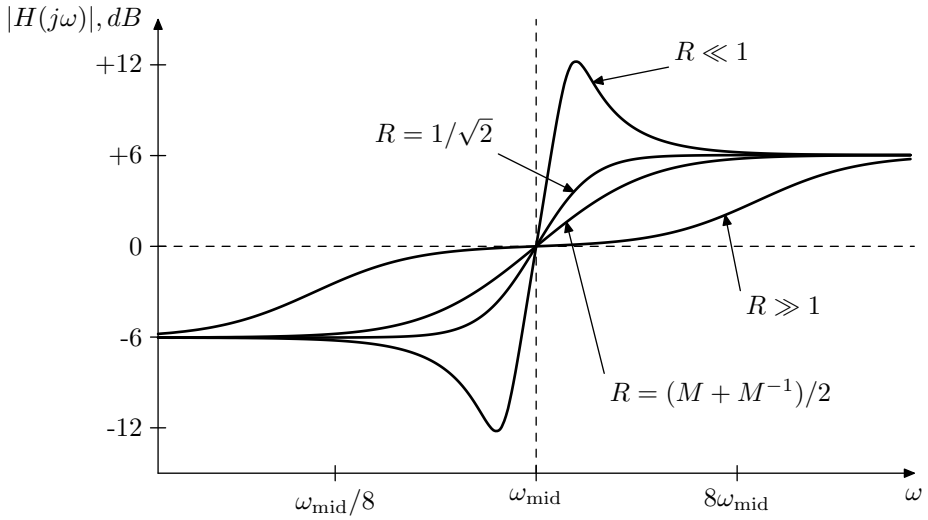


Figure 10.9: Amplitude responses of a 2-pole tilting filter for $M^2 = 2$ and various R.

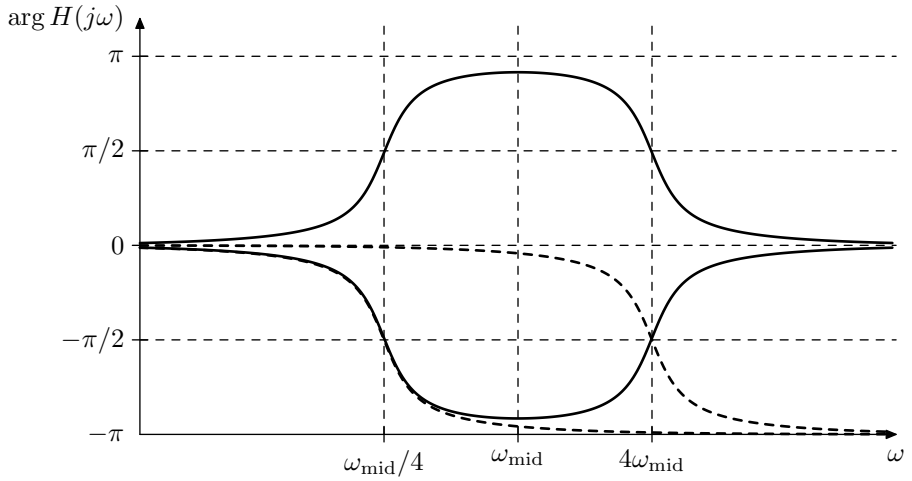


Figure 10.10: Phase response of the 2-pole tilting filter for $M = 4$ (positive) and for $M = 1/4$ (negative). Damping $R = 1/4$. Dashed curves represent phase responses of the underlying 2-pole lowpasses at cutoffs M and M^{-1}.

Steepness control

As one could notice from Fig. 10.9, the damping parameter R affects the steepness of the amplitude response slope at $\omega = 1$. Let's analyse it in more detail. First, we write $H_{\mathrm{tilt}}(s)$ as

$$H_{\mathrm{tilt}}(s) = \frac{M^2 s^2 + 2RMs + 1}{s^2 + 2RMs + M^2} = \frac{M^2 s + 2RM + 1/s}{s + 2RM + M^2/s} = \frac{G(s)}{G(1/s)}$$

where

$$G(s) = M^2 s + 2RM + 1/s$$

Then, considering the derivative of the fully logarithmic-scale amplitude response at $\omega = 1$, we obtain (assuming $x = 0$ thoughout the entire transformation chain):

$$\begin{aligned}
&\frac{\mathrm{d}}{\mathrm{d}x} \ln |H_{\text{tilt}}(je^x)| \bigg|_{x=0} = \frac{\mathrm{d}}{\mathrm{d}x} \ln \frac{|G(je^x)|}{|G(-je^{-x})|} = \frac{\mathrm{d}}{\mathrm{d}x} \ln \frac{|G(je^x)|}{|G(je^{-x})|} = \\
&= \frac{\mathrm{d}}{\mathrm{d}x} \left(\ln |G(je^x)| - \ln \left| G(je^{-x}) \right| \right) = 2 \frac{\mathrm{d}}{\mathrm{d}x} \ln |G(je^x)| = \frac{\mathrm{d}}{\mathrm{d}x} \ln |G(je^x)|^2 = \\
&= \frac{\frac{\mathrm{d}}{\mathrm{d}x} |G(je^x)|^2}{|G(je^x)|^2} = \frac{\frac{\mathrm{d}}{\mathrm{d}x} |jM^2 e^x + 2RM - je^{-x}|^2}{|G(j)|^2} = \\
&= \frac{\frac{\mathrm{d}}{\mathrm{d}x} (4R^2M^2 + (M^2 e^x - e^{-x})^2)}{4R^2M^2 + (M^2 - 1)^2} = \frac{\frac{\mathrm{d}}{\mathrm{d}x} (M^4 e^{2x} - 2M + e^{-2x})}{4R^2M^2 + (M^2 - 1)^2} = \\
&= \frac{2M^4 e^{2x} - 2e^{-2x}}{4R^2M^2 + (M^2 - 1)^2} = \frac{2(M^2 - M^{-2})}{4R^2 + (M - M^{-1})^2}
\end{aligned}$$

The maximum possible value is attained at $R = 0$ and is equal to

$$\frac{\mathrm{d}}{\mathrm{d}x} \ln |H_{\text{tilt}}(je^x)| \bigg|_{x=0} = \frac{2(M^2 - M^{-2})}{(M - M^{-1})^2} = 2 \cdot \frac{M + M^{-1}}{M - M^{-1}} < \infty \qquad \text{for } M \neq 1$$

That is, we can't reach infinite steepness. Further, as one can see from Fig. 10.9, for $R \to 0$ the amplitude response gets a peak and a dip, which are generally undesired for a shelving EQ.

On the other hand, given a sufficiently large R, we can attain arbitrarily small steepness. However, for $R \geq 1$ the filter falls apart into a product of two 1-pole tilting filters.[6] As R grows, the 1-pole cutoffs get further apart and one can see the two separate "tilting inflection points" in the amplitude response (Fig. 10.9).

We need therefore to restrict R to "a reasonable range". But how do we define this range? Let's analyse several characteristic values of R.

At $R = 1$ we have a two times vertically stretched (in the decibel scale) amplitude response of the 1-pole tilting filter with the same cutoff $\omega_c = M$:

$$\frac{M^2 s^2 + 2Ms + 1}{s^2 + 2Ms + M^2} = \left(\frac{Ms + 1}{s + M} \right)^2$$

It should be no surprise that at $R = 1/\sqrt{2}$ we obtain a Butterworth transform of the 1-pole tilting filter with cutoff $\omega_c = M^2$:

$$\left| \frac{M^2 s^2 + \sqrt{2} \cdot Ms + 1}{s^2 + \sqrt{2} \cdot Ms + M^2} \right|^2 \bigg|_{s=j\omega} = \frac{(1 - M^2\omega^2)^2 + 2M^2\omega^2}{(M^2 - \omega^2)^2 s^2 + 2M^2\omega^2} =$$

[6] This can be derived from the fact that in this case $H(s)$ has real poles and zeros which are mutually reciprocal. Thus, each such reciprocal pole/zero pair makes up a 1-pole tilting filter. The gain M^2 of the tilting 2-pole filter is distributed into two 1-pole tilting filter's gains, each equal to M.

$$= \frac{1+M^4\omega^4}{M^4+\omega^4} = \left|\frac{M^2\cdot j\omega^2+1}{j\omega^2+M^2}\right| = \left|\frac{M^2 s+1}{s+M^2}\right|\Bigg|_{s=j\omega^2}$$

We can also obtain the response identical to the response of the just mentioned 1-pole tilting filter with cutoff $\omega_c = M^2$ by combining such 1-pole tilting filter with a "unit-gain" (fully transparent) tilting filter $(s+1)/(s+1)$:

$$H_{\text{tilt}}(s) = \frac{M^2 s+1}{s+M^2} = \frac{M^2 s+1}{s+M^2}\cdot\frac{s+1}{s+1} = \frac{M^2 s^2+(M^2+1)s+1}{s^2+(M^2+1)s+M^2} =$$
$$= \frac{M^2 s^2+\dfrac{M^2+1}{M}\cdot Ms+1}{s^2+\dfrac{M^2+1}{M}\cdot Ms+M^2} = \frac{M^2 s^2+(M+M^{-1})\cdot Ms+1}{s^2+(M+M^{-1})\cdot Ms+M^2}$$

Therefore, such response is attained at $R = (M+M^{-1})/2 \geq 1$.

Thus we are having two good candidates for the boundaries of the "reasonable range of R". One boundary can be at $R = (M+M^{-1})/2$, corresponding to the amplitude response of the 1-pole tilting filter with cutoff $\omega_c = M^2$, the other boundary is at $R = 1/\sqrt{2}$, corresponding to the same response shrunk horizontally two times.[7] The steepness at $\omega = 1$ therefore varies by a factor of 2 within that range, which also can be verified explicitly:

$$\frac{\dfrac{\mathrm{d}}{\mathrm{d}x}\ln|H_{\text{tilt}}(je^x)|\Big|_{x=0,\ R=1/\sqrt{2}}}{\dfrac{\mathrm{d}}{\mathrm{d}x}\ln|H_{\text{tilt}}(je^x)|\Big|_{x=0,\ R=(M+M^{-1})/2}} =$$
$$= \frac{2(M^2-M^{-2})}{2+(M-M^{-1})^2}\cdot\frac{(M+M^{-1})^2+(M-M^{-1})^2}{2(M^2-M^{-2})} =$$
$$= \frac{(M+M^{-1})^2+(M-M^{-1})^2}{2+(M-M^{-1})^2} = \frac{2(M^2+M^{-2})}{M^2+M^{-2}} = 2$$

These boundary responses of the "reasonable range of R" can be found among the responses shown by Fig. 10.9.

10.5 Higher-order shelving

Butterworth shelving of the 1st kind

Given a 1-pole tilting filter:

$$H_{\text{tilt}}(s) = \frac{Ms+1}{s+M} = M\frac{s+M^{-1}}{s+M}$$

we have reciprocal "cutoffs" in the numerator and the denominator. Respectively the zero is reciprocal to the pole. According to (8.12c) and (8.12d), this

[7] Since the 2-pole tilting filter is essentially a ratio of two 2-pole lowpass filters with mutually reciprocal cutoffs, and since these lowpasses obtain a resonance peak at $R < 1/\sqrt{2}$, it is intuitively clear that either immediately below $R = 1/\sqrt{2}$ or possibly starting from a slightly lower boundary these peaks will show up in amplitude response of the tilting filter. As these peaks are usually undesired, we probably shouldn't go below $R = 1/\sqrt{2}$.

property is preserved by the Butterworth transformation. The dampings of the poles and zeros obtained after the Butterworth transformation of the 1st kind depend solely on the transformation order and thus are identical in the numerator and denominator:

$$\mathcal{B}\left[H_{\text{tilt}}(s)\right] = M \cdot \left(\frac{s + M^{-1/N}}{s + M^{1/N}}\right)^{N \wedge 1} \cdot \prod_n \frac{s^2 + 2R_n M^{-1/N} s + (M^{-1/N})^2}{s^2 + 2R_n M^{1/N} s + (M^{1/N})^2} =$$

$$= \left(M^{1/N} \frac{s + M^{-1/N}}{s + M^{1/N}}\right)^{N \wedge 1} \cdot \prod_n (M^{1/N})^2 \frac{s^2 + 2R_n M^{-1/N} s + (M^{-1/N})^2}{s^2 + 2R_n M^{1/N} s + (M^{1/N})^2}$$

Therefore we obtain a serial chain of 1- and 2-pole tilting filters. The low- and high-shelving filters are transformed similarly. Alternatively we can simply reuse the obtained Butterworth transformation of the tilting filter, multiplying or dividing it by the factor M.

Notice that the factor being M rather than M^N is a kind of a notational difference. After the Butterworth transformation of N-th order the original change of cutoff by the M factor will turn into a change of cutoff by the $M^{1/N}$ factor. Raising $M^{1/N}$ to the N-th power (according to (10.8)) to obtain the multiplication factors gives M.

Butterworth shelving of the 2nd kind

Remember that the "cutoffs" of the numerator and denominator of a 2-pole tilting filter are mutually reciprocal, while the "dampings" are equal:

$$H_{\text{tilt}}(s) = \frac{M^2 s^2 + 2RMs + 1}{s^2 + 2RMs + M^2} = M^2 \cdot \frac{s^2 + 2RM^{-1}s + M^{-2}}{s^2 + 2RMs + M^2}$$

so the numerator "cutoff" is M^{-1} and the denominator "cutoff" is M.

By using the Butterworth transform cutoff property (8.12c) we obtain that $\mathcal{B}\left[H_{\text{tilt}}(s)\right]$ must have the following form:

$$\mathcal{B}\left[H_{\text{tilt}}(s)\right] = M^2 \cdot \prod_{n=1}^{N} \frac{s^2 + 2R_n M^{-1/N} s + M^{-2/N}}{s^2 + 2R_n M^{1/N} s + M^{2/N}} =$$

$$= \prod_{n=1}^{N} M^{2/N} \frac{s^2 + 2R_n M^{-1/N} s + M^{-2/N}}{s^2 + 2R_n M^{1/N} s + M^{2/N}}$$

which is in agreement with the reciprocal cutoff preservation property of the Butterworth transformation. Thus we have obtained a serial chain of 2-pole tilting filters with numerator "cutoff" $M^{-1/N}$ and denominator "cutoff" $M^{1/N}$. The low- and high-shelving filters can be transformed similarly or obtained from the transformed tilting filter.

Wide-range slope

Let $H_2(s)$ be a 2-pole tilting filter. In the discussion of the 2-pole shelving filters we mentioned that such filter smoothly varies its response from the one of a 1-pole shelving filter $H_1(s)$ to $\mathcal{B}_2\left[H_1(s)\right]$ as R varies from $R_1 = (M + M^{-1})/2$ to

$R_2 = 1/\sqrt{2}$:

$$H_2(s)\Big|_{R=R_1} = H_1(s) \tag{10.17a}$$

$$H_2(s)\Big|_{R=R_2} = \mathcal{B}_2\left[H_1(s)\right] \tag{10.17b}$$

where the steepness of the amplitude response respectively doubles on that range.

Now let R initially be equal to R_1 and imagine we have smoothly decreased it to $R = R_2$. Suppose at this moment we swapped the 2-pole tilting filter $H_2(s)$ with a 4-pole tilting filter $H_4(s) = \mathcal{B}_2\left[H_2(s)\right]$ simultaneously resetting the damping back to $R = R_1$. Using (10.17) we have

$$\mathcal{B}_2\left[H_2(s)\Big|_{R=R_1}\right] = H_2(s)\Big|_{R=R_2}$$

and thus this swapping doesn't change the frequency response of the filter. Now we can vary R from R_1 to R_2 again to smoothly double the amplitude response once more.

So it seems we have found a way to vary the steepness of the tilting filter by a factor of 4 without getting the unwanted amplitude response artifacts which occur in a 2-pole tilting filter on an excessive range of R. However the problem is the swapping of $H_2(s)$ with $H_4(s)$ and back. In mathematical notation the swapping is seamless, because the transfer function doesn't change during the swap. In a real implementation however the swapping means replacing one filter structure with another and this will generate a transient, unless the internal states of the two filters are perfectly matched at the moment of the swapping.

Recall, however, that at $R = R_1$ the 2-pole $H_2(s)$ can be decomposed into two 1-poles, where the second of the 1-poles is fully transparent:

$$H_2(s)\Big|_{R=R_1} = H_1(s) \cdot \frac{s+1}{s+1}$$

Applying Butterworth transformation of order $N = 2$ to both sides we obtain

$$H_4(s)\Big|_{R=R_1} = \mathcal{B}_2\left[H_1(s)\right] \cdot \mathcal{B}_2\left[\frac{s+1}{s+1}\right] = H_2(s)\Big|_{R=R_2} \cdot \frac{s^2+\sqrt{2}s+1}{s^2+\sqrt{2}s+1}$$

This means that we could have a 4-pole filter

$$H_4(s) = H_{2a}(s) \cdot H_{2b}(s)$$

(where $H_{2a}(s)$ and $H_{2b}(s)$ are 2-pole sections) all the time. As long as we are interested in a 2-pole response $H_2(s)$ we let

$$H_{2a}(s) = H_2(s)$$

$$H_{2b}(s) = \frac{s^2+\sqrt{2}s+1}{s^2+\sqrt{2}s+1}$$

As we are switching from $H_2(s)\big|_{R=R_2}$ to $\mathcal{B}_2\left[H_2(s)\big|_{R=R_1}\right]$ neither of the sections $H_{2a}(s)$ $H_{2b}(s)$ is changed. From this point on we can further change R from R_1

to R_2 updating the coefficients of $H_{2a}(s)$ $H_{2b}(s)$ according to $H_{2a}(s) \cdot H_{2b}(s) = \mathcal{B}_2\left[H_2(s)\right]$.

This procedure can be repeated again, that is, having reached $R = R_2$ for the 4-pole shelving response, we can replace $H_4(s)$ by

$$H_8(s) = \mathcal{B}_2\left[H_4(s)\right] = \mathcal{B}_4\left[H_2(s)\right]$$

simultaneously resetting R to $R = R_1$. The idea is the same, we decompose $H_8(s)$ into

$$H_8(s) = H_{4a}(s) \cdot H_{4b}(s)$$

and we let

$$\begin{aligned} H_{4a}(s) &= H_4(s) \\ H_{4b}(s) &= \mathcal{B}_2\left[\frac{s^2 + \sqrt{2}s + 1}{s^2 + \sqrt{2}s + 1}\right] = \mathcal{B}_4\left[\frac{s+1}{s+1}\right] \end{aligned}$$

until the point of the switching from $H_4(s)$ to $H_8(s)$ where we start having

$$\begin{aligned} H_{4a}(s) &= \mathcal{B}_2\left[H_{2a}(s)\right] \\ H_{4b}(s) &= \mathcal{B}_2\left[H_{2b}(s)\right] \end{aligned}$$

Of course the same procedure can be further repeated as many times as desired (keeping in mind that the order of the resulting filter grows exponentially). Thus we can choose some power of 2 as a maximum desired filter order and switch this filter's response from $H_2(s)$ to $H_4(s)$ to $H_8(s)$ etc. each time R reaches R_2. The steepness of the amplitude response thereby smoothly varies by a factor equal to the filter's order.[8]

Another way of looking at this is noticing that, as the response steepness grows, we are traversing the responses defined by $H_1(s)$, $\mathcal{B}_2\left[H_1(s)\right]$, $\mathcal{B}_4\left[H_1(s)\right]$, etc. Therefore we can consider this as if it was a smooth variation of the Butterworth tilting filter's order.[9]

10.6 Band shelving

The 2-pole bandshelving filter can be easily obtained by applying the LP to BP substitution to the 1-pole low-shelving filter. Or, we can apply the LP to BP substitution to the 1-pole tilting filter (obtaining a kind of a "band-tilting" filter) and multiply the result by the necessary gain factor.

Let's do the latter. Given

$$H_{\text{tilt}}(s) = \frac{Ms + 1}{s + M} = \frac{s + 1/M}{s/M + 1}$$

[8] Note that the relative steepness κ of the amplitude response thereby provides a natural way to control the steepness variation, where $\kappa = k/k_1$, where k is the actual derivative of the amplitude response at the midpoint and k_1 is the same derivative for $H_1(s)$ (for the currently chosen tilting amount M).

[9] Unfortunately there is no 100% generalization of this process for lowpass, highpass or bandpass filters, since the rolloff of these filter types is fixed to an integer multiple of 6dB/oct and can't be varied in a smooth way.

we perform the substitution

$$s \leftarrow \frac{1}{2R}\left(s + s^{-1}\right)$$

obtaining

$$H(s) = \frac{\frac{1}{2R}\left(s + s^{-1}\right) + 1/M}{\frac{1}{2RM}\left(s + s^{-1}\right) + 1} = \frac{Ms + 2R + Ms^{-1}}{s + 2RM + s^{-1}} = \frac{Ms^2 + 2Rs + M}{s^2 + 2RMs + 1} =$$
$$= \frac{Ms^2 + M^{-1} \cdot 2RMs + M}{s^2 + 2RMs + 1} = MH_{\mathrm{LP}}(s) + M^{-1}H_{\mathrm{BP1}}(s) + MH_{\mathrm{HP}}(s)$$

where the SVF damping is equal to RM, where R is determined by the bandwidth of the LP to BP transformation. The obtained filter could be referred to as "band-tilting" filter (Fig. 10.11).

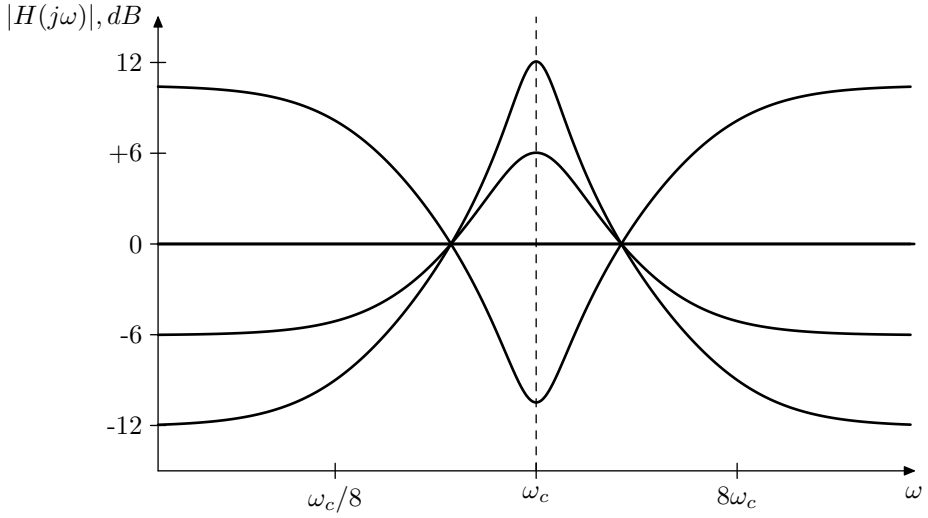


Figure 10.11: Amplitude response of 2-pole band-tilting filter for various M.

In order to turn this filter into the band-shelving filter, apparently, we have to divide the response by M:[10]

$$H(s) = \frac{s^2 + M^{-2} \cdot 2RMs + 1}{s^2 + 2RMs + 1} =$$
$$= H_{\mathrm{LP}}(s) + M^{-2}H_{\mathrm{BP1}}(s) + H_{\mathrm{HP}}(s) = 1 + (M^{-2} - 1)H_{\mathrm{BP1}}(s)$$

(mind that the SVF damping is still being equal to RM). Thus, the 2-pole band-shelving filter can be implemented by mixing the (normalized) bandpass signal to the input signal. The amplitude response at the cutoff is $H(j) = M^{-2}$ which thereby defines the shelving gain. The desired bandwidth of the shelving can be

[10] Alternatively, by multiplying the response by M we obtain a kind of "inverted band-shelving" filter, where the shelving bands are to the left and to the right of the passband in the middle.

achieved using the properties of the LP to BP substition, namely the formula (4.20).

It is interesting to observe that the above band-shelving transfer function can be rewritten as

$$H(s) = \frac{s^2 + 2RM^{-1}s + 1}{s^2 + 2RMs + 1} \tag{10.18}$$

that is we have a ratio of two filters with different dampings RM and RM^{-1}, where the filters themselves could be lowpass, highpass or bandpass (the important thing being that they have identical numerators, which then cancel each other).

Band shelving of higher orders

The band-shelving Butterworth filter of the 2nd kind is obtained by applying the Butterworth transformation to the 2-pole band-shelving filter (10.18):

$$H(s) = \frac{s^2 + 2RM^{-1}s + 1}{s^2 + 2RMs + 1}$$

Thus we have a ratio of two 2nd order polynomials both having unit cutoff but different damping. Applying the Butterworth transformation we therefore obtain a cascade of 2nd-order sections with unit cutoff:

$$H'(s) = \prod_n \frac{s^2 + 2R'_n s + 1}{s^2 + 2R_n s + 1}$$

Apparently each such 2nd-order section is a 2-pole bandshelving filter with the shelving boost and the bandwidth defined by the parametrs R_n and R'_n. Fig. 10.12 shows the amplitude response.

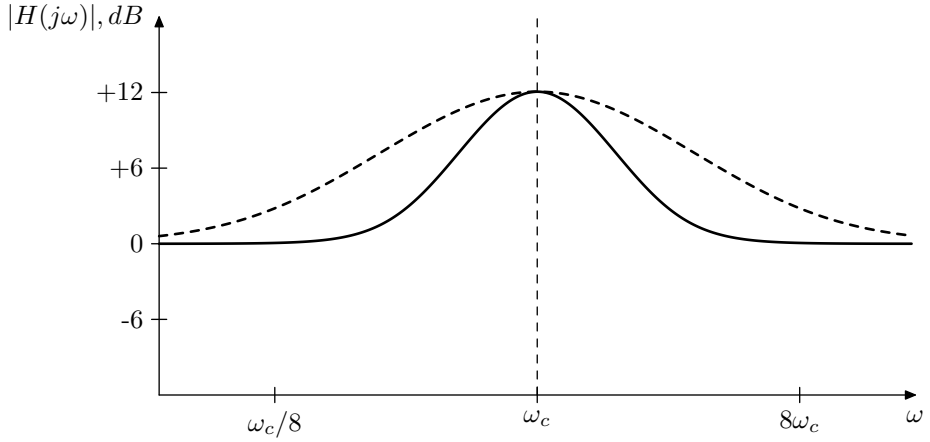


Figure 10.12: 4th order band-shelving Butterworth filter of the 2nd kind vs. 2nd order band-shelving filter (dashed line).

Another kind of band-shelving filter can be obtained by applying the LP to BP substitution to a Butterworth low-shelving filter of the 2nd kind. A useful feature of this approach is that by choosing the order of the low-shelving filter (and thus choosing the steepness of the low-shelving filter's slope) one can choose the steepness of the slopes of the band-shelving filter.

10.7 Elliptic shelving

We have mentioned that the mixing approach of (10.12) can be applied to EMQF filters. Of all equations (10.13) it's probably easiest to use (10.13a) to construct a high-shelving filter, the other filters can be derived from it in a trivial way. As (10.13a) is a monotonic mapping of the range $[0, +\infty]$ of $\bar{f}^2$ onto the range of $|H_{\mathrm{HS}}(j\omega)|$ contained between 1 and β^2, we are going to have $|H_{\mathrm{HS}}(j\omega)|$ smoothly varying from 1 to β^2 as $\bar{R}_N$ varies from 0 to ∞, just with some ripples in the pass and shelving bands.

Letting $\bar{f}(\omega) = \bar{R}_N(\omega)$ in (10.13a):

$$|H_{\mathrm{HS}}(j\omega)|^2 = \frac{1 + \beta^2 \bar{R}_N^2(\omega)}{1 + \beta^{-2} \bar{R}_N^2(\omega)} \tag{10.19}$$

we obtain the amplitude response shown in Fig. 10.13. Notice that due to the monotonic nature of the mapping (10.13a) the pass and shelving band ripples do not oscillate around the *reference gains* 1 and β^2 (corresponding to $\bar{R}_N = 0$ and $\bar{R}_N = \infty$, but are rather occurring "into the inside" of the range between 1 and β^2.

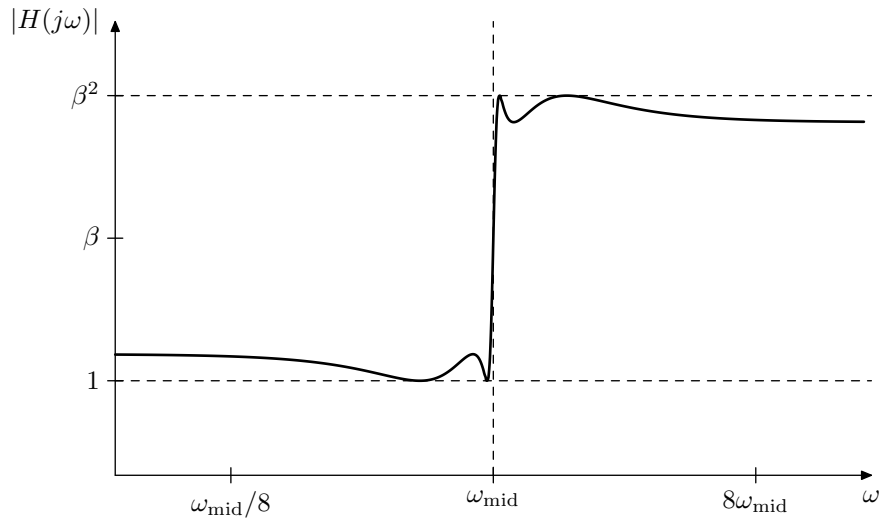


Figure 10.13: Amplitude response of an elliptic high-shelving filter. Horizontal dashed lines denote the reference gains 1 and β^2.

Implementation

From (10.19) we obtain the pole and zero equations for $H_{\mathrm{HS}}(s)$:

$$1 + \beta^{-2} \bar{R}_N^2(\omega) = 0 \tag{10.20a}$$

$$1 + \beta^{2} \bar{R}_N^2(\omega) = 0 \tag{10.20b}$$

where we ignore the poles of $\bar{R}_N$, since they cancel each other within $H_{\mathrm{HS}}(s)$ anyway. The equations (10.20) are essentially identical to (9.144), where we let

$\lambda = 1$ and $\varepsilon = \beta^{-1}$ or $\varepsilon = \beta$ respectively.[11]

Having obtained the poles and zeros we can define the leading gain coefficient g of the cascade form (8.1) from the requirement

$$H(0) = \sqrt{\frac{1 + \beta^2 \bar{R}_N^2(0)}{1 + \beta^{-2} \bar{R}_N^2(0)}} = \sqrt{\frac{1 + \beta^2 \left(\operatorname{Re} j^N\right)^2 \tilde{k}}{1 + \beta^{-2} \left(\operatorname{Re} j^N\right)^2 \tilde{k}}}$$

We could also use a simpler requirement:

$$|H(j)| = \beta$$

however we should mind the possibility of accidentally obtaining a 180° phase response at $\omega = 0$.

Control parameters

In order to compute the passband ($\omega \ll 1$) ripple amplitude, we can notice that in the passband the value of $\bar{R}^2(\omega)$ varies between 0 and $\tilde{k}$. By (10.19) the maximum deviation from the reference gain 1 will be at the gain equal to

$$\delta = \sqrt{\frac{1 + \beta^2 \tilde{k}}{1 + \beta^{-2} \tilde{k}}} \tag{10.21}$$

thus in the passband $|H_{\mathrm{HS}}(j\omega)|$ varies between 1 and δ. If $\beta > 1$ then $\delta > 1$ and vice versa. By the reciprocal symmetry (10.5) the deviation from the shelving band's reference gain β^2 is the same, just in the opposite direction, thus in the shelving band $|H_{\mathrm{HS}}(j\omega)|$ varies between β^2 and β^2/δ.

From (10.21) it's easy to notice that reciprocating β reciprocates δ and vice versa. Therefore without loss of generality, for the sake of simplicity we can restrict the discussion to $\beta \geq 1$, $\delta \geq 1$, in which case the passband ripples occcur within $[1, \delta]$ and the shelving band ripples occur within $[\beta^2/\delta, \beta^2]$. The case of $\beta \leq 1$, $\delta \leq 1$ will follow automatically, where the ripple ranges will be $[\delta, 1]$ and $[\beta^2, \beta^2/\delta]$ respectively.

Recall that the value of $\tilde{k}$ grows simultaneously with k, where the latter is defining the elliptic transition band $[\sqrt{k}, 1/\sqrt{k}]$. Thus we are having three user-facing parameters, each of those being *independently* related to its respective variable:[12]

Transition bandwidth:	$\tilde{k}$
Shelving gain:	β
Ripple amplitude:	δ

where the dependency between the three variables is given by (10.21).

Apparently at fixed $\tilde{k} > 0$ the value of δ grows with β and vice versa. At fixed $\beta > 1$, the value of δ grows with $\tilde{k}$ and vice versa. At fixed $\delta > 1$, the

[11] Thus, differently from how we used (9.144) in the discussion of elliptic lowpass, we treat ε and λ now as independent variables. This doesn't affect the solution process of (9.144), since the respective transformations didn't use the interdependency of ε and λ.

[12] We should mind that the dependency between the transition bandwidth and $\tilde{k}$ is reciprocal-like: larger $\tilde{k}$ means smaller bandwidth. The other two dependencies are straightforward.

values of β and k change in opposite directions. Thus, if e.g. we want a smaller transition band, this means we want larger k and larger $\tilde{k}$, which means larger δ (given a fixed β). This means there is a tradeoff between the transition band width (which we usually want small) and the ripple amplitude (which we also usually want small). There are similar tradeoffs between the other two pairs of the user-facing parameters.

Given any two of the three parameters, we can find the third one from (10.21). The explicit expressions for β and $\tilde{k}$ can be obtained by transforming (10.21) to

$$1 + \beta^2\tilde{k} = \delta^2 + \beta^{-2}\tilde{k}\delta^2$$

$$\beta^2 + \beta^4\tilde{k} = \beta^2\delta^2 + \tilde{k}\delta^2$$

from where on one hand

$$\beta^4\tilde{k} - (\delta^2 - 1)\beta^2 - \tilde{k}\delta^2 = 0$$

$$\beta^2 = \frac{\delta^2 - 1}{2\tilde{k}} + \sqrt{\left(\frac{\delta^2 - 1}{2\tilde{k}}\right)^2 + \delta^2} \tag{10.22}$$

(where apparently the restriction is $\delta \geq 1$), on the other hand

$$(\beta^4 - \delta^2)\tilde{k} = \beta^2(\delta^2 - 1)$$

$$\tilde{k} = \beta^2\frac{\delta^2 - 1}{\beta^4 - \delta^2} \tag{10.23}$$

(where $1 \leq \delta < \beta$ will ensure $0 < \tilde{k} < 1$) and k can be obtained by (9.133).

At $\beta = 1$ the formula (10.23) doesn't work, since the amplitude response of H_{HS} is simply a horizontal line at unity gain and any of the values of $\tilde{k}$ will do. Respectively at $\tilde{k} = 0$ the formula (10.22) doesn't work, since δ must be equal to 1 in this case.

Notably, since (10.21) works equally well for $\beta < 1$ and $\delta < 1$, so do (10.22) (under the restriction $\delta \leq 1$) and (10.23) (under the restriction $\beta < \delta \leq 1$). In practice the numerical evaluation of (10.22) for $\delta < 1$ could raise concerns of potential precision losses, therefore it's better to apply (10.22) to the reciprocal value of δ, which is larger than 1, and then reciprocate the result once again.

Centered ripples

If we allow equiripples in the pass and shelving bands, it would be reasonable to require that these equiripples are not unipolar but rather centered around the required reference gains of these bands. We are going now to derive the respective formulas, which will be slightly simpler to do in terms of the tilting filter:

$$|H_{\text{tilt}}(j\omega)|^2 = \beta^{-2}\frac{1 + \beta^2\bar{R}_N^2(\omega)}{1 + \beta^{-2}\bar{R}_N^2(\omega)}$$

where again, for simplicity of discussion, without loss of generality we will assume $\beta \geq 1$, $\delta \geq 1$.

As a first step, we shall define the new reference gains, corresponding to the logarithmic centers of the equiripples. Since the left shelving band ripples occur

within $[\beta^{-1}, \beta^{-1}\delta]$, their logarithmic center is at $\beta^{-1}\sqrt{\delta}$. Similarly, since the right shelving band ripples occur within $[\beta/\delta, \beta]$, their logarithmic center is at $\beta/\sqrt{\delta}$. Therefore we introduce $\tilde{\beta} = \beta/\sqrt{\delta}$ and the new reference gains $\tilde{\beta}^{-1}$ and $\tilde{\beta}$ at the logarithmic centers of the equiripple ranges (Fig. 10.14).

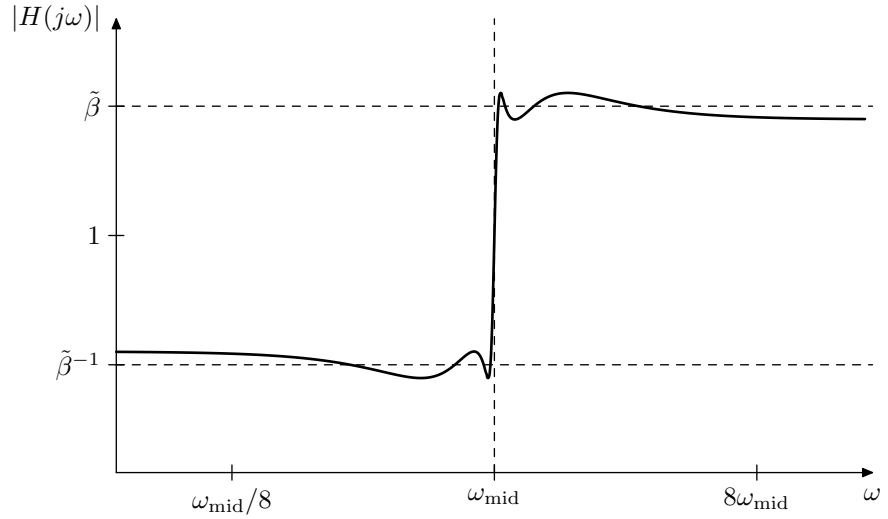


Figure 10.14: Centered reference gains $\tilde{\beta}^{-1}$ and $\tilde{\beta}$ of an elliptic tilting filter.

We want to use $\tilde{\beta}$ instead of β as one of the three control parameters. Since the entire framework of elliptic shelving filters has been developed in terms of k, β and δ, we'll need to be able to convert from $\tilde{\beta}$ to β. At the first sight this seems to be trivially done by $\beta = \tilde{\beta}\sqrt{\delta}$, however this can be done only if we know δ.

Recall that we have three control parameters $\tilde{k}$, β and δ but only two freedom degrees, which means that we can specify only two of the three parameters, while the third parameter needs to be found by the respective relations. Similarly, if the three control parameters are now $\tilde{k}$, $\tilde{\beta}$ and δ, we are going to specify only two of them. So, if we specify $\tilde{\beta}$ and δ, then indeed we can simply find $\beta = \tilde{\beta}\sqrt{\delta}$ and then find $\tilde{k}$ by (10.23). Specifying $\tilde{k}$ and δ is apparently the same as before and doesn't pose any new probleems. However there is yet an option of specifying $\tilde{k}$ and $\tilde{\beta}$, in which case we need to find either β or δ, so that the other variable can be found from (10.22) or (10.21).

Let's find β. Substituting the equation (10.21) into $\beta = \tilde{\beta}\sqrt{\delta}$ we obtain

$$\beta^4 = \tilde{\beta}^4 \cdot \frac{1 + \beta^2\tilde{k}}{1 + \beta^{-2}\tilde{k}}$$

$$\beta^4 + \tilde{k}\beta^2 = \tilde{\beta}^4 + \tilde{k}\tilde{\beta}^4\beta^2$$

$$\beta^4 - \tilde{k}(\tilde{\beta}^4 - 1)\beta^2 - \tilde{\beta}^4 = 0$$

$$\beta^2 = \tilde{k}\frac{\tilde{\beta}^4 - 1}{2} + \sqrt{\left(\tilde{k}\frac{\tilde{\beta}^4 - 1}{2}\right)^2 + \tilde{\beta}^4} \tag{10.24}$$

Similarly to (10.22), formula (10.24) also works for $\beta \le 1$, $\delta \le 1$, but due to numeric reasons in this case it's better to apply it to the reciprocal δ and then reciprocate the result.

Thus we have developed a way to express the tilting filter in terms of the centered reference gains β^{-1} and β by converting from $\tilde{\beta}$ to β either by $\beta = \tilde{\beta}\sqrt{\delta}$ or by (10.24). For the high- and low-shelving filters one needs to additionally take into account that the new reference gains imply different multiplication factors for conversion from tilting to the respective shelving factors:

$$H_{\text{HS}}(s) = \tilde{\beta} \cdot H_{\text{tilt}}(s)$$
$$H_{\text{LS}}(s) = \tilde{\beta}^{-1} \cdot H_{\text{tilt}}(s)$$

so that the centered passband reference gain is at 1 and the centered shelving reference gain is at $\tilde{\beta}^2$ or $\tilde{\beta}^{-1}$ respectively.

Relation to Butterworth shelving

At $k = 0$ we have $\bar{R}_N(x) = x^N$ and the elliptic shelving filters turn into respective 1st-kind Butterworth shelving filters. However also notice that a 2nd-kind Butterworth shelving filter of order N at $R = 1/\sqrt{2}$ is equal to the 1st-kind Butteworth shelving of the same order, which in turn is equal to the elliptic shelving filter of the same order at $k = 0$. That is at $R = 1/\sqrt{2}$ and $k = 0$ all three kinds of shelving filters coincide.

In that sense elliptic shelving can be seen as another way of extending the 2nd-kind Butterworth variable-slope shelving into the range beyond $R = 1/\sqrt{2}$. Instead of reducing R below $1/\sqrt{2}$ (which would result in one large resonance peak in each of the pass and/or shelving bands), we could switch to the elliptic filter parameters[13] obtaining a number of smaller equiripples. This particularly means that in the wide-range slope technique described in Section 10.5 instead of increasing the filter order at $R = 1/\sqrt{2}$ we could switch to elliptic equations (if we are willing to accept the ripples).

At $N = 2$ however there is essentially no difference between 2nd-kind Butterworth shelving at $R \le 1/\sqrt{2}$ and elliptic shelving, except for different formal control parameters. Indeed, both filters have two poles and two zeros which are mutually reciprocal and are also having conjugate symmetry. This leaves only two degrees of freedom, one degree corresponds to choosing the cutoff of the poles (which simultaneously defines the cutoff of the zeros as the reciprocal value of the poles cutoff) the other degree being the damping of the poles (which simultaneously defines the damping of the zeros as both dampings must be equal). However the first degree of freedom is taken by controlling the shelving gain and the second degree of freedom is taken by varying the R or the k parameter respectively. Thus the difference between the two filters can be only in how the control parameters are translated to the transfer function and

[13] Notice that such switching is completely smooth, as the transfer functions of the filters are completely identical at this point and the "physical" orders of the filters are identical too (there is no pole/zero cancellation as we had in the Butterworth filter order switching). Strictly speaking, the statement that the transfer functions are identical holds only under the restriction that the phase responses of the filters are in sync, rather that 180° off, which is a matter of the sign in front of the transfer functions. However usually we are having zero phase responses at $\omega = 0$, therefore the signs will be automatically matched.

in the leading gain coefficients of the transfer functions (where we would have $|H_{\mathrm{HS}}(0)| = 1$ in the Butterworth case and $|H_{\mathrm{HS}}(0)| = \delta$ in the elliptic case).

Combining with other techniques

Elliptic design of shelving filters can be combined with other design techniques. Particulary, we can apply the LP to BP transformation to an elliptic low-shelving filter to obtain an elliptic band-shelving filter.

In principle one also could apply Butterworth transformation to elliptic filters to increase the slope steepness. However, since we are already having ripples in the pass and shelving bands, it would be more efficient to simply increase the order of elliptic filter, thereby attaining higher slope steepnesses (compared to applying the Butterworth transformation) at the same ripple amplitude.

10.8 Crossovers

Sometimes we would like to process different frequency bands of a signal differently. In the simplest case we would want to split the signal into low- and high-frequency parts, process one of them or both in some way and then merge them back (Fig. 10.15). This kind of filters, splitting the signal into different frequency bands are called *crossovers*.

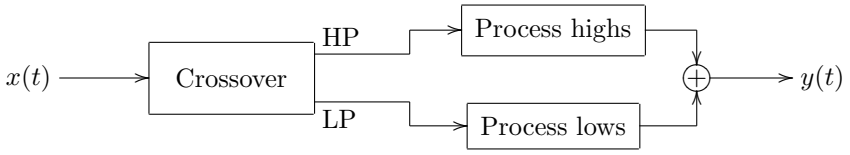


Figure 10.15: The crosssover idea.

In principle we could take any a pair of lowpass and highpass filters to build a crossover, but some combinations would work better than the others. Particularly, imagine that the processing of different bands changes the signals very slightly, or sometimes maybe even doesn't change them at all. In that case it would be really nice if the original signal was unaltered by the structure in Fig. 10.15, that is $y(t) = x(t)$. However, this naturally expected property will not be given for granted.

Suppose we use a multimode 1-pole as a crossover basis, in which case the low- and high-pass filters share the same cutoff. Without loss of generality we could let $\omega_c = 1$ (in other words, the *crossover frequency* will be at $\omega = 1$):

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

Adding low- and high-pass transfer functions we have

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

Thus, if the low- and high-pass signals are unmodified by the processing, adding them together at the end of the network in Fig. 10.15 would restore the original signal exactly.

However the same doesn't hold anymore for 2-poles:

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

in which case we have

$$H(s) = H_{\mathrm{LP}}(s) + H_{\mathrm{HP}}(s) = \frac{s^2+1}{s^2+2Rs+1} \neq 1$$

In fact, as we may recall, the above $H(s)$ is a notch filter. Of course, we could add the missing bandpass component, e.g. splitting it equally between the low- and high-bands:

$$\frac{1+Rs}{1+2Rs+s^2} + \frac{s^2+Rs}{1+2Rs+s^2} = 1$$

but then the rolloff of the resulting low- and high-pass filters becomes 6dB/oct instead of former 12dB/oct, leading to the question, why using such 2-pole in the first place when a 1-pole would have done similarly.

Butterworth crossovers

If we relax the requirement of the sum of unprocessed signals being exactly equal to the original signal and allow a phase shift in the sum, while retaining the amplitudes, we essentially require that the low- and high-passes should add to an allpass:

$$|H(s)| = |H_{\mathrm{LP}}(s) + H_{\mathrm{HP}}(s)| = 1$$

Let's take (1st kind) Butterworth low- and high-passes at the same cutoff $\omega_c = 1$:

$$H_{\mathrm{LP}}(s) = \frac{1}{P(s)}$$
$$H_{\mathrm{HP}}(s) = \frac{s^N}{P(s)}$$

where N is the filter order and $P(s)$ denotes the common denominator of $H_{\mathrm{LP}}(s)$ and $H_{\mathrm{HP}}(s)$. Remember that the denominator $P(s)$ is defined by the equation

$$|P(j\omega)|^2 = 1 + \omega^{2N}$$

while all roots of $P(s)$ must lie in the left complex semiplane.

Adding the low- and high-passes together we obtain

$$H(s) = H_{\mathrm{LP}}(s) + H_{\mathrm{HP}}(s) = \frac{1}{P(s)} + \frac{s^N}{P(s)} = \frac{1+s^N}{P(s)}$$

Assuming N is odd, for $s = j\omega$ we get

$$|H(j\omega)|^2 = \left|\frac{1 + (j\omega)^N}{P(j\omega)}\right|^2 = \frac{1 + \omega^{2N}}{|P(j\omega)|^2} = 1 \qquad (N \text{ odd})$$

Notably, the same property holds for the difference of Butterworth low- and high-passes of odd order

$$|H_{\text{LP}}(s) - H_{\text{HP}}(s)|^2 = \left|\frac{1 - (j\omega)^N}{P(j\omega)}\right|^2 = \frac{1 + \omega^{2N}}{|P(j\omega)|^2} = 1 \qquad (N \text{ odd})$$

For an even order however $1 \pm s^N$ becomes purely real for $s = j\omega$

$$1 \pm s^N = 1 \pm j^N\omega^N = 1 \pm (-1)^{N/2}\omega^N$$

and we get either a zero at $\omega = 1$ if the above gets the form $1 - \omega^N$, or, if it gets the form $1 + \omega^N$ then

$$|H(j\omega)|^2 = \left|\frac{1 + (j\omega)^N}{P(j\omega)}\right|^2 = \frac{\left(1 + \omega^N\right)^2}{1 + \omega^{2N}} = \frac{1 + 2\omega^N + \omega^{2N}}{1 + \omega^{2N}} =$$
$$= 1 + \frac{2\omega^N}{1 + \omega^{2N}} = 1 + \left(\frac{1 + \omega^{2N}}{2\omega^N}\right)^{-1} = 1 + 2\left(\omega^N + \frac{1}{\omega^N}\right)^{-1}$$

Apparently the expression in parentheses is symmetric in logarithmic scale around $\omega = 1$ and attains a minumum at this point, respectively $|H(j\omega)|^2$ attains a maximum, which we can evaluate by substituting $\omega = 1$, obtaining $|H(j)|^2 = 2$. Respectively $|H(j)| = \sqrt{2}$ thus the amplitude response of $H(s)$ has a +3dB bump at $\omega = 1$.

As both $H_{\text{LP}}(s)$ and $H_{\text{HP}}(s)$ share the same denominator, they can be implemented by a single generalized SVF (the controllable canonical form in Fig. 8.1) using the modal outputs for the numerators 1 and s^N respectively. Alternatively one could use multimode features of the serial cascade representation. Parallel representation is also possible, where we would pick up different modal mixtures of the same parallel 2-poles as the low- and high-pass signal respectively.

Linkwitz–Riley crossovers

If instead of Butterworth lowpass and highpass filters we take squared Butterworth filters:

$$H(s) = H_1(s) + H_2(s)$$
$$H_1(s) = H_{\text{LP}}^2(s) = \left(\frac{1}{P(s)}\right)^2$$
$$H_2(s) = (-1)^N H_{\text{HP}}^2(s) = (-1)^N \left(\frac{s^N}{P(s)}\right)^2$$
$$|P(j\omega)|^2 = 1 + \omega^{2N}$$

(notice the conditional inversion of the squared highpass signal) we do obtain a perfect allpass $H(s)$ for any N:

$$H(j\omega) = H_1(j\omega) + H_2(j\omega) = \frac{1}{P^2(j\omega)} + (-1)^N \frac{(j\omega)^{2N}}{P^2(j\omega)} =$$

$$= \frac{1 + (-1)^N j^{2N} \omega^{2N}}{P^2(j\omega)} = \frac{1 + \omega^{2N}}{P^2(j\omega)} = \frac{P(j\omega)P(-j\omega)}{P^2(j\omega)} = \frac{P(-j\omega)}{P(j\omega)} \quad (10.25)$$

and thus, since $P(j\omega)$ is Hermitian, $|H(j\omega)| = 1$. A crossover designed in this way is referred to as *Linkwitz–Riley crossover*. Since the denominators in (10.25) are identical, we again can use a shared structure, such as a generalized SVF or a multimode serial cascase to produce the output signals of both H_1 and H_2. The parallel representation is problematic, since we are now having repeated poles due to the squaring of the denominators.[14]

Note that the phase responses of $H_1(s)$ and $H_2(s)$ are identical, since the phase contributions of their numerators are zero, while the phase contributions of their denominators are identical. This in-phase relationship of the split bands is the key feature of Linkwitz–Riley crossovers[15] (contrary to the somewhat common opinion that the key feature of Linkwitz–Riley crossovers is the absence of the +3dB bump, which, as we have seen is also e.g. the case with odd-order Butterworth crossovers).

The in-phase relationship actually also includes $H(s)$:

$$\arg H_1(j\omega) = \arg H_2(j\omega) = \arg H(j\omega)$$

Indeed

$$\arg H(j\omega) = \arg P(-j\omega) - \arg P(j\omega) = -2 \arg P(j\omega) = \arg \frac{1}{P^2(j\omega)}$$

where we used the Hermitian property of $P(j\omega)$.

Generalized Linkwitz–Riley crossovers

The Linkwitz–Riley design consisting of two squared Butterworth filters is a special case of a more generic idea which we will discuss below.[16]

First we need a kind of auxiliary lemma. Let $Q(s)$ be a real polynomial of s. We now state that the formal frequency response $Q(j\omega)$ is real nonnegative if and only if $Q(s)$ can be written in the form $Q(s) = P(s)P(-s)$, where $P(s)$ is some other real polynomial of s. The proof goes like follows.

Suppose $Q(s) = P(s)P(-s)$. Then for $\omega \in \mathbb{R}$

$$Q(j\omega) = P(j\omega)P(-j\omega) = P(j\omega)P\left((j\omega)^*\right) = P(j\omega)P^*(j\omega) = |P(j\omega)|^2 \geq 0$$

Conversely, suppose $Q(j\omega) \geq 0$. Since $Q(j\omega)$ is simultaneously real and Hermitian, it must be even, and so must be $Q(s)$. Therefore it can be factored into $Q(s) = P(s)P(-s)$. Let's chose $P(s)$ to contain the left complex semiplane

[14] In principle, by general considerations, one should be able to connect two identical parallel representations in series and obtain modal mixtures from those in a fashion similar to the multimode serial cascade. The author however didn't verify the feasibility of this approach.

[15] The author has been made aware of the importance of the in-phase property of Linkwitz–Riley crossovers by a remark by Teemu Voipio. The author also learned the cascaded phase correction approach shown in Fig. 10.24 from the same person.

[16] The idea to generalize the Linkwitz–Riley design arose from a remark by Max Mikhailov, that Linkwitz–Riley crossovers can be also built based on naive 1-pole lowpasses, which in the BLT terms can be formally seen as a special kind of high-shelving filters. It is quite possible that this idea has been already developed elsewhere, however at the time of the writing the author is not aware of other sources.

roots of $Q(s)$, thereby $P(-s)$ will contain the right complex semiplane roots. If $Q(s)$ has roots on the imaginary axis, these roots will all have even multiplicities (since otherwise $Q(j\omega)$ will be changing sign at these points) and therefore we can split these roots into two identical halves, which we assign to $P(s)$ and $P(-s)$ respectively. Since $Q(s)$ is real, its poles are conjugate symmetric and so will be the poles of $P(s)$ and $P(-s)$.

We still need to show that the leading coefficient of $P(s)$ will be real. Let g denote the leading coefficient of P. Then the leading term of $Q(s) = P(s)P(-s)$ is $gs^N g(-s)^N = (-1)^N g^2 s^{2N}$. By substituting $s = j\omega$ we obtain the leading term of $Q(j\omega)$, which is $(-1)^N g^2 (j\omega)^{2N} = (-1)^N g^2 (-1)^N \omega^{2N} = g^2 \omega^{2N}$. However the coefficient g^2 of the leading term $g^2\omega^{2N}$ of $Q(j\omega)$ must be positive, otherwise $Q(j\omega)$ would become negative at large ω. Since g^2 is positive, g is real. That completes the proof.

Now, given two real polynomials $Q_1(s)$ and $Q_2(s)$ with real nonnegative frequency responses, we can construct a third real polynomial as their sum

$$Q_1(s) + Q_2(s) = Q(s) \tag{10.26}$$

Since the frequency responses of $Q_1(s)$ and $Q_2(s)$ are nonnegative, so is the frequency response of $Q(s)$. By the previous discussion, the above equation can be rewritten as

$$P_1(s)P_1(-s) + P_2(s)P_2(-s) = P(s)P(-s) \tag{10.27}$$

Dividing both sides by the right-hand side, we obtain

$$\frac{P_1(s)P_1(-s)}{P(s)P(-s)} + \frac{P_2(s)P_2(-s)}{P(s)P(-s)} = 1 \tag{10.28}$$

We wish to interpret the two terms in the left-hand side as transfer functions. However, these functions are not stable, since the roots of $P(-s)$ are lying in the right semiplane. We can however multiply both parts by $P(-s)/P(s)$:

$$\frac{P_1(s)P_1(-s)}{P^2(s)} + \frac{P_2(s)P_2(-s)}{P^2(s)} = \frac{P(-s)}{P(s)} \tag{10.29}$$

thereby making both filters stable and turning the right-hand side into an (also stable) allpass. Also notice that the orders of $P_1(s)$ and $P_2(s)$ do not exceed the order of $P(s)$, therefore the terms of (10.29) are nonstrictly proper rational functions of s, as required for transfer functions of (integrator-based) differential filters. Introducing

$$H_1(s) = \frac{P_1(s)}{P(s)} \cdot \frac{P_1(-s)}{P(s)} \tag{10.30a}$$

$$H_2(s) = \frac{P_2(s)}{P(s)} \cdot \frac{P_2(-s)}{P(s)} \tag{10.30b}$$

$$H_{\mathrm{AP}}(s) = \frac{P(-s)}{P(s)} \tag{10.30c}$$

we rewrite (10.29) as

$$H_1(s) + H_2(s) = H_{\mathrm{AP}}(s) \tag{10.31}$$

and thus we have built a crossover (provided $H_1(s)$ is a kind of a lowpass and $H_2(s)$ is a kind of a highpass). Again, the denominators are identical and we can use a shared structure for H_1 and H_2.

Notice that (10.31) is simply (10.26) divided by $P^2(s)$. The phase responses of all terms of (10.26) are apparently zero, therefore the phase responses of all terms of (10.31) are identical and simply equal to $-2\arg P(j\omega)$:

$$\arg H_1(j\omega) = \arg H_2(j\omega) = \arg H(j\omega) = -2\arg P(j\omega)$$

The identical phase responses, as we should remember, are the key feature of Linkwitz–Riley crossover design, thus we have built a kind of generalized Linkwitz–Riley crossover.

The identical phase responses of H_1, H_2 and H_{AP} also allow to rewrite (10.31) in terms of amplitude responses:

$$|H_1(s)| + |H_2(s)| = 1 \tag{10.32}$$

It is often convenient to define

$$G_1(s) = \frac{P_1(s)}{P(s)}$$
$$G_2(s) = \frac{P_2(s)}{P(s)}$$

By (10.30), $H_n(s) = G_n(s)G_n^-(s)$ where $G_n^-(s) = P_n(-s)/P(s)$. Apparently $|H_n(j\omega)| = |G_n(j\omega)|^2$ and therefore (10.32) turns into

$$|G_1(j\omega)|^2 + |G_2(j\omega)|^2 = 1 \tag{10.33}$$

The interpretation in terms of G_1 and G_2 suggests another, somewhat more practical approach to building generalized Linkwitz–Riley crossovers. We start with a pretty much random filter $G_1(s) = P_1(s)/P(s)$, although satisfying $|G_1(j\omega)|^2 \le 1$, so that (10.33) can hold. From $P_1(s)$ and $P(s)$, using (10.30), we obtain $H_1(s)$ and $H_{\mathrm{AP}}(s)$ and can simply find $H_2(s)$ as $H_2(s) = H_{\mathrm{AP}}(s) - H_1(s)$. In principle, the obtained $H_2(s)$ can be used as it is, but we can also further factor it into $H_2(s) = P_2(s)P_2(-s)/P^2(s)$ thereby obtaining $G_2(s)$.[17] Of course, in order for $H_1(s)$ and $H_2(s)$ to count as a "reasonable" crossover, $H_1(s)$ must be a lowpass or lowpass-like filter (which can be ensured by choosing a lowpass-like $G_1(s)$), and the obtained $H_2(s)$ must be highpass-like. Or the other way around.

Alternatively we might be able to simply "guess" $H_1(s)$ and $H_2(s)$ (or, equivalently, $P_1(s)$ and $P_2(s)$, or $G_1(s)$ and $G_2(s)$). E.g. the previously discussed Butterworth filter-based Linkwitz–Riley crossover arises by choosing $G_1(s)$ to be a Butterworth lowpass and $G_2(s) = G_1(1/s)$ to be a Butterworth highpass, which gives $G_1^-(s) = G_1(s)$, $G_2^-(s) = (-1)^N G_2(s)$ and respectively $H_1(s) = G_1^2(s)$, $H_2(s) = (-1)^N G_2^2(s)$. The same result is obtained by by choosing $P_1(s) = 1$,

[17] In order to show that this factoring is possible, multiply $H_2(j\omega) = H_{\mathrm{AP}}(j\omega) - H_1(j\omega)$ by $P^2(j\omega)$, obtaining $H_2(j\omega)P^2(j\omega) = P(j\omega)P(-j\omega) - P_1(j\omega)P_1(-j\omega) \ge 0$, where the latter inequality follows from $|G_1(j\omega)|^2 \le 1$.

$P_2(s) = s^N$ (respectively $P_1(-s) = 1$ and $P_2(-s) = (-1)^N s^N$). This gives (10.27) in the form

$$P(s)P(-s) = P_1(s)P_1(-s) + P_2(s)P_2(-s) = 1 + (-1)^N s^{2N}$$

from where

$$Q(j\omega) = 1 + \omega^{2N} = |P(j\omega)|^2 = P(j\omega)P(-j\omega)$$

where $P(s)$ is the Butterworth denominator. The equation (10.29) respectively takes the form

$$\left(\frac{1}{P(s)}\right)^2 + (-1)^N \left(\frac{s^N}{P(s)}\right)^2 = \frac{P(-s)}{P(s)}$$

which is essentially the same as (10.25).

Symmetric generalized Linkwitz–Riley crossovers

Ideally in (10.31) we would like to have symmetric amplitude responses

$$|H_2(j\omega)| = |H_1(j/\omega)| \tag{10.34}$$

as it was e.g. the case with Butterworth-based Linkwitz–Riley crossover. Apparently (10.34) is not guaranteed for an arbitrary pair of $H_1(s)$ and $H_2(s)$ which satisfies (10.31) (where satisfying (10.31) is understood in the sense that the sum of $H_1(s)$ and $H_2(s)$ is an allpass). We would like to find a way of obtaining generalized Linkwitz–Riley crossovers satisfying (10.34).

Recall that (10.33) is just another intepretation of the crossover equation (10.31). On the other hand, compare (10.33) to (10.4). By (10.4), the equation (10.33) will be satisfied by G_1 and G_2 related through an LP to HP transformation, if $f(1/x) = 1/f(x)$, where $f(x)$ is the function used to construct G_1 by (9.18).

This is not sufficient yet, as besides satisfying (10.33) (and respectively (10.31)), we need to have the same poles in $G_1(s)$ and $G_2(s)$, so that they can share the same denominator $P(s)$. However we have already shown that $G_1(s)$ and $G_2(s)$ will have the same poles if $f(1/x) = 1/f(x)$.

Therefore, in order to obtain a generalized Linkwitz–Riley crossover with symmetric amplitude responses, we need to take $G_1(s)$ obtained from $f(x)$ satisfying $f(1/x) = 1/f(x)$ and $G_2(s) = G_1(1/s)$.

EMQF Linkwitz–Riley crossovers

We already know one function $f(\omega)$ satisfying $f(1/x) = 1/f(x)$: the normalized elliptic rational function $\bar{R}_N$. Therefore EMQF filters might be a good candidate for symmetric generalized Linkwitz–Riley crossovers. Notice that at $k \to 0$ EMQF filters turn into Butterworth filters and we obtain a classical (Butterworth-based) Linkwitz–Riley crossover.

Therefore let $G_1(s) = P_1(s)/P(s)$ be an EMQF (lowpass) filter and $G_2(s) = G_1(1/s) = P_2(s)/P(s)$ be the respective highpass. Recall that the zeros of elliptic lowpass filters are positioned on the imaginary axis in pairs symmetric relatively to the origin, with the exception of the zero at the infinity, which occurs if the order N of the filter is odd. Therefore $P_1(s)$ can be written as

$$P_1(s) = g_1 \cdot \prod_{\operatorname{Im} z_n > 0} (s^2 - z_n^2)$$

which means that $P_1(-s) = (-1)^N P_1(s)$. Respectively $P_2(s)$ can be written as

$$P_2(s) = g_2 \cdot s^{N\wedge 1} \prod_{\operatorname{Im} z_n > 0} (s^2 - 1/z_n^2)$$

where the $s^{N\wedge 1}$ factor arises from the zero of $G_1(s)$ occuring at the infinity which turns into a zero of $G_2(s)$ occurring at the origin. Therefore $P_2(-s) = (-1)^N P_2(s)$.

Therefore $H_1(s) = G_1^2(s)$, $H_2(s) = (-1)^N G_2^2(s)$ and (10.31) takes the form

$$G_1^2(s) + (-1)^N G_2^2(s) = \frac{P(-s)}{P(s)}$$

Fig. 10.16 shows the example of amplitude responses of H_1 and H_2.

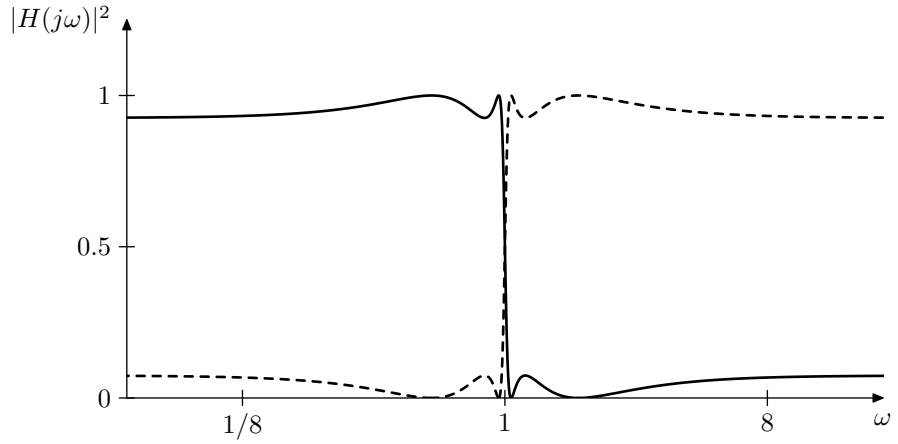


Figure 10.16: Amplitude responses of EMQF crossover low- (solid) and high-pass (dashed) outputs.

Centered ripples

Next we will describe a way to further improve the amplitude response of generalized Linkwitz–Riley crossovers. The techniques can be applied to pretty much any generalized Linkwitz–Riley crossover, but for the sake of simpler presentation we'll be using the EMQF crossover as an example.

Recall that the phase responses of H_1, H_2 and H_{AP} are identical. Let $\varphi(\omega) = \arg H_1(j\omega) = \arg H_2(j\omega) = \arg H_{\mathrm{AP}}(j\omega)$ be this common phase response. Then we can introduce the zero phase frequency response functions

$$\bar{H}_1(j\omega) = e^{-j\varphi(\omega)} H_1(j\omega)$$
$$\bar{H}_2(j\omega) = e^{-j\varphi(\omega)} H_2(j\omega)$$
$$\bar{H}_{\mathrm{AP}}(j\omega) = e^{-j\varphi(\omega)} H_{\mathrm{AP}}(j\omega) \equiv 1$$

Notice that since $\arg \bar{H}_1(j\omega) = \arg \bar{H}_2(j\omega) = \arg \bar{H}_{\mathrm{AP}}(j\omega) = 0$, we have

$$\bar{H}_1(j\omega) = |H_1(j\omega)|$$

$$\bar{H}_2(j\omega) = |H_2(j\omega)|$$
$$\bar{H}_{\mathrm{AP}}(j\omega) = |H_{\mathrm{AP}}(j\omega)| = 1$$

That is we can consider $\bar{H}_1(j\omega)$, $\bar{H}_2(j\omega)$ and $\bar{H}_{\mathrm{AP}}(j\omega)$ as amplitude response functions.

We are now going to construct some linear combinations of the above zero phase frequency responses. Since they are all related to the original frequency responses via one and the same factor $e^{-j\varphi(\omega)}$, linear combinations of $\bar{H}_1$, $\bar{H}_2$ and $\bar{H}_{\mathrm{AP}}$ correspond to exactly the same linear combinations of H_1, H_2 and H_{AP}. E.g.

$$\alpha\bar{H}_1(j\omega) + \beta\bar{H}_2(j\omega) = e^{-j\varphi(\omega)} \cdot (\alpha H_1(j\omega) + \beta H_2(j\omega))$$

We can think of these linear combinations as of linear combinations of amplitude responses, resulting in the new amplitude responses, with the reservation that the new "amplitude responses" may become negative (which in terms of true amplitude responses would have been interpreted as changing the phase response by 180°).

Consider that the passband ripple amplitude of $\bar{R}_N$ is $\sqrt{\tilde{k}}$, while the stopband ripple amplitude is $1/\sqrt{\tilde{k}}$. Respectively the passband ripples of $\bar{H}_1$ and $\bar{H}_2$ oscillate within $[1/(1+\tilde{k}), 1]$, while the stopband ripples oscillate within $[0, \tilde{k}/(1+\tilde{k})]$, which corresponds to the absolute maximum deviations $\tilde{k}/(1+\tilde{k})$ from the ideal values of 1 (passband) and 0 (stopband).

Note that so far the deviations are unipolar. The deviation from 1 occurs towards zero, while the deviation from zero occurs towards 1. We could make these deviations bipolar instead, simultaneously reducing the maximum deviation. The (linear) midpoints of the oscillation ranges are $(\tilde{k}/2)/(1+\tilde{k})$ for the stopband and $(1+\tilde{k}/2)/(1+\tilde{k})$ for the passband. We can take the range $[(\tilde{k}/2)/(1+\tilde{k}), (1+\tilde{k}/2)/(1+\tilde{k})]$ between these middles and stretch it to $[0, 1]$ This can be achieved by the transformation

$$\bar{H}' = (1+\tilde{k})\bar{H} - \tilde{k}/2 = (1+\tilde{k})\bar{H} - \frac{\tilde{k}}{2} \cdot \bar{H}_{\mathrm{AP}}$$

which should be applied to both lowpass and highpass:

$$\bar{H}_1' = (1+\tilde{k})\bar{H}_1 - \frac{\tilde{k}}{2} \cdot \bar{H}_{\mathrm{AP}}$$
$$\bar{H}_2' = (1+\tilde{k})\bar{H}_2 - \frac{\tilde{k}}{2} \cdot \bar{H}_{\mathrm{AP}}$$

Thereby the deviation amplitude is multiplied by $1+\tilde{k}$, but simultaneously the deviations become centered around the ideal values 0 and 1, which effectively halves the deviations. Thus the deviation amplitude is effectively multiplied by $(1+\tilde{k})/2$ becoming equal simply to $\tilde{k}/2$ (Fig. 10.17).

Multiplying the above equations by $e^{j\varphi(\omega)}$ we obtain the same transformation for H_1 and H_2:

$$H_1' = (1+\tilde{k})H_1 - \frac{\tilde{k}}{2} \cdot H_{\mathrm{AP}}$$
$$H_2' = (1+\tilde{k})H_2 - \frac{\tilde{k}}{2} \cdot H_{\mathrm{AP}}$$

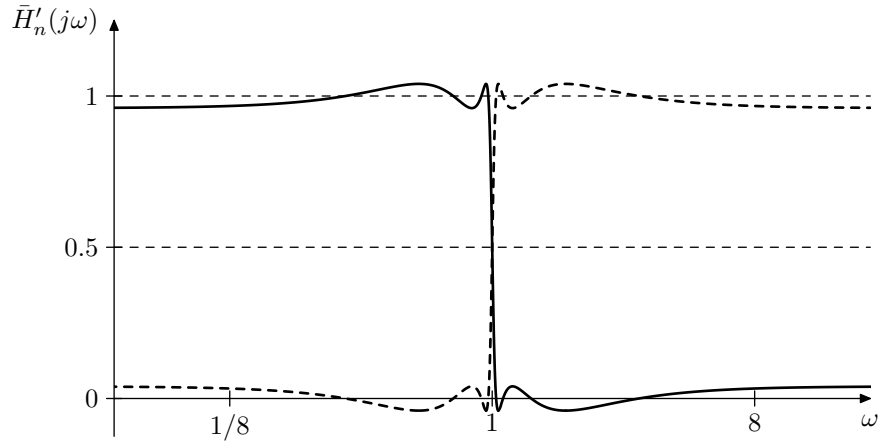


Figure 10.17: Zero-phase frequency responses of adjusted EMQF crossover low- (solid) and high-pass (dashed) outputs.

Note that thereby we still have

$$H'_1 + H'_2 = (1+\tilde{k})H_1 + (1+\tilde{k})H_2 - \tilde{k}H_{\mathrm{AP}} = H_{\mathrm{AP}} + \tilde{k}H_{\mathrm{AP}} - \tilde{k}H_{\mathrm{AP}} = H_{\mathrm{AP}}$$

Phase correction

If we need to do some processing in parallel to the crossover, then we should keep in mind that the crossover signals are phase shifted, therefore it could be a good idea to introduce the same phase shift into the signal which bypasses the crossover.

At this point we will assume that the crossover is (generalized) Linkwitz–Riley, therefore all phase shifts are identical. In this case the simplest way to construct a phase-shifted bypass signal is by adding the LP and HP outputs of the crossover together, which by the previous discussion should be an allpass signal with exactly the same phase shift as in LP and HP signals (Fig. 10.18). Notice that the LP and HP outputs of the crossover in Fig. 10.18 correspond to the $H_1(s)$ and $H_2(s)$ transfer functions in (10.31). Particularly, if we're using a Butterworth or an EMQF crossover, the squared HP signal needs to be inverted for odd N.

The approach of Fig. 10.18 doesn't work if the bypass signal processing path is not starting from the same point where the crossover is connected. In this case we might need an explicit phase-correction allpass. Fig. 10.19 shows the option of doing the phase correction prior to the processing of the bypass signal.

Rather than constructing the correction allpass following the idea of Fig. 10.18 (that is building such an allpass as another crossover with LP and HP outputs added), it is more efficient to construct this allpass directly. Indeed, by (10.30), given a crossover whose order is $2N$, the order of the allpass $H_{\mathrm{AP}}(s) = P(-s)/P(s)$ is only N. Therefore it is more efficient to implement the correction allpass simply as an N-th order filter:

In Fig. 10.19 we could swap the order of the phase correction and the processing of the bypass signal as shown in Fig. 10.20. If the processing is nonlinear, this may result in an audible change in the sound. One could argue that the

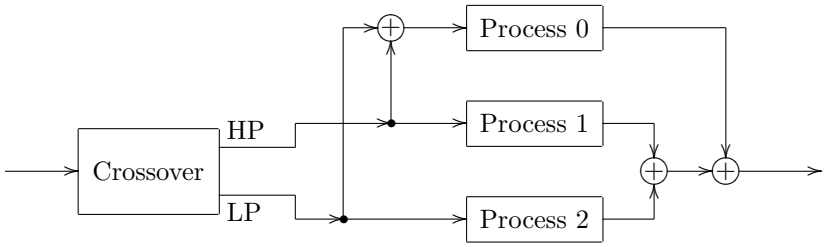


Figure 10.18: Adjusting the phase of the bypass signal.

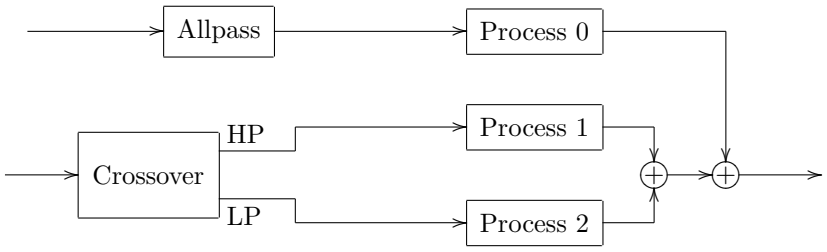


Figure 10.19: Adjusting the phase of the signal from a different source.

option shown in Fig. 10.20 is better, since the nonlinear processing is done on the original signal, while the allpass correction of the processing results would be usually inaudible (unless another nonlinear processor is following), and thus the bypass processing would sound pretty much identical to the one in the absence of the phase shifts. However, there is a counterargument that all other processing is done on phase-shifted signals, and it would be more consistent to do the same for the bypass signal.

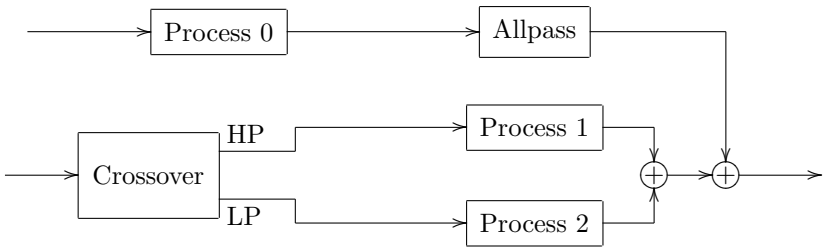


Figure 10.20: Correction allpass at the end of processing.

A more complicated situation arises if we want to stack the crossovers to make a multiband crossover because in this case the phase correction is needed even if there is no bypass signal. Consider Fig. 10.21, where A_2 denotes an allpass introducing the phase shift corresponding to the crossover C_2. The LP and HP outputs of the crossover C_1 are completely in-phase, therefore the signal going through the processor P_1 is, from the phase shift perspective, essentially the same as bypass signal of the crossover C_2 and thus needs phase correction equivalent to the phase contribution of C_2. Or, looking from a slightly different angle, the input signals of processors P_2 and P_3 contain phase shifts from both crossovers, while the input signal of processor P_1 contains the phase shift only from the first crossover and thus needs an additional phase shift by A_2.

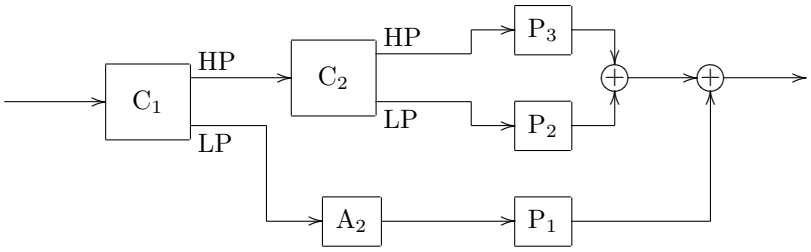


Figure 10.21: Phase correction in 3-way crossover mixing.

If the bypass signal processing is present, we could modify the structure of Fig. 10.21 as shown in Fig. 10.22. An alternative option is presented in Fig. 10.23 and yet another option (requiring one more corection allpass) in Fig. 10.24. Notice that Fig. 10.22 does all phase shifting at the beginning and Fig. 10.24 does all phase shifting at the end, while the structure in Fig. 10.23 is a kind of in-between mixture of Fig. 10.22 and Fig. 10.24. These ideas generalize by induction to higher numbers of bands, where in Fig. 10.22 we'll be adding new crossover-allpass pairs on the left, whereas in Fig. 10.24 we would be adding crossovers on the left and allpasses on the right.

In four-way crossover mixing there are new options, e.g. there is a symmetric band splitting option shown in Fig. 10.25. However practically it is not much different from the approach of Fig. 10.22 generalized to 4 bands, since the total phase shifts in the input signals of all processing units contain the total sums of the phase shifts associated with all crossovers in either case.

Note that in Fig. 10.25 one could also wish to replace the allpasses A_2 and A_3 with a single allpass A_{23} in one of the paths, which just corrects the difference between the phase responses of C_2 and C_3. This is however not possible. Indeed, assuming identical orders of C_2 and C_3, their phase responses are identical at each of the points $\omega = 0$ and $\omega = \infty$. Therefore the phase response of A_{23} must be equal to zero at both $\omega = 0$ and $\omega = \infty$. But this is not possible for a differential allpass.[18] The argument becomes somewhat more complicated if the crossovers are allowed to have different orders, where one would need to consider

[18]This would have been formally possible if A_{23} is allowed to be unstable, however the order of A_{23} would have been equal to the sum of the orders of A_2 and A_3. We mention this because this has a clear analogy to the phase splitter discussed later in the text.

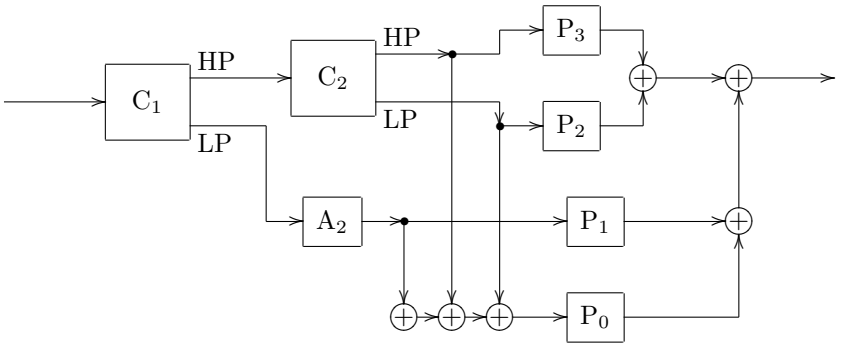


Figure 10.22: Phase correction of bypass signal in 3-way crossover mixing.

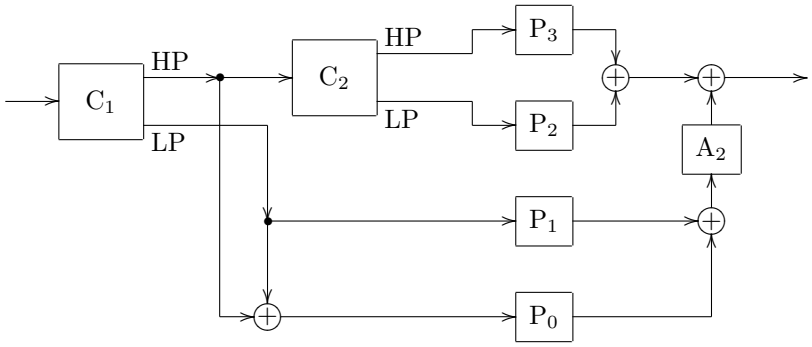


Figure 10.23: Another way of phase correction of bypass signal in 3-way crossover mixing.

the factored forms of A_2 and A_3, essentially reaching the same conclusion.

10.9 Even/odd allpass decomposition

Suppose we are given a filter $H(s)$ defined by (9.18). In this section we are going to show that $H(s)$ is expressible as a linear combination of the "even" and "odd" allpasses, that is allpasses based on the even and odd poles of $H(s)$.

Recall that we have defined even poles as solutions of $f = j$ (or equivalently $1 + jf = 0$) and odd poles as solutions of $f = -j$ (or equivalently $1 - jf = 0$). Let's introduce the following notation:

$$(1+jf)_- = \prod_{\substack{1+jf(-jp_n)=0 \\ \operatorname{Re} p_n<0}} (s-p_n)$$

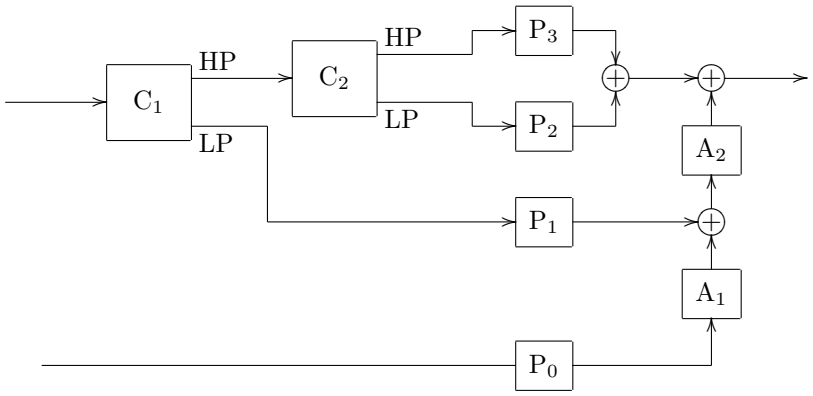


Figure 10.24: 3-way crossover mixing with all phase correction done at the end.

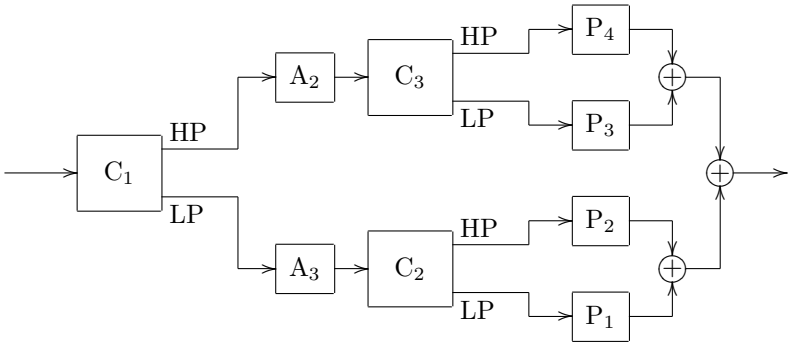


Figure 10.25: Symmetric 4-way crossover.

$$(1+jf)_+ = \prod_{\substack{1+jf(-jp_n)=0 \\ \mathrm{Re}\, p_n>0}} (s-p_n)$$

$$(1+jf)_\pm = \prod_{1+jf(-jp_n)=0} (s-p_n) = (1+jf)_+(1+jf)_-$$

that is the product is being taken over all left- or respectively right-semiplane *even* poles p_n of $H(s)H(-s)$ in the first two lines, and over all even poles of $H(s)H(-s)$ in the third line. We will also use $(1-jf)_-$, $(1-jf)_+$ and $(1-jf)_\pm$ with similar meanings for the respective products based on the odd poles of $H(s)H(-s)$. We also introduce

$$(f)_\infty = \prod_{f(-jz_n)=\infty} (s-z_n)$$

where z_n goes over all poles of $f(-js)$, or, equivalently, over all zeros of $H(s)$.

In this notation we could express the construction of $H(s)$ from its poles and zeros as

$$H(s) = g_? \cdot \frac{(f)_\infty}{(1-jf)_-(1+jf)_-} = \\ = H(j\omega_0)\frac{[(1-jf)_-](j\omega_0)\cdot[(1+jf)_-](j\omega_0)}{[(f)_\infty](j\omega_0)} \cdot \frac{(f)_\infty}{(1-jf)_-(1+jf)_-} \tag{10.35}$$

where $g_?$ denotes a placeholder for the yet unknown gain coefficient, which we then find from the requirement of $H(s)$ to have a specific value $H(j\omega_0)$ at some point $s = j\omega_0$ on the imaginary axis, and where $[(1-jf)_-](\omega_0)$ denotes the value of $(1-jf)_-$ at $s = j\omega_0$ and so on. In the simplest case we will let $\omega_0 = 0$, which gives $H(j\omega_0) = H(0) = 1/\sqrt{1+f^2(0)}$, however we will also need to be able to take other choices of ω_0.

Now let's introduce the "even" and "odd" allpasses:

$$H_e(s) = g_e \cdot \frac{(1-jf)_+}{(1+jf)_-} \tag{10.36a}$$

$$H_o(s) = g_o \cdot \frac{(1+jf)_+}{(1-jf)_-} \tag{10.36b}$$

where the gains g_e and g_o are defined by the conditions $H_e(j\omega_0) = 1$ and $H_o(j\omega_0) = 1$:

$$g_e = \frac{[(1+jf)_-](j\omega_0)}{[(1-jf)_+](j\omega_0)}$$

$$g_o = \frac{[(1-jf)_-](j\omega_0)}{[(1+jf)_+](j\omega_0)}$$

(note that allpasses defined in this manner can be trivially built as cascades of 2nd- and 1st-order sections). The allpass property of H_e and H_o follows from the fact that $f(\omega)$ is a real function of ω, therefore the even an odd poles of $H(s)H(-s)$ (which are respectively the solutions of $1+jf = 0$ and $1-jf = 0$) are mutually conjugate in terms of ω, that is they are symmetric with respect to the imaginary axis in terms of s. Figs. 8.11, 8.12 and other similar figures illustrate.

Apparently both H_e and H_o are stable filters. If additionally $f(\omega)$ is an odd function and $\omega_0 = 0$, then H_e and H_o are real. Indeed, suppose $1+jf(-js) = 0$, that is s is an even pole. Then s^* is also an even pole since

$$1+jf(-js^*) = 1-jf(js^*) = 1-jf((-js)^*) = 1-j\cdot(f(-js))^* = \\ = 1+(jf(-js))^* = (1+jf(-js))^* = 0^* = 0$$

The same can be shown for odd poles. Therefore the poles of each of the H_e and H_o are mutually conjugate and, since $H_e(0) = H_o(0) = 1$, both fiilters are real. If $f(\omega)$ is not an odd function, particularly if $f(\omega)$ is even, then the poles of H_e and H_o do not have the conjugate symmetry, therefore H_e and H_o are essentially complex filters. However this shouldn't be a problem, since we will use H_e and H_o only as intermediate transformation helpers.

Now we attempt express $H(s)$ as a linear combination of H_e and H_o. Consider the obvious algebraic relationship:

$$1 + \frac{1-jf}{1+jf} = 2\frac{1}{1+jf} \tag{10.37}$$

where $f = f(\omega)$. Equation (10.37), if interpreted in terms of $s = j\omega$, can be understood as a relationship between three transfer functions, the two transfer functions 1 and $(1-jf)/(1+jf)$ in the left-hand side adding up to the doubled transfer function $1/(1+jf)$ in the right-hand side. The poles of these transfer functions are identical[19] and consist of the full set of the even poles of $H(s)H(-s)$.

By analysing the behavior of these transfer functions for $\omega \in \mathbb{R}$ we also notice that the two functions in the left-hand side of (10.37) are allpasses, while the transfer function $1/(1+jf)$ in the right-hand side has an amplitude response identical to $|H(s)|$. So, amplitude response-wise (10.37) is already what we are looking for and we just need to correct it so that it also becomes what we want transfer function-wise.

Let's multiply (10.37) by H_o:

$$H_o(s) + \frac{1-jf}{1+jf}H_o(s) = 2\frac{1}{1+jf}H_o(s) \tag{10.38}$$

Considering the product in the right-hand side of (10.38) we have

$$\frac{1}{1+jf}H_o(s) = g_? \cdot \frac{(f)_\infty}{(1+jf)_\pm} \cdot \frac{(1+jf)_+}{(1-jf)_-} = g_? \cdot \frac{(f)_\infty}{(1+jf)_-(1-jf)_-}$$

Comparing to (10.35) we notice that we essentially have obtained $H(s)$. Matching the values at $s = j\omega_0$ to find $g_?$ (and remembering that $H_o(j\omega_0) = 1$) we obtain

$$\frac{1}{1+jf}H_o(s) = \frac{H(s)}{(1+jf(\omega_0))H(j\omega_0)}$$

Considering the second term in the left-hand side of (10.38) we obtain

$$\frac{1-jf}{1+jf}H_o(s) = g_? \cdot \frac{(1-jf)_\pm}{(1+jf)_\pm} \cdot \frac{(1+jf)_+}{(1-jf)_-} = g_? \cdot \frac{(1-jf)_+}{(1+jf)_-}$$

Comparing to (10.36a) we notice that we essentially have obtained $H_e(s)$. Matching the values at $s = j\omega_0$ we obtain

$$\frac{1-jf}{1+jf}H_o(s) = \frac{1-jf(\omega_0)}{1+jf(\omega_0)}H_e(s)$$

Thus (10.38) turns into

$$H_o(s) + \frac{1-jf(\omega_0)}{1+jf(\omega_0)}H_e(s) = 2\frac{H(s)}{(1+jf(\omega_0))H(j\omega_0)}$$

[19] The identity transfer function 1 in the left-hand side obviously has no poles, but we could also write it as $(1+jf)/(1+jf)$ in which case it formally has the same poles (which are then cancelled by the zeros).

or

$$(1 + jf(\omega_0))H_o(s) + (1 - jf(\omega_0))H_e(s) = 2\frac{H(s)}{H(j\omega_0)} \tag{10.39}$$

Thus we have represented $H(s)$ as a linear combination of $H_o(s)$ and $H_e(s)$.

If $\omega_0 = 0$ and f is an odd function, then $f(j\omega_0) = f(0) = 0$. Thus we obtain $1 \pm jf(\omega_0) = 1$ and $H(j\omega_0) = 1/\sqrt{1 + f^2(0)} = 1$ and therefore (10.39) turns into

$$H_o(s) + H_e(s) = 2H(s)$$

If the order of f is even, then generally $f(0) \neq 0$ and the coefficients of the linear combination (10.39) are complex. Note that forcing $f(0) = 0$ or choosing another ω_0 such that $f(\omega_0) = 0$ in this case doesn't help, since the allpasses themselves are still complex. However, as we already mentioned, this won't be a problem for our puproses.

10.10 Analytic filter

Sometimes in signal processing we want to deal with the so called *analytic signals*, which are defined as signals whose Fourier spectrum doesn't contain any negative frequencies (that is the amplitudes of the negative frequency partials are all zero). Since spectra of real signals must be Hermitian, apparently analytic signals can't be real, thus they are essentially complex.

Occasionally there is a need to convert a real signal into an analytic signal by dropping all of its negative frequency partials. This is very similar to the lowpass filtering, except that this time we want to dampen not the frequencies $|\omega| > 1$ but the frequencies $\omega < 0$. Such filter can be referred to as *analytic filter*. The process of removing the negative frequencies from a real signal is also known as the *Hilbert transform*, for that reason the analytic filter is probably more commonly known under the name *Hilbert transformer*.

The opposite conversion is simple: we just take the doubled real part of the analytic signal. That is, given an analytic signal $x_{>0}(t)$ we can restore the original signal $x(t)$ by

$$x(t) = 2\,\mathrm{Re}\, x_{>0}(t) \tag{10.40}$$

This effectively turns each complex partial of the form $X(\omega)e^{j\omega t}$ to a real partial $2{\cdot}|X(\omega)|{\cdot}\cos(\omega t{+}\arg X(\omega))$, which can be equivalently seen as adding a negative frequency partial $X^*(\omega)e^{-j\omega t}$.

Construction from a lowpass

The basic idea of constructing an analytic filter is simple, we take a unit-cutoff lowpass filter (so that the passband is $|\omega| < 1$ and the stopband is $|\omega| > 1$) and rotate its transfer functions along the imaginary Riemann circle:

$$H_{>0}(s) = H_{\mathrm{LP}}(\rho_{-j}(s)) \tag{10.41}$$

This effectively rotates the frequency response along the real Riemann circle:

$$H_{>0}(j\omega) = H_{\mathrm{LP}}(j\rho_{-1}(\omega))$$

thereby transforming the passband $(-1, 1)$ to $(0, +\infty)$ and the stopband $|\omega| > 1$ to $(-\infty, 0)$ (Fig. 10.26).

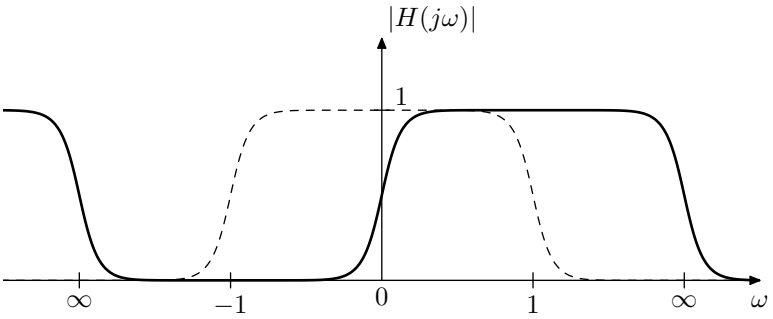


Figure 10.26: Conversion of a unit-cutoff lowpass filter into an analytic filter by a rotation of the real Riemann circle.

The poles and zeros of H_{LP} are respectively transformed by the inverse of ρ_{-j}, which is ρ_{+j}. Therefore the poles $\tilde{p}_n$ and zeros $\tilde{z}_n$ of $H_{>0}$ can be explicitly obtained from the poles p_n and zeros z_n of H_{LP} by $\tilde{p}_n = \rho_{+j}(p_n)$, $\tilde{z}_n = \rho_{+j}(z_n)$. The gain coefficient of $H_{>0}$ can be found by equating the frequency responses of $H_{>0}$ and H_{LP} at corresponding frequencies, e.g. $H_{>0}(j) = H_{\text{LP}}(0)$. Note that in principle, we can multiply $H_{>0}(s)$ by an arbitrary complex number of unit magnitude, as this wouldn't change the ampitude response of $H_{>0}$. Particularly, we could let $H_{>0}(0) = |H_{\text{LP}}(-j)| = 1/\sqrt{1 + f^2(-1)}$.

Parallel allpass implementation

In practical implementation we usually don't want to deal with complex signals. In this case the output of the filter is a fundamentally complex signal, so we can't avoid that. However we could try to construct as much as possible of $H_{>0}$ staying in real signal domain. Particularly we could attempt to express $H_{>0}$ using real filters, where we then do some postprocessing by mixing the real outputs of those filters with possibly complex coefficients. This is indeed possible.

Suppose H_{LP} is implemented using (9.18). Recall that by (10.39) the lowpass filter H_{LP} can be represented as a linear combination of allpasses. This linear combination must be preserved by the rotation ρ_{-j} in (10.41) giving

$$\frac{2H_{>0}(s)}{H_{\text{LP}}(\omega_0)} = (1 + jf(\omega_0))H_o(\rho_{-j}(s)) + (1 - jf(\omega_0))H_e(\rho_{-j}(s))$$

Since ρ_{-j} rotates along the imaginary axis (in terms of s plane and its respective Riemann sphere), the allpass property should be preserved by this transformation and $H_o(\rho_{-j}(s))$ and $H_o(\rho_{-j}(s))$ must still be allpasses. It would be convenient to reexpress $H_o(\rho_{-j}(s))$ and $H_o(\rho_{-j}(s))$ in terms of their new poles after the transformation by ρ_{-j}.

As mentioned, the poles are transformed by $\tilde{p}_n = \rho_{+j}(p_n)$. Therefore let's introduce the new allpasses

$$\tilde{H}_o(s) = \tilde{g}_o \cdot \prod_{p_n \text{ odd}} \frac{s + \tilde{p}_n^*}{s - \tilde{p}_n}$$

$$\tilde{H}_e(s) = \tilde{g}_e \cdot \prod_{p_n \text{ even}} \frac{s + \tilde{p}_n^*}{s - \tilde{p}_n}$$

where $\tilde{g}_o$ and $\tilde{g}_e$ are defined from the conditions $\tilde{H}_o(0) = 1$, $\tilde{H}_e(0) = 1$. Apparently,

$$H_o(\rho_{-j}(s)) = g_? \cdot \tilde{H}_o(s)$$
$$H_e(\rho_{-j}(s)) = g_? \cdot \tilde{H}_e(s)$$

where $g_?$ denote two different yet uknown coefficients. Substituting $s = 0$ into the above we find that these coefficients must be simply equal to $H_o(-j)$ and $H_e(-j)$ respectively and therefore

$$\frac{2H_{>0}(s)}{H_{\text{LP}}(j\omega_0)} = (1 + jf(\omega_0))H_o(-j)\tilde{H}_o(s) + (1 - jf(\omega_0))H_e(-j)\tilde{H}_e(s)$$

Up to this point we have been explicitly keeping the freedom of choice of ω_0. This has been done on purpose, as now we can see a good choice for ω_0. By letting $\omega_0 = -1$ we have $H_o(-j) = 1$ and $H_e(-j) = 1$ and thereby

$$\frac{2H_{>0}(s)}{H_{\text{LP}}(-j)} = (1 + jf(-1))\tilde{H}_o(s) + (1 - jf(-1))\tilde{H}_e(s)$$

or

$$H_{>0}(s) = H_{\text{LP}}(-j) \cdot \left(\frac{1 + jf(-1)}{2} \tilde{H}_o(s) + \frac{1 - jf(-1)}{2} \tilde{H}_e(s) \right) \tag{10.42}$$

Equation (10.42) would be an acceptable answer, provided $\tilde{H}_o(s)$ and $\tilde{H}_e(s)$ are real filters. Since $\tilde{H}_o(0) = 1$ and $\tilde{H}_e(0) = 1$ by construction, we only need to make sure that the poles of each of the $\tilde{H}_o(s)$ and $\tilde{H}_e(s)$ are conjugate symmetric.

Recall that the poles of $\tilde{H}_o(s)$ and $\tilde{H}_e(s)$ are obtained by $\tilde{p}_n = \rho_{+j}(p_n)$. We therefore wonder, what would be the relationship between the two preimages p_1, p_2 of a conjugate pair of poles $\tilde{p}_2 = \tilde{p}_1^*$. Considering visually the effect of ρ_{+j} on the Riemann sphere, we could guess that $p_2 = 1/p_1^*$. Verifying algebraically:

$$\begin{aligned}\tilde{p}_2 = \rho_{+j}(p_2) = \rho_{+j}(1/p_1^*) = j\rho_{+1}(-j/p_1^*) = j\rho_{+1}\left((j/p_1)^*\right) = j\left(\rho_{+1}(j/p_1)\right)^* = \\ = \left(-j\rho_{+1}(j/p_1)\right)^* = \left(j\rho_{+1}(p_1/j)\right)^* = \left(j\rho_{+1}(-jp_1)\right)^* = \left(\rho_{+j}(p_1)\right)^* = \tilde{p}_1^*\end{aligned}$$

Therefore, given that poles of H_{LP} have the conjugate reciprocal symmetry $p_2 = 1/p_1^*$ (that is, if s is a pole of H_{LP} then so is $1/s^*$), the poles of $\tilde{H}_o$ and $\tilde{H}_e$ will have the conjugate symmetry.

The poles of H_{LP} will have the conjugate reciprocal symmetry, given that $f(\omega)$ is a real function such that $f(1/x) = 1/f(x)$. Indeed, suppose $f(1/x) = 1/f(x)$ and $1 + f^2(-js) = 0$. Then

$$\begin{aligned}1 + f^2(-j/s^*) = 1 + f^2(1/js^*) = 1 + f^2(1/(-js)^*) = 1 + \left(f^2(1/(-js))\right)^* = \\ = \left(1 + f^2(1/(-js))\right)^* = \left(1 + \frac{1}{f^2(-js)}\right)^* = \left(\frac{1 + f^2(-js)}{f^2(1/(-js))}\right)^* = 0^* = 0\end{aligned}$$

We already know one specific kind of lowpass filter where f has such reciprocal symmetry: the EMQF filter (with the Butterworth filter as its limiting case).

Note that EMQF poles not only have conjugate reciprocal symmetry, but are simply lying on the unit circle, in which case conjugate reciprocation simply maps the poles to themselves: $1/p^* = p$. Since ρ_{+j} maps the unit circle to the real axis, the poles $\tilde{p}_n$ are real. Also, since ρ_{+j} is a rotation in the direction of the imaginary axis, it maps left semiplane poles to the left semiplane poles and thus $\tilde{p}_n < 0\ \forall n$. Thus, we are having stable real H_o and H_e whose poles are also all real.

Real and imaginary allpasses

The expression (10.42) can be simplified a bit further. Notice that for an EMQF filter we are having $f(-1) = (-1)^N$ where N is the filter's order. Respectively $H_{\mathrm{LP}}(-j) = 1/\sqrt{2}$. Therefore (10.42) turns into

$$H_{>0}(s) = \begin{cases} \frac{1}{\sqrt{2}} \cdot \left(\frac{1+j}{2} \tilde{H}_o(s) + \frac{1-j}{2} \tilde{H}_e(s) \right) & N \text{ even} \\ \frac{1}{\sqrt{2}} \cdot \left(\frac{1-j}{2} \tilde{H}_o(s) + \frac{1+j}{2} \tilde{H}_e(s) \right) & N \text{ odd} \end{cases}$$

Recall that we can multiply $H_{>0}(s)$ by any unit-magnitude complex number without changing the amplitude response. Particulary, we could multiply it by $j^{1/2} = (1+j)/\sqrt{2}$ obtaining

$$H_{>0}(s) = \begin{cases} \frac{\tilde{H}_e(s) + j\tilde{H}_o(s)}{2} & N \text{ even} \\ \frac{\tilde{H}_o(s) + j\tilde{H}_e(s)}{2} & N \text{ odd} \end{cases} \tag{10.43}$$

Thus the real and imaginary parts of the output signal of $H_{>0}$ are obtained completely separately from two parallel allpasses $\tilde{H}_o$ and $\tilde{H}_e$.

10.11 Phase splitter

There is another, conceptually completely different, but closely mathematically related approach to constructing the Hilbert transformer. Considering a single positive-frequency complex sinusoidal partial

$$e^{j\omega t} = \cos\omega t + j \sin\omega t = \cos\omega t + j \cos(\omega t - \pi/2) \qquad (\omega > 0)$$

we notice that the imaginary part of the signal is phase-delayed by 90° relatively to the real part. Now let $\hat{H}_{>0}$ denote the analytic filter operator. That is, applying $\hat{H}_{>0}$ discards the negative frequency partials from the signal. Then, applying analytic filtering to $\cos\omega t$ we have

$$\hat{H}_{>0} \cos\omega t = \hat{H}_{>0} \frac{e^{j\omega t} + e^{-j\omega t}}{2} = \frac{e^{j\omega t}}{2} = \frac{\cos\omega t + j\cos(\omega t - \pi/2)}{2} \qquad (\omega > 0)$$

Respectively, for a general real signal we have

$$\hat{H}_{>0} \int_0^\infty a(\omega) \cos(\omega t + \varphi(\omega)) \frac{\mathrm{d}\omega}{2\pi} =$$

$$= \frac{1}{2}\int_0^\infty a(\omega)\cos(\omega t + \varphi(\omega))\,\frac{\mathrm{d}\omega}{2\pi} + \frac{j}{2}\int_0^\infty a(\omega)\cos(\omega t + \varphi(\omega) - \pi/2)\,\frac{\mathrm{d}\omega}{2\pi}$$

Introducing notations:

$$x(t) = \int_0^\infty a(\omega)\cos(\omega t + \varphi(\omega))\,\frac{\mathrm{d}\omega}{2\pi}$$
$$x_{-90}(t) = \int_0^\infty a(\omega)\cos(\omega t + \varphi(\omega) - \pi/2)\,\frac{\mathrm{d}\omega}{2\pi}$$

(where $x_{-90}(t)$ is the signal $x(t)$ with all real sinusoidal partials phase-shifted by $-90°$) we have

$$\hat{H}_{>0}x(t) = \frac{x(t) + jx_{-90}(t)}{2} \tag{10.44}$$

Equation (10.44) gives us another approach to the implementation of the analytic filter: we take the halved original signal as its own real part, and phase shift the partials of its real spectrum by $-90°$ to obtain the imaginary part. Also notice that (10.44) is exactly the opposite of (10.40).

Differently from the approach in Section 10.10 where we didn't care about the phase, the approach of (10.44) explicitly preserves the phase of the partials, thus (10.44) defines a zero-phase analytic filter. Unfortunately, as we shall see later, such filter cannot be implemented by a stable differential system. Still, the whole approach is somewhat more straightforward than the one of Section 10.10.

We will also develop a number of useful explicit expressions, which will be helpful in the construction of $H_{>0}$. In principle, the same expressions could have been derived in Section 10.10 (as the answers are essentially the same), however the derivations will be somewhat more direct in the context of the new approach.

Complex spectral form

Before we get to the construction of the $-90°$ phase shifter, we need to reexpress this phase shifting in terms of complex spectral partials:

$$x(t) = \int_0^\infty \frac{a(|\omega|)}{2}\left(e^{j(\omega t+\varphi(\omega))} + e^{-j(\omega t+\varphi(\omega))}\right)\frac{\mathrm{d}\omega}{2\pi}$$
$$x_{-90}(t) = \int_0^\infty \frac{a(|\omega|)}{2}\left(e^{j(\omega t+\varphi(\omega)-\pi/2)} + e^{-j(\omega t+\varphi(\omega)-\pi/2)}\right)\frac{\mathrm{d}\omega}{2\pi} =$$
$$= \int_0^\infty \frac{a(|\omega|)}{2}\left(e^{-j\pi/2}e^{j(\omega t+\varphi(\omega))} + e^{j\pi/2}e^{-j(\omega t+\varphi(\omega))}\right)\frac{\mathrm{d}\omega}{2\pi} \tag{10.45}$$

That is we need to phase shift the positive frequency partials by $-90°$ and phase shift the negative frequency partials by $+90°$. Since the amplitudes are unchanged by the phase-shifting, this is an allpass transformation, which we can denote by the $\hat{H}_{-90}$ operator:

$$x_{-90}(t) = \hat{H}_{-90}x(t)$$

Note that the fact that the negative frequencies need to be phase shifted by the opposite amount is in agreement with the fact that $x_{-90}(t)$, being the imaginary part of $\hat{H}_{>0}x(t)$, needs to be a real signal. Therefore $\hat{H}_{-90}$ needs to preserve

the hermiticity of the spectrum of $x(t)$. This means that the frequency response of $\hat{H}_{-90}$ must be a Hermitian function, which implies the phase response being an odd function.

The frequency response of the $\hat{H}_{-90}$ allpass is obviously

$$H_{-90}(j\omega) = \begin{cases} -j & \text{if } \omega > 0 \\ j & \text{if } \omega < 0 \end{cases}$$

We are still uncertain as to which value to assign to $H_{-90}(0)$. In principle, according to (10.45) the zero-frequency partial should be completely killed in x_{-90}, thus

$$H_{-90}(j\omega) = -j \operatorname{sgn} \omega = \begin{cases} -j & \text{if } \omega > 0 \\ 0 & \text{if } \omega = 0 \\ j & \text{if } \omega < 0 \end{cases}$$

Strictly speaking, this kills the allpass property of $\hat{H}_{-90}$ at $\omega = 0$, but this is actually the only way to keep $H_{-90}(j\omega)$ Hermitian.

Now we can rewrite (10.44) in the pure operator form

$$\hat{H}_{>0} = \frac{1 + j\hat{H}_{-90}}{2} \tag{10.46}$$

or in the frequency response form

$$H_{>0}(j\omega) = \frac{1 + jH_{-90}(j\omega)}{2} = \frac{1 + \operatorname{sgn} \omega}{2} \tag{10.47}$$

The complex spectrum interpretation of (10.44) gives another insight into why does it describe an analytic filter. Given a positive-frequency complex sinusoidal signal $x(t) = e^{j\omega t}$ we have $x_{-90}(t) = -je^{j\omega t}$ and respectively

$$\hat{H}_{>0}x(t) = \frac{e^{j\omega t} + j \cdot (-j)e^{j\omega t}}{2} = \frac{e^{j\omega t} + e^{j\omega t}}{2} = x(t) \qquad (\omega > 0)$$

that is $x(t)$ is unchanged by $\hat{H}_{>0}$. On the other hand, if $\omega < 0$, then $x_{-90}(t) = je^{j\omega t}$ and respectively

$$\hat{H}_{>0}x(t) = \frac{e^{j\omega t} + j \cdot je^{j\omega t}}{2} = \frac{e^{j\omega t} - e^{j\omega t}}{2} = 0 \qquad (\omega < 0)$$

The DC at $\omega = 0$ is neither a positive- nor a negative-frequency partial. According to what we discussed above, it is killed by $\hat{H}_{-90}$ and thus

$$\hat{H}_{>0}1(t) = \frac{1(t) + 0j}{2} = \frac{1}{2} \qquad (\omega = 0)$$

(where $1(t)$ denotes a signal equal to 1 everywhere).

Rational 90° phase shifting allpass

We are looking for an allpass filter H_{-90} whose frequency response is

$$H_{-90}(j\omega) = -j \operatorname{sgn} \omega \tag{10.48}$$

Apparently $H_{-90}(s)$ can't be a rational function, since rational functions are continuous everywhere except at their poles, where they gradually approach infinity, thus a rational function cannot accommodate a jump from j to $-j$ which $H_{-90}(j\omega)$ has at $\omega = 0$. But we still can build a rational $H(s)$ which approximates the ideal $H_{-90}(j\omega)$ for $s = j\omega$.

As mentioned earlier, the ideal H_{-90} is an allpass everywhere except at $\omega = 0$. Since we are building an approximation of H_{-90} anyway, we can ignore that fact and build an approximation which is a perfect allpass. This will simplify our goal, since then we can construct the allpass in terms of its phase response. Therefore let

$$\varphi_\infty(\omega) = \begin{cases} -90^\circ & \forall\omega > 0 \\ +90^\circ & \forall\omega < 0 \end{cases}$$

be the ideal phase response of our allpass.[20] So, how can we build a rational allpass transfer function approximating $\varphi_\infty(\omega)$?

Consider the fact that the frequency response of an allpass can be explicitly written in terms of its phase response:

$$H(j\omega) = e^{j\varphi(\omega)}$$

Using (9.11a) we can rewrite the same as

$$H(j\omega) = \rho_{+1}\left(j \tan\frac{\varphi(\omega)}{2}\right)$$

However ρ_{+1}, being a 1st-order rational function, maps rational functions to rational functions of the same order and back. Thus, if we have a rational function $\Phi(\omega)$ of some order N such that

$$\Phi(\omega) = \tan\frac{\varphi(\omega)}{2} \tag{10.49}$$

then $j\Phi(\omega)$ and $H(j\omega) = \rho_{+1}(j\Phi(\omega))$ will also be rational functions of the same order N and the phase response of H will be equal to $\varphi(\omega)$.

Letting $s = j\omega$ we rewrite $H(j\omega) = \rho_{+1}(j\Phi(\omega))$ as

$$H(s) = \rho_{+1}(j\Phi(-js)) \tag{10.50}$$

Will $H(s)$ be a real function of s? Since $\varphi(\omega)$ must be real odd, so must be $\Phi(\omega)$. This implies that it must be representable in the form $\Phi(\omega) = \omega\Phi_2(\omega^2)$ where Φ_2 is some other real function. Therefore

$$H(s) = \rho_{+1}(j\Phi(-js)) = \rho_{+1}\big(j\cdot(-js)\Phi_2((-js)^2)\big) = \rho_{+1}(s\Phi_2(-s^2))$$

and thus $H(s)$ is real.

Before proceeding to the construction of $\Phi(\omega)$ we would like to give one warning. The allpass transfer functions $H(s)$, which will arise from the application of (10.50) to the obtained $\Phi(\omega)$, will be unstable. This corresponds to

[20] Since we want our allpass approximation of H_{-90} to be a real filter, its phase response must be odd, which leaves only two possible values at $\omega = 0$: $\varphi(0) = 0$ or $\varphi(0) = 180^\circ$. If we formally include $\omega = \infty$ into the range of frequencies of interest, then we notice that the phase response at $\omega = \infty$ has the same two options.

the fact that phase responses of stable differential allpasses cannot stay around $\pm 90^\circ$ over a large range of ω.[21] This is a fundamental limitation, which we'll have to deal with. Later in this section we will describe a way of addressing this problem.

Construction of $\Phi(\omega)$

The ideal $\Phi(\omega)$ is apparently

$$\Phi_\infty(\omega) = \tan\frac{\varphi_\infty(\omega)}{2} = \begin{cases} -1 & \forall\omega > 0 \\ 1 & \forall\omega < 0 \end{cases}$$

We wish to find a rational $\Phi(\omega) \approx \Phi_\infty(\omega)$. This will ensure $\varphi(\omega) \approx \varphi_\infty(\omega)$.

Let $f(x)$ be a real rational function satisfying the unit-cutoff lowpass conditions (9.21). We would like to compose $f(x)$ with other functions in such a way that the result is an approximation of Φ_∞. This composition should still result in a real rational function and, ideally, also preserve the order of f. Therefore, good candidates for the elements of such composition are the rotations of real Riemann circle $\rho_{\pm 1}$.

As a first step, we map the pass- and stop-band areas of $f(x)$ (that is $f(x) \approx 0$ and $f(x) \approx \infty$) to the areas where $\Phi(x) = \pm 1$. This is achieved by $\Phi(x) = \rho_{\pm 1}(f(x))$ (where the $\pm$ signs are matched). We thereby obtain $\Phi(x)$ which has the desired values in the "pass"- and "stop"-bands, however the bands themselves are incorrectly positioned on the argument axis, still coinciding with the pass- and stop-bands of a unit-cutoff lowpass. We could fix this by a real Riemann circle rotation of the argument. Which turns our candidate compositions into

$$\Phi(x) = \rho_{\pm 1}(f(\rho_{\pm 1}(x))) \tag{10.51}$$

where we initially treat the $\pm$ signs as independent.

However actually the $\pm$ signs in (10.51) cannot be independent. E.g. if we choose the "inner rotation" (the rotation of the argument of $f(x)$) to be ρ_{+1}, this maps the original lowpass passband $|x| \ll 1$ to $-\infty \ll x \ll 0$. In this area we want $\Phi(x) = 1$, therefore we have to choose the "outer rotation" (the rotation of the value of $f(x)$) to be ρ_{+1} as well. In a similar way we could choose both rotations to be ρ_{-1}. This means that the $\pm$ signs in (10.51) must be matched.

Intuitively it is clear that any "lowpass" kind of $f(x)$ should result in (10.51) giving an approximation of Φ_∞. However we also need $\Phi(\omega)$ to be an odd function. Let's see what kind of restriction this means for $f(x)$. By (9.13) the rotations $\rho_{\pm 1}$ map the odd symmetry to the reciprocal symmetry, which means that

$$\Phi(-\omega) = -\Phi(\omega) \iff f(1/x) = 1/f(x)$$

which effectively brings us to the idea to use the EMQF function $f(x) = \bar{R}_N(x)$ (or the Butterworth filter function $f(x) = x^N$ as its limiting case).

Having chosen $f(x) = \bar{R}_N(x)$ we can refine the formula (10.51) a little. Suppose we chose the "+" signs in (10.51). Then $\Phi(0) = \rho_{+1}(f(1)) = \rho_{+1}(1) = \infty$.

[21] In order to convince oneself that this is indeed so, one could factor a generic stable allpass tranfer function into 1st- and 2nd-order sections and consider their phase responses, which are monotonically decaying.

Vice versa, if we choose the "−" signs, then $\Phi(0) = \rho_{-1}(f(-1)) = \rho_{+1}((-1)^N)$ which is 0 if N is odd and ∞ if N is even. In principle, this is not a very big problem, and both options are valid, but it would be just nice to have $\Phi(0) = 0$ and respectively $H(0) = 1$ all the time. This is achieved by changing (10.51) into

$$\Phi(\omega) = -\rho_{-1}(f(\rho_{+1}(\omega))) \tag{10.52}$$

The readers can convince themselves that (10.52) also gives an approximation of Φ_∞ and that in this case $\Phi(0) = 0$ regardless of N. Fig. 10.27 illustrates.

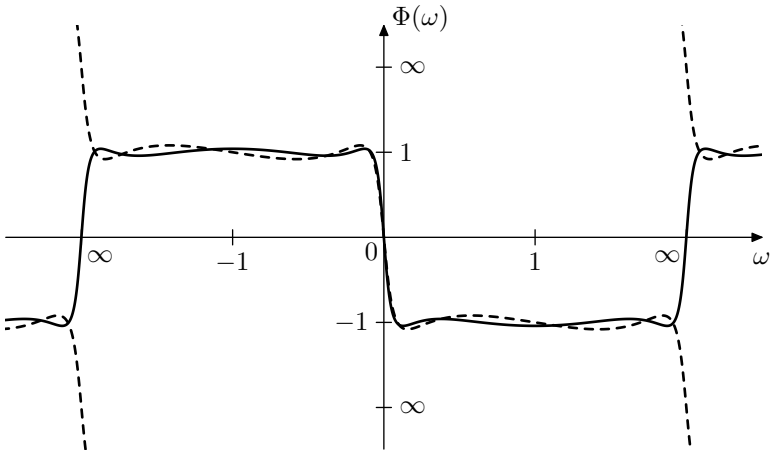


Figure 10.27: $\Phi(\omega)$ obtained from (10.52) and $f(x) = \bar{R}_N(x)$ for even (solid) and odd (dashed) N.

Explicit expression for EMQF $\Phi(\omega)$

Sticking to the idea to use the EMQF function $f(x) = \bar{R}_N(x)$ we will refer to the 90° phase shifter that we are constructing as "EMQF phase shifter". Even though this is some kind of a misnomer, this should provide a pretty clear identification of the approach we use.

Substituting $f(x) = \bar{R}_N(x)$ into (10.52) we obtain

$$\Phi(\omega) = -\rho_{-1}(\bar{R}_N(\rho_{+1}(\omega))) \tag{10.53}$$

Since $\bar{R}_N$ is a real rational function of order N, so is $\Phi(\omega)$.

In the real period-based preimage representation terms we have

$$\begin{aligned} x &= \overline{\mathrm{cd}}_K\, u \\ v &= Nu \\ \bar{R}_N(x) &= \overline{\mathrm{cd}}_{\tilde{K}}(v) \end{aligned}$$

Expressing (10.53) in the same terms we have

$$\begin{aligned} \omega &= \rho_{-1}\left(\overline{\mathrm{cd}}_K\, u\right) \\ v &= Nu \\ \Phi(\omega) &= -\rho_{-1}\left(\overline{\mathrm{cd}}_{\tilde{K}}\, v\right) \end{aligned}$$

which by (9.111) turns into

$$\begin{aligned}\omega &= -\overline{\mathrm{nd}}_{K_2}\,\frac{u}{2}\\ \frac{v}{2} &= N\frac{u}{2}\\ \Phi(\omega) &= \overline{\mathrm{nd}}_{\tilde{K}_2}\,\frac{v}{2}\end{aligned}$$

Replacing $u/2$ with u and $v/2$ with v we obtain

$$\begin{aligned}\omega &= -\overline{\mathrm{nd}}_{K_2}\,u\\ v &= Nu\\ \Phi(\omega) &= \overline{\mathrm{nd}}_{\tilde{K}_2}\,v\end{aligned}$$

and finally, switching to the explicit scaling form:

$$\omega = -\,\mathrm{nd}(u, k_2) \tag{10.54a}$$

$$v = N\frac{\tilde{K}_2}{K_2}u = \frac{\tilde{K}_2'}{K_2'}u \tag{10.54b}$$

$$\Phi(\omega) = \mathrm{nd}(v, \tilde{k}_2) \tag{10.54c}$$

where $K_2 = K(k_2)$ and $\tilde{K}_2 = K(\tilde{k}_2)$ are the quarter periods corresponding to the elliptic moduli $k_2 = \mathcal{L}^2(k)$ and $\tilde{k}_2 = \mathcal{L}^2(\tilde{k})$, that is k_2 and $\tilde{k}_2$ are obtained by the double Landen transformation from k and $\tilde{k}$. Note that, since double Landen transformation solely changes the quarter period ratios K'/K and $\tilde{K}'/\tilde{K}$ by a factor of 4, the degree equation (9.112) stays essentially the same: $\tilde{K}_2'/\tilde{K}_2 = NK_2'/K_2$. Thus the imaginary periods of the two $\overline{\mathrm{nd}}$ functions are matched, while the real period is scaled by N.

Turning the representation form into the explicit form we obtain, e.g. using real period argument normalization

$$\Phi(\omega) = \overline{\mathrm{nd}}_{\tilde{K}_2}\left(N\,\overline{\mathrm{nd}}_{K_2}^{-1}(-\omega)\right) \tag{10.55}$$

Expression (10.55) defines another (normalized) elliptic rational function. Differently from the already familiar $\bar{R}_N$, this function has equiripples around ± 1 in the bands centered around $\omega = \pm 1$. Strictly speaking the amplitudes of the upwards- and downwards-pointing ripples of $\Phi(\omega)$ (shown in Fig. 10.27) are not equal, rather, the values are mutually reciprocal at the upwards- and downward-pointing peaks. It is just that in arctangent scale the reciprocal values correspond to equal deviations from 1 or from -1. Therefore the true equiripple behavior occurs in the arctangent rather than linear scale. However according to (10.49) the function $\varphi(\omega)$ (which is our true goal) is exactly the arctangent scale representation of $\Phi(\omega)$. Therefore $\varphi(\omega)$ will have true equiripples. For the sake of clarity we provide a graph of $\varphi(\omega)$ in Fig. 10.28, however notice that the only difference between Figs. 10.28 and 10.27. is the labelling of the vertical axis.

Bands of EMQF $\Phi(\omega)$

It is instructive to analyse $\Phi(\omega)$ in terms of its bands in the preimage domain. The readers can convince themselves that the bands are:

Transition band 1: $\quad |\omega| \le \sqrt{k_2'} \qquad\qquad |\Phi(\omega)| \le \sqrt{\tilde{k}_2'}$

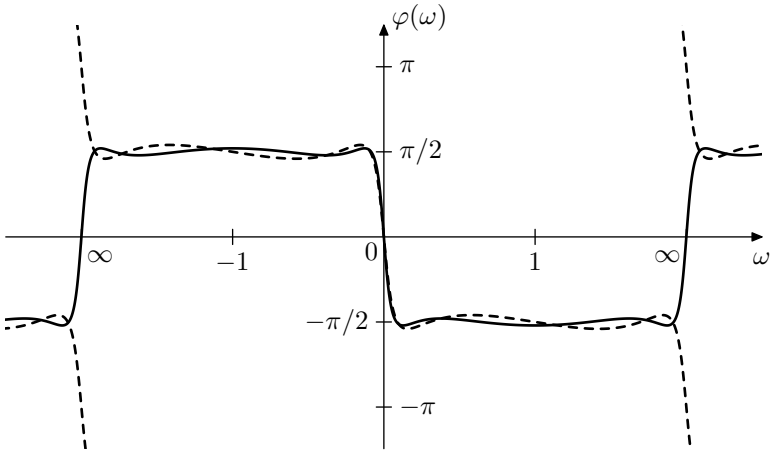


Figure 10.28: $\varphi(\omega)$ obtained from (10.55) and (10.49) for even (solid) and odd (dashed) N.

Passband 1:	$\sqrt{k_2'} \le \omega \le 1/\sqrt{k_2'}$	$-1/\sqrt{\tilde{k}_2'} \le \Phi(\omega) \le -\sqrt{\tilde{k}_2'}$
Transition band 2:	$\lvert\omega\rvert \ge 1/\sqrt{k_2'}$	depends on N
Passband 2:	$-1/\sqrt{k_2'} \le \omega \le -\sqrt{k_2'}$	$\sqrt{\tilde{k}_2'} \le \Phi(\omega) \le 1/\sqrt{\tilde{k}_2'}$

where in the transition band 2 we have $|\Phi(\omega)| \le \sqrt{\tilde{k}_2'}$ for even N and $|\Phi(\omega)| \ge 1/\sqrt{\tilde{k}_2'}$ for odd N. Fig. 9.51 can be referred to as an illustration.

It is not difficult to realize that that the "passband" ripples of $\Phi(\omega)$ are essentially obtained from the ripples that the $\overline{\text{nd}}$ function has on the real axis (which thereby results in the upwards-pointing peaks being reciprocal to the downwards-pointing peaks) and on a parallel line which is away from the real axis by one half of its imaginary period. The details are left as an exercise to the reader.

Poles and zeros of EMQF phase shifter

We would like to construct $H(s)$ from its poles and zeros. Since $H(s)$ is an allpass, it is sufficient to find the poles, while the zeros can be trivially obtained from the poles. However we could also consider obtaining the zeros explicitly.

Starting with the equations $H(s) = \infty$ and $H(s) = 0$ we apply the inverted (10.50), which is $\Phi(-js) = -j\rho_{-1}(H(s))$, yielding

$$\Phi(-js) = \mp j \tag{10.56}$$

where "$-$" should be taken for poles and "$+$" for zeros.

At this point there are different possibilities how to continue. Particularly, we could apply (10.53) which gives $\bar{R}_N(\rho_{+1}(-js)) = \pm j$ or equivalently $\bar{R}_N(-j\rho_{+j}(s)) = \pm j$. This would be pretty much the same as what we have been solving in Section 10.10.[22] It could be more interesting and practical,

[22]Except that we would obtain both stable and unstable poles this time, since there is no explicit restriction of the solutions having to be in the left semiplane.

though, to take a different path, which will allow us to obtain simple explicit expressions for the poles and zeros of $H(s)$. The obtained poles and zeros will be of course the same, since we are solving the same equations, just in a different way.

Let's use the preimage representation (10.54) to solve (10.56), in a similar way to how we were solving the pole equations for other filter types. Recall that by the imaginary argument property, $\overline{\mathrm{nd}}$ is essentially the same as $\overline{\mathrm{cd}}$, just rotated 90° in its complex argument plane. Therefore, while $\overline{\mathrm{cd}}$ was generating quasielliptic curves for its argument moving parallel to the real axis, $\overline{\mathrm{nd}}$ will generate the same curves for its argument moving parallel to the imaginary axis. In order to solve (10.56), we would like the curves to go through $\pm j$, however the movement parallel to the imaginary axis in the preimage domain is not very useful for solving (10.56), since the imaginary periods are matched for the preimages of ω and $\Phi(\omega)$, and therefore we will not obtain all possible solutions.

We should rather move parallel to the real axis. Apparently, in this case we won't generate quasielliptic curves in the representation domain, but rather the kind of lines shown in Fig. 9.54. Since $\overline{\mathrm{cd}}$ and respectively $\overline{\mathrm{nd}}$ take each value only once within a quater-period grid cell, and since the values $\pm j$ occur on horizontal lines where $\overline{\mathrm{nd}}$ turns into $j\,\overline{\mathrm{sc}}$ or $-j\,\overline{\mathrm{sc}}$ (Fig. 9.51), we need to move in one of these lines. The representation will then simply move along the imaginary axis[23] in one and the same direction, looping through the ∞ point.

Choosing the horizontal line $\operatorname{Im} v = \tilde{K}_2'$ as the principal preimage line we have $\overline{\mathrm{nd}}(v, \tilde{k}_2) = j\,\overline{\mathrm{sc}}(\operatorname{Re} v, \tilde{k}_2)$. We wish to have representation moving upwards along the imaginary axis therefore the preimages need to move towards the right (going along the line $\operatorname{Im} v = \tilde{K}_2'$). In terms of u the same movement corresponds to moving along the line

$$\operatorname{Im} u = \frac{K_2'}{\tilde{K}_2'} \operatorname{Im} v = K_2'$$

where the direction of movement of u is, obviously, also towards the right. Notice that any other possible choices of the principal preimage line of v do not generate any additional solutions of (10.56), since ω will be simply traversing along the entire imaginary axis in any case.

The value $\Phi(\omega) = \overline{\mathrm{nd}}(v, \tilde{k}_2)$ moving upwards along the imagniary axis will be traversing the points $\pm j$ at

$$v = j\tilde{K}_2' + \left(\frac{1}{2} + n\right) \tilde{K}_2 \qquad (n \in \mathbb{Z})$$

where at even n we'll have $\Phi(\omega) = j$ and at odd n we'll have $\Phi(\omega) = -j$. That is, even n correspond to zeros and odd n correspond to poles. The values of u are respectively

$$u = j\frac{K_2'}{\tilde{K}_2'}\tilde{K}_2' + \frac{K_2}{N\tilde{K}_2}\left(\frac{1}{2} + n\right)\tilde{K}_2 = jK_2' + \frac{\frac{1}{2} + n}{N} K_2$$

[23] Apparently the imaginary axis belongs to the family of lines shown in Fig. 9.54, being the boundary case between the two groups of lines on the left and on the right.

from where

$$\omega = -\overline{\mathrm{nd}}\, u = -j\,\overline{\mathrm{sc}}\left(\frac{\frac{1}{2}+n}{N}K_2,\ k_2\right)$$

from where by $s = j\omega$ we obtain

$$s = \overline{\mathrm{sc}}\left(\frac{\frac{1}{2}+n}{N}K_2,\ k_2\right) \tag{10.57}$$

where even n correspond to zeros and odd n correspond to poles. Note that in the Butterworth limit $k_2 \to 0$ the equation (10.57) turns into

$$s = \tan\left(\frac{\pi}{2}\cdot\frac{\frac{1}{2}+n}{N}\right)$$

Since all values in (10.57) are real, the solutions given by (10.57) are also real. That is the poles and zeros of $H(s)$ are real and $H(s)$ can be factored into 1st-order allpasses.

Since the period of $\overline{\mathrm{sc}}$ is $2K_2$, there are $2N$ different values of s given by (10.57). Half of them are zeros and the other half are poles, thus there are N zeros and N poles, where the poles and zeros are inverleaved (Fig. 10.29). Apparently, n can run over any range of $2N$ consecutive integers. A particularly convenient range is $n = -N \ldots (N-1)$. In this case $n = -1$ and $n = 0$ give one pole/zero pair where the pole and the zero are mutually opposite. This pole/zero pair corresponds to the lowest-cutoff 1-pole allpass factor, which is stable since the pole is obtained from $n = -1$. The values $n = 1$ and $n = -2$ give another pole/zero pair corresponding to the next 1-pole factor, which is unstable since the pole is obtained from $n = 1$. The third 1-pole factor will be stable again etc.

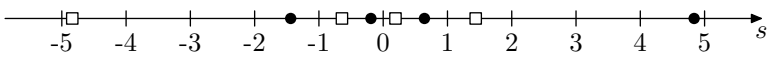


Figure 10.29: Poles (black dots) and zeros (white squares) of an EMQF phase shifter $H(s)$ for $N = 4$.

One could notice in Fig. 10.29 that there is reciprocal symmetry within the set of poles and zeros of $H(s)$. That is if s is a pole or a zero of $H(s)$, then so is $1/s$. Apparently, this is due to the property (9.74b) of the elliptic tangent function $\overline{\mathrm{sc}}$.

We could also derive a simple rule for remembering, whether for $n = -1$ one obtains a pole or a zero, that is whether the closest to zero negative value of s given by (10.57) is a stable allpass factor's pole or an unstable allpass factor's zero. First, notice that negating the allpass's cutoff is equivalent to the substitution $\omega \leftarrow -\omega$. Since the frequency response of a real filter is Hermitian, its phase response is odd, thus negating ω is equivalent to negating the phase response. Thus, since the phase responses of stable allpasses are decreasing, the phase responses of unstable (negative cutoff) allpasses are increasing. Now consider the phase shifter $H(s)$ which is a product of stable and unstable allpasses. Intuitively, as ω starts to increase from 0, the phase response first has to decrease to approximately $-90°$, therefore the allpass factor with the lowest cutoff in the chain must be stable.[24]

[24]The same reasoning can be applied to (10.43), where we want the imaginary signal to be

Bandwidth of EMQF phase shifter

In our discussion of the bands of EMQF $\Phi(\omega)$ we have established that the equiripple "passband" ranges are

$$\begin{aligned} -k_2'^{-1/2} &< \omega < -k_2'^{1/2} \\ k_2'^{1/2} &< \omega < k_2'^{-1/2} \end{aligned}$$

that is

$$k_2'^{1/2} < |\omega| < k_2'^{-1/2}$$

where we could notice that the logarithmic center of the "passband" is thereby at $\omega = 1$.

Respectively, the logarithmic bandwidth Δ expressed in octaves is a logarithm base 2 of the ratio of the passband's boundaries:

$$\Delta = \log_2 \frac{k_2'^{-1/2}}{k_2'^{1/2}} = \log_2 k_2'^{-1} = -\log_2 k_2'$$

which gives us a way to immediately find k_2' from a given bandwidth:[25]

$$k_2' = 2^{-\Delta}$$

Since the boundaries of the bands of $\varphi(\omega)$ are identical to the bands of $\Phi(\omega)$, the above formulas equally apply to the bands of $\varphi(\omega)$.

The value of $\tilde{k}_2'$, which effectively defines the amplitude of the ripples, can be computed (after having constructeed $\Phi(\omega)$) from

$$\tilde{k}_2'^{1/2} = -\Phi(k_2'^{1/2})$$

However it is more practical to directly compute the deviation of $\arg H(jk_2'^{1/2})$ from the target value $-90°$ (after having constructed $H(s)$). According to the above formula, $\omega = k_2'^{1/2}$ should be the point of maximum phase deviation (within the equiripple range) and thus the deviation of $\varphi(k_2'^{1/2}) = \arg H(jk_2'^{1/2})$ from $-90°$ should give the amplitude of the equiripples.

Since $\tilde{k}_2'$ and k_2' increase or decrease simultaneously, $H(s)$ will get larger ripple amplitudes for larger bandwidths and vice versa. Increasing the order N will result in a smaller ripple amplitude for the same bandwidth.

Apparently the "passband" doesn't need to be centered at $\omega = 1$ and can be shifted to any other center frequency by the cutoff substitution $s \leftarrow s/\omega_c$. This raises a related question of prewarping, where we could notice that the situation is pretty similar to the prewarping of a normalized 2-pole bandpass filter (discussed in connection with the LP to BP transformation in Section 4.6). Therefore the suggested way of handling the prewarping of $H(s)$ consists of the following steps:

---

phase shifted by $-90°$ compared to the real one. Therefore the lowest-cutoff allpass factor (corresponding to the pole closest to the origin) must be in the imaginary signal's allpass. Since the poles of the allpasses in (10.43) are obtained by Riemann sphere rotation ρ_{+j}, the pole closest to the origin will be obtained from the pole closest to $-j$, which is an even pole for N odd and an odd pole for N even.

[25] Note that if desired, we can also find k from k_2' by (9.110) (where we let $k_0 = k$).

1. Given the desired "passband" $[\omega_1, \omega_2]$:

$$\omega_1 = \omega_c \cdot 2^{-\Delta/2}$$
$$\omega_2 = \omega_c \cdot 2^{\Delta/2}$$

prewarp its boundaries separately:

$$\tilde{\omega}_1 = \mu(\omega_1)$$
$$\tilde{\omega}_2 = \mu(\omega_2)$$

thereby obtaining the new prewarped "passband" of a different bandwidth and center frequency:

$$\tilde{\omega}_c = \sqrt{\tilde{\omega}_1 \tilde{\omega}_2}$$
$$\tilde{\Delta} = \log_2 \frac{\tilde{\omega}_2}{\tilde{\omega}_1}$$

2. Given the new bandwidth and assuming a unit center frequency, construct the allpass $H(s)$ as previously described in this section.

3. Apply the cutoff substitution $s \leftarrow s/\tilde{\omega}_c$ to $H(s)$, which effectively means multiplying the cutoffs of the underlying 1-poles by the new center frequency $\tilde{\omega}_c$.

This approach effectively implements an idea similar to the usage of a single prewarping point discussed in Section 3.8, which takes care of preserving the correct ratios between the cutoffs of the individual filters in the system. In principle it could be okay to prewarp each of the 1-pole factors of $H(s)$ individually instead, however that apparently will somewhat destroy the optimality of the equiripple $\varphi(\omega)$.

Phase splitter

Half (or approximately half, if N is odd) of the poles of $H(s)$ are unstable and we can't implement $H(s)$ directly. However, there is one trick which allows to work around this limitation. Before describing this trick we will switch the notation back from $H(s)$ to $H_{-90}(s)$ to highlight the fact that the filter performs a $-90°$ phase shift (of the positive frequencies).

Let's factor $H_{-90}(s)$ into a product of two allpasses:

$$H_{-90}(s) = H_+(s) H_-(s)$$

where $H_+(s)$ contains only the right-semiplane (unstable) poles and $H_-(s)$ contains only the left-semiplane (stable) poles. As usual, we could assume or require that $H_+(0) = 1$ and $H_-(0) = 1$, which is achievable, given $\varphi(\omega) = 0$ and respectively $H(0) = 1$.

Given a signal $x(t) = e^{st}$ we wish to obtain the signal $y(t) = H_{-90}(s)x(t)$. Consider two other signals:

$$x'(t) = H_+^{-1}(s)x(t)$$
$$y'(t) = H_+^{-1}(s)y(t) = H_-(s)x(t)$$

where $H_+^{-1}(s) = 1/H_+(s)$. Notice that H_+^{-1} is a stable allpass and so is apparently H_-, thus $x'(t)$ and $y'(t)$ can be obtained from $x(t)$ by processing $x(t)$ by stable allpasses H_+^{-1} and H_-. Notice that

$$y'(t) = H_-(s)x(t) = H_-(s)H_+(s)x'(t) = H_{-90}(s)x'(t)$$

that is $y'(t)$ and $x'(t)$ are in a 90° phase shift relationship.

Apparently the same idea applies to abitrary $x(t)$, which we can express in the operator notation as

$$x'(t) = \hat{H}_+^{-1}x(t) \tag{10.58a}$$

$$y'(t) = \hat{H}_+^{-1}y(t) = \hat{H}_-x(t) \tag{10.58b}$$

$$y'(t) = \hat{H}_{-90}x'(t) \tag{10.58c}$$

where $\hat{H}_{-90}$ is the operator denoting the processing of a signal by the filter H_{-90}. Thus, even though we cannot phase-shift the input signal x by 90°, we can obtain two derived allpass signals x' and y', where the phase difference between x' and y' is 90°. Respectively, the combined signal

$$\begin{aligned} x_{>0}(t) &= \frac{x'(t) + jy'(t)}{2} = \frac{\hat{H}_+^{-1} + j\hat{H}_-}{2}x(t) = \\ &= \hat{H}_+^{-1}\frac{x(t) + jy(t)}{2} = \hat{H}_+^{-1}\frac{1 + j\hat{H}_{-90}}{2}x(t) \end{aligned} \tag{10.59}$$

is an analytic version of $x(t)$, where the phase shift of this analytic version relatively to $x(t)$ is defined by H_+^{-1}. The approach of generating two allpass signals which are in a 90° phase relationship is referred to as *phase splitting* and is illustrated in Fig. 10.30. Since $x'/2$ is the real part of the analytic signal and $y'/2$ is the imaginary part, the allpass H_+^{-1} produces the (doubled) real part and the allpass H_- produces the (doubled) imaginary part and therefore we can refer to H_+^{-1} and H_- as real and imaginary allpasses respectively.

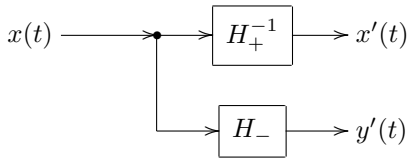


Figure 10.30: Phase splitter.

Notice that (10.59) is essentially the same as we have in (10.43), where H_o and H_e are corresponding to H_+^{-1} and H_- (where which specific filter corresponds to which depends on the order N). Thus (10.43) also describes a phase splitter, just obtained from a different angle.

10.12 Frequency shifter

Even though frequency shifter is not a filter in the strict sense, its most critical part will be based around the Hilbert transformer, which *is* a filter. For that reason the discussion of frequency shifters may belong to the filter topic.

Suppose we are given a signal $x(t)$ represented by its complex spectrum:

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

By multiplying the signal $x(t)$ with a complex sinusoidal signal $e^{j\Delta\omega\cdot t}$ we effectively shift the frequencies of all partials by $\Delta\omega$:

$$\begin{aligned} y(t) = e^{j\Delta\omega\cdot t} x(t) = e^{j\Delta\omega\cdot t} \int_{-\infty}^{\infty} X(\omega) e^{j\omega t} \frac{\mathrm{d}\omega}{2\pi} = \int_{-\infty}^{\infty} X(\omega) e^{j\Delta\omega\cdot t} e^{j\omega t} \frac{\mathrm{d}\omega}{2\pi} = \\ = \int_{-\infty}^{\infty} X(\omega) e^{j(\omega+\Delta\omega)t} \frac{\mathrm{d}\omega}{2\pi} \end{aligned} \tag{10.60}$$

This is not very interesting, since given a real $x(t)$ we obtain a complex $y(t)$. Obviously, it's because we multiplied by the complex signal $e^{j\Delta\omega\cdot t}$. In terms of signal spectra, the spectrum of $x(t)$ was Hermitian, however by shifting the spectrum by $\Delta\omega$ we destroyed the Hermitian propetry.

However, this is also not exactly what we want if we think of frequency shifting. The complex spectrum is a more or less purely mathematical concept, while the one more intuitively related to our hearing of sounds is the real spectrum, and it's the partials of the real spectrum whose frequencies we'd rather want to shift. That is, given

$$x(t) = \int_{0}^{\infty} a(\omega) \cos\bigl(\omega t + \varphi(\omega)\bigr) \frac{\mathrm{d}\omega}{2\pi}$$

we wish to obtain

$$y(t) = \int_{0}^{\infty} a(\omega) \cos\bigl((\omega + \Delta\omega)t + \varphi(\omega)\bigr) \frac{\mathrm{d}\omega}{2\pi} \tag{10.61}$$

Notably, if $\Delta\omega < 0$, then some of the frequencies $\omega + \Delta\omega$ in (10.61) will be negative and will alias with the positive frequencies of the same absolute magnitude. This can be either ignored, or $x(t)$ can be prefiltered to make sure it doesn't contain frequencies below $-\Delta\omega$. So, except for the just mentioned highpass prefiltering option, the possible aliasing of the negative frequencies doesn't affect the subsequent discussion.

We can rewrite (10.61) as

$$\begin{aligned} y(t) &= \int_{0}^{\infty} a(\omega) \cos\bigl((\omega + \Delta\omega)t + \varphi(\omega)\bigr) \frac{\mathrm{d}\omega}{2\pi} = \\ &= \int_{0}^{\infty} a(\omega) \cos\bigl(\Delta\omega t + \omega t + \varphi(\omega)\bigr) \frac{\mathrm{d}\omega}{2\pi} = \\ &= \int_{0}^{\infty} a(\omega) \Bigl(\cos\Delta\omega t \cos\bigl(\omega t + \varphi(\omega)\bigr) - \sin\Delta\omega t \sin\bigl(\omega t + \varphi(\omega)\bigr) \Bigr) \frac{\mathrm{d}\omega}{2\pi} = \\ &= \cos\Delta\omega t \cdot \int_{0}^{\infty} a(\omega) \cos\bigl(\omega t + \varphi(\omega)\bigr) \frac{\mathrm{d}\omega}{2\pi} - \\ &\qquad - \sin\Delta\omega t \cdot \int_{0}^{\infty} a(\omega) \sin\bigl(\omega t + \varphi(\omega)\bigr) \frac{\mathrm{d}\omega}{2\pi} = \\ &= \cos\Delta\omega t \cdot \int_{0}^{\infty} a(\omega) \cos\bigl(\omega t + \varphi(\omega)\bigr) \frac{\mathrm{d}\omega}{2\pi} - \end{aligned}$$

$$- \sin\Delta\omega t \cdot \int_0^\infty a(\omega)\cos\left(\omega t + \varphi(\omega) - \frac{\pi}{2}\right)\frac{\mathrm{d}\omega}{2\pi} =$$
$$= x(t)\cos\Delta\omega t - x_{-90}(t)\sin\Delta\omega t \tag{10.62}$$

where

$$x_{-90}(t) = \int_0^\infty a(\omega)\cos\left(\omega t + \varphi(\omega) - \frac{\pi}{2}\right)\frac{\mathrm{d}\omega}{2\pi}$$

is a signal obtained from $x(t)$ by phase-shifting all partials by $-90°$. In the operator notation the same can be expressed as

$$y(t) = \cos\Delta\omega t \cdot x(t) - \sin\Delta\omega t \cdot \hat{H}_{-90}x(t) = \left(\cos\Delta\omega t - \sin\Delta\omega t \cdot \hat{H}_{-90}\right)x(t) \tag{10.63}$$

We have already found out how to obtain a $-90°$ phase shifted signal in Section 10.11, except that we also found than such signal cannot be directly obtained. We will address this slightly later, while for now we shall take a different look at the same problem of frequency shifting.

Analytic signal approach

Looking again at (10.60) we can notice that the positive frequency partials are correctly shifted and it's the negative frequency partials which make trouble. So, if the negative partials weren't there in the first place:

$$x_{>0}(t) = \int_0^\infty X(\omega)e^{j\omega t}\frac{\mathrm{d}\omega}{2\pi}$$

we would have obtained

$$y_{>0}(t) = e^{j\Delta\omega\cdot t}x_{>0}(t) = e^{j\Delta\omega\cdot t}\int_0^\infty X(\omega)e^{j\omega t}\frac{\mathrm{d}\omega}{2\pi} = \int_0^\infty X(\omega)e^{j\Delta\omega\cdot t}e^{j\omega t}\frac{\mathrm{d}\omega}{2\pi} =$$
$$= \int_0^\infty X(\omega)e^{j(\omega+\Delta\omega)t}\frac{\mathrm{d}\omega}{2\pi} \tag{10.64}$$

Comparing (10.64) to (10.61) we notice that they essentially consist of the same frequency partials, except that $y_{>0}(t)$ is missing the negative part of its spectrum. The negative part of the spectrum can be restored by (10.40), and thus (10.64) and (10.61) are related via

$$y(t) = 2\,\mathrm{Re}\,y_{>0}(t)$$

This is easier to see in the operator notation:

$$y(t) = 2\,\mathrm{Re}\,y_{>0}(t) = 2\,\mathrm{Re}\left(e^{j\Delta\omega t}x_{>0}(t)\right) = 2\,\mathrm{Re}\left(e^{j\Delta\omega t}\hat{H}_{>0}x(t)\right) =$$
$$= 2\,\mathrm{Re}\left(e^{j\Delta\omega t}\frac{1+j\hat{H}_{-90}}{2}x(t)\right) = \mathrm{Re}\left(e^{j\Delta\omega t}(1+j\hat{H}_{-90})\right)x(t) =$$
$$= \mathrm{Re}\left(('cos\Delta\omega t + j\sin\Delta\omega t)(1+j\hat{H}_{-90})\right)x(t) =$$
$$= \left(\cos\Delta\omega t - \sin\Delta\omega t\hat{H}_{-90}\right)x(t)$$

which is identical to (10.63), thus both approaches are equivalent.

Implementation

Let $\hat{H}_+^{-1}$ be the allpass from (10.58). Multiplying (10.63) by $\hat{H}_+^{-1}$ we obtain

$$\begin{aligned}\hat{H}_+^{-1}y(t) &= \hat{H}_+^{-1}\left(\cos\Delta\omega t - \sin\Delta\omega t\cdot\hat{H}_{-90}\right)x(t) = \\ &= \left(\cos\Delta\omega t\cdot\hat{H}_+^{-1} - \sin\Delta\omega t\cdot\hat{H}_{-90}\hat{H}_+^{-1}\right)x(t) = \\ &= \left(\cos\Delta\omega t\cdot\hat{H}_+^{-1} - \sin\Delta\omega t\cdot\hat{H}_-\right)x(t) \qquad (10.65)\end{aligned}$$

If we are willing to accept the phase-shifted signal $\hat{H}_+^{-1}y(t)$ instead of $y(t)$ (and as it seems, we don't have much other choice) a frequency shifter can be simply implemented by the structure in Fig. 10.31.

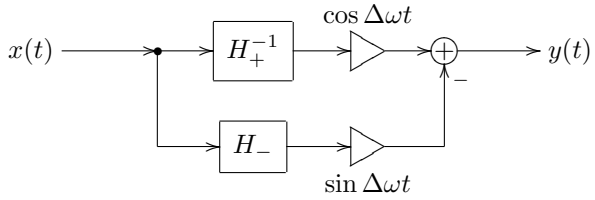


Figure 10.31: Frequency shifter.

Notably, replacing $\Delta\omega$ by $-\Delta\omega$ in (10.65) we obtain

$$\begin{aligned}\hat{H}_+^{-1}y(t) &= \hat{H}_+^{-1}\left(\cos\Delta\omega t + \sin\Delta\omega t\cdot\hat{H}_{-90}\right)x(t) = \\ &= \left(\cos\Delta\omega t\cdot\hat{H}_+^{-1} + \sin\Delta\omega t\cdot\hat{H}_-\right)x(t) \qquad (10.66)\end{aligned}$$

This means that we can extend the frequency shifter in Fig. 10.31 to a one that shifts simultaneously in both directions, obtaining the diagram in Fig. 10.32.[26]

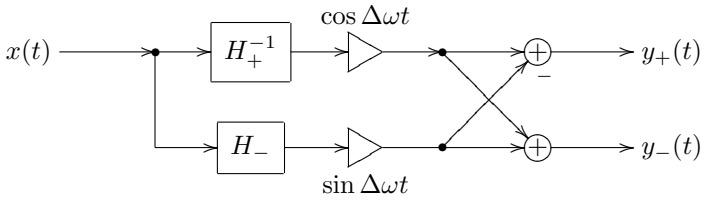


Figure 10.32: A bidirectional frequency shifter.

Adding together the frequency-shifted signals from (10.65) and (10.66) we notice that

$$\hat{H}_+^{-1}\left(\cos\Delta\omega t - \sin\Delta\omega t\cdot\hat{H}_{-90}\right) +$$

[26]The signal notations y_+ and $y_=$ denote the positive- and negative-shifted signals respectively and shouldn't be confused with the "+" and "−" subscripts of H_+^{-1} and H_- which denote the stable and unstable poles.

$$+ \hat{H}_+^{-1} \left(\cos\Delta\omega t + \sin\Delta\omega t \cdot \hat{H}_{-90} \right) = \hat{H}_+^{-1} \cdot 2\cos\Delta\omega t$$

or

$$\left(\cos\Delta\omega t \cdot \hat{H}_+^{-1} - \sin\Delta\omega t \cdot \hat{H}_- \right) + \\ + \left(\cos\Delta\omega t \cdot \hat{H}_+^{-1} + \sin\Delta\omega t \cdot \hat{H}_- \right) = \hat{H}_+^{-1} \cdot 2\cos\Delta\omega t$$

That is, the sum of y_+ and y_- in Fig. 10.32 essentially produces the ring modulation of $x(t)$ by $\cos\Delta\omega t$, except that the result of this ring modulation is doubled and phase-shifted by $\hat{H}_+^{-1}$. So frequency-shifting and ring-modulation by a sinusoid seem are very closely related. The same can be analyzed in the complex spectral domain:

$$\begin{aligned} \cos\Delta\omega t \cdot x(t) &= \frac{e^{j\Delta\omega t} + e^{-j\Delta\omega t}}{2} \int_{-\infty}^{\infty} X(\omega) e^{j\omega t} \frac{\mathrm{d}\omega}{2\pi} = \\ &= \frac{1}{2} \int_{-\infty}^{\infty} X(\omega) e^{j\omega t} e^{j\Delta\omega t} \frac{\mathrm{d}\omega}{2\pi} + \frac{1}{2} \int_{-\infty}^{\infty} X(\omega) e^{j\omega t} e^{-j\Delta\omega t} \frac{\mathrm{d}\omega}{2\pi} = \\ &= \frac{1}{2} \int_{-\infty}^{\infty} X(\omega) e^{j(\omega+\Delta\omega)t} \frac{\mathrm{d}\omega}{2\pi} + \frac{1}{2} \int_{-\infty}^{\infty} X(\omega) e^{j(\omega-\Delta\omega)t} \frac{\mathrm{d}\omega}{2\pi} \end{aligned}$$

Thus in the case of the ring modulation by a sinusoid, the partials are frequency-shifted in both directions.

Aliasing

If $\Delta\omega > 0$ then for some partials the sum $\omega + \Delta\omega$ may exceed the Nyquist frequency, respectively they will alias to $2\pi - (\omega + \Delta\omega)$ (assuming unit sampling period $T = 1$). This kind of aliasing is similar to the one occurring at $\omega + \Delta\omega < 0$ in case of $\Delta\omega < 0$, however, while the aliasing around $\omega = 0$ also occurs in the analog case, aliasing around Nyquist frequency is a purely digital phenomenon.

It is therefore up to the effect designer, whether the aliasing around $\omega = 0$ should be prevented, or allowed. The aliasing at Nyquist is however usually undersired. It can be avoided by prefiltering the frequency band $[\pi - \Delta\omega, \pi]$, which can be done by a lowpass filter with a cutoff around $\pi - \Delta\omega$. Notice that $\pi - \Delta\omega$ is a discrete-time cutoff value and thus doesn't need prewarping.

The aliasing around $\omega = 0$ can be prevented in a similar way by using a highpass with a cutoff at $-\Delta\omega$ (since in this case we assume $\Delta\omega < 0$, the cutoff will thereby be positive). Note that since the phase splitter has a limited bandwidth, one also may consider filtering out the signal outside that bandwidth anyway, regardless of $\Delta\omega$.

10.13 Remez algorithm

The equiripple behavior of Chebyshev polynomials and elliptic rational functions is a characteristic feature of the so-called *minimax approximations*. T_N, $\mathit{\Lambda}_N$, R_N, $\bar{R}_N$ and the function $\Phi(\omega)$ used to build the phase splitter all provide specific analytic-form solutions to specific minimax problems. However, in a more general situation we might want a numerical solution approach.[27]

[27] The description of Remez algorithm (which is a numerical minimax optimization algorithm) was included into earlier revisions of this book as an alternative to the use of elliptic

Suppose we are given a function $f(x)$ and its approximation $\tilde{f}(x)$. There are different ways to measure the quality of the approximation. One way to measure this quality is the maximum error of the approximation on the given interval of interest $x \in [a, b]$:

$$E = \max_{[a,b]} \left| \tilde{f}(x) - f(x) \right| \tag{10.67}$$

We therefore wish to minimize the value of E. That is we want to minimize the maximum error of the approximation. Such approximations are hence called *minimax* approximations.[28]

Gradient search methods do not work well for minimax optimizations. Therefore a different method, called *Remez algorithm*,[29] needs to be used. As of today, internet resources concerning the Remez algorithm seem quite scarce, nor does this method seem to be a subject of common math textbooks. This might suggest that Remez algorithm belongs to a rather esoteric math area. The algorithm itself, however, is very simple. We will therefore cover the essentials of that algorithm in this book.[30]

Suppose $\tilde{f}(x)$ is a polynomial:

$$\tilde{f}(x) = \sum_{n=0}^{N} a_n x^n \tag{10.68}$$

Apparently, there are $N+1$ degrees of freedom in the choice of $\tilde{f}(x)$, each degree corresponding to one of the coefficients a_n. Therefore we can force the function $\tilde{f}(x)$ to take arbitrarily specified values at $N+1$ arbitrarily chosen points $\bar{x}_n$. Particularly, we can require

$$\tilde{f}(\bar{x}_n) = f(\bar{x}_n) \qquad n = 0, \ldots, N$$

or equivalently require the error to be zero at $\bar{x}_n$:

$$\tilde{f}(\bar{x}_n) - f(\bar{x}_n) = 0 \qquad n = 0, \ldots, N \tag{10.69}$$

(notice that the equations (10.69) are linear in respect to the unknowns a_n and therefore are easily solvable). If the points $\bar{x}_n$ are approximately uniformly spread over the interval of interest $[a, b]$ then intuitively we can expect $\tilde{f}(x)$ to be a reasonably good approximation of $f(x)$ (Fig. 10.33).

---

functions to construct phase splitters. Now that the book is strongly focusing on elliptic functions anyway, the discussion of Remez algorithm might feel almost redundant. However the author felt that this is still quite valuable resource to be simply dropped from the book. Particularly, Remez algorithm is useful for building low-cost approximations of functions, although, depending on the context, minimax solutions are not necessarily the best ones for a given purpose.

[28] The maximum of the absolute value of a function is also the L_∞ norm of the function. Therefore minimax approximations are optimizations of the L_∞ norm.

[29] The Remez algorithm should not be confused with the Parks–McClellan algorithm. The latter is a specific restricted version of the former. For whatever reason, the Parks–McClellan algorithm is often referred to as the Remez algorithm in the signal processing literature.

[30] The author's primary resource for the information about the Remez algorithm was the documentation for the math toolkit of the *boost* library by J.Maddock, P.A.Bristow, H.Holin and X.Zhang.

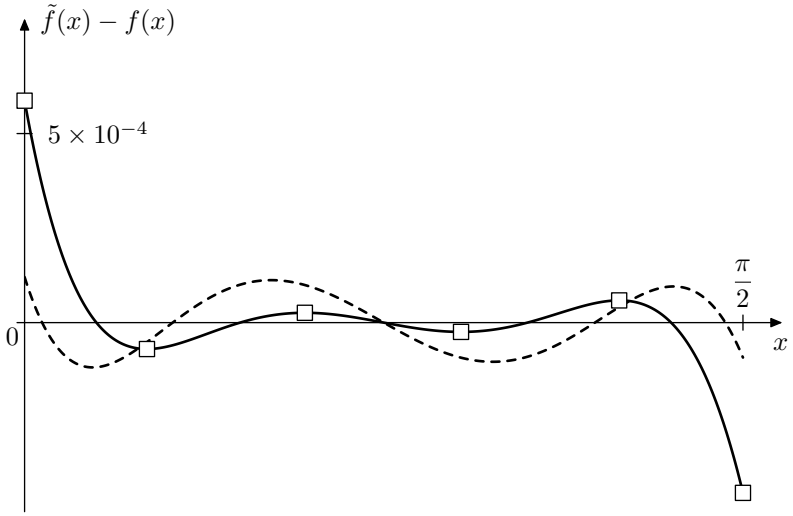


Figure 10.33: The error of the 4-th order polynomial approximations of $\sin x$ on $[0, \pi/2]$. The approximation with uniformly spaced zeros at 9°, 27°, 45°, 63°, 81° (solid line) and the one with Chebyshev zeros (dashed line). The empty square-shaped dots at the extrema of the error are the control points of the Remez algorithm.

This based on the uniform zero spacing approximation is however not the best one. Indeed, instead let $\bar{x}_n$ equal the (properly scaled) zeros of the Chebyshev polynomial of order $N+1$:

$$\begin{aligned} \bar{x}_n &= \frac{a+b}{2} + \frac{b-a}{2} z_n \qquad\qquad \bar{x}_n \in (a,b) \qquad z_n \in (-1,1) \\ T_{N+1}(z_n) &= \cos\big((N+1)\arccos z_n\big) = 0 \\ z_n &= -\cos\frac{\frac{1}{2}+n}{N+1}\pi \qquad n = 0, \ldots, N \end{aligned}$$

where the minus sign in front of the cosine ensures that z_n are in ascending order. Comparing Chebyshev zeros approximation (the dashed line in Fig. 10.33) to the uniform zeros approximation, we can see that the former is much better than the latter, at least in the minimax sense.

A noticeable property of the Chebyshev zeros approximation clearly observable in Fig. 10.33 is that the extrema of the approximation error (counting the extrema at the boundaries of the interval $[a, b]$!) are approximately equal in absolute magnitude and have alternating signs. This is a characteristic trait of minimax approximations: the error extrema are equal in magnitude and alternating in sign.

So, we might attempt to build a minimax approximation by trying to satisfy the *equiripple error oscillation* requirement. That is, instead of seeking to minimize the maximum error, we simply seek an error which oscillates between the two boundaries of opposite sign and equal absolute value. Somewhat surprisingly, this is a much simpler task.

Intuitive description of Remez algorithm

Consider the solid line graph in Fig. 10.33. Intuitively, imagine a "control point" at each of the extrema. Now we "take" the control point which has the largest error (the one at $x = 0$) and attempt to move it towards the x axis, reducing the error value at $x = 0$. Since there are 6 control points (4 at local extrema plus 2 at the boundaries), but only 5 degrees of freedom (corresponding to the coefficients a_n), at least one of the other control points needs to move (or several or all of them can move). Intuitively it's clear that if we lower the error at $x = 0$, then it will grow at some other points of $[a, b]$. However, since we have the largest error at $x = 0$ anyway, we can afford the error growing elsewhere on $[a, b]$, at least for a while. Notice that during such change the x positions of control points will also change, since the extrema of the error do not have to stay at the same x coordinates.

As the error elsewhere at $[a, b]$ becomes equal in absolute magnitude to the one at $x = 0$, we have two largest-error control points which need to be moved simultaneously from now on. This can be continued until only one "free" control point remains. Simultaneously reducing the error at 5 of 6 control points we thereby increase the error at the remaining control point. At some moment both errors will become equal in absolute magnitude, which means that the error at all control points is equal in absolute magnitude. Since the control points are located at the error extrema, we have thereby an equiripple oscillating error.

Remez algorithm for polynomial approximation

Given $\tilde{f}(x)$ which is a polynomial (10.68), the process of "pushing the control points towards zero" has a simple algorithmic expression. Indeed, we seek $\tilde{f}(x)$ which satisfies

$$\tilde{f}(\hat{x}_n) + (-1)^n \varepsilon = f(\hat{x}_n) \qquad n = 0, \ldots, N+1 \tag{10.70}$$

where $\hat{x}_n$ are the (unknown) control points (including $\hat{x}_0 = a$ and $\hat{x}_{N+1} = b$) and ε is the (unknown) signed maximum error. Thus, the unknowns in (10.70) are a_n (the polynomial coefficients), $\hat{x}_n$ (the control points at the extrema) and ε (the signed maximum error). Notice that the equations (10.70) are linear in respect to a_n and ε, which leads us to the following idea.

Suppose we already have some initial guess for $\tilde{f}(x)$, like the uniform zero polynomial in Fig. 10.33 (or the Chebyshev zero polynomial, which is even better). Identifying the extrema of $\tilde{f}(x) - f(x)$ we obtain a set of control points $\hat{x}_n$. Now, given these $\hat{x}_n$, we simply solve (10.70) for a_n and ε (where we have $N + 2$ equations and $N + 2$ unknowns in total), thereby obtaining a new set of a_n. In a way this is cheating, because $\hat{x}_n$ are not the control points anymore, since they are not anymore the extrema of the error (and if they were, we would already have obtained a minimax approximation by simply finding these new a_n). However, the polynomial defined by the new a_n has a much better maximum error (Fig. 10.34)!

So we simply update the control points $\hat{x}_n$ to the new positions of the extrema and solve (10.70) again. Then again update the control points and solve (10.70) and so on. This is the Remez algorithm for polynomial approximation. We still need to refine some details about the algorithm though.

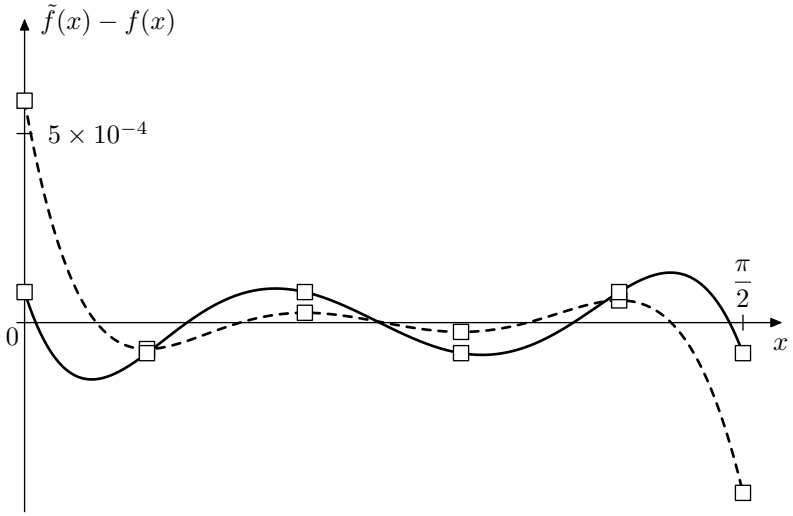


Figure 10.34: The approximation error before (dashed line) and after (solid line) a single step of the Remez polynomial approximation algorithm. The empty square-shaped dots are the control points.

- The function $f(x)$ should be reasonably well-behaved (whatever that could mean) in order for Remez algorithm to work.

- As a termination condition for the iteration we can simply check the equiripple property of the error at the control points. That is, having obtained the new a_n, we find the new control points $\hat{x}_n$ and then compute the errors $\varepsilon_n = \tilde{f}(\hat{x}_n) - f(\hat{x}_n)$. If the absolute values of ε_n are equal up to the specified precision, this means that we have an approximation which is minimax up to the specified error, and the algorithm may be stopped.

- The initial approximation $\tilde{f}(x)$ needs to have the alternating sign property. This is more or less ensured by using (10.69) to construct the initial approximation. A good choice for $\bar{x}_n$ (as demonstrated by Fig. 10.33) are the roots of the Chebyshev polynomial of order one higher than the order of the approximating polynomial $\tilde{f}(x)$.[31]

- The control points $\hat{x}_n$ are the zeros of the error derivative $(\tilde{f} - f)'$ (except for $\hat{x}_0 = a$ and $\hat{x}_{N+1} = b$). There is exactly one local extremum on each interval $(\bar{x}_n, \bar{x}_{n+1})$ between the zeros of the error. Therefore, $\hat{x}_{n+1}$ can be simply found as the zeros of the error derivative by bisection of the intervals $(\bar{x}_n, \bar{x}_{n+1})$.

- After having obtained new a_n, the old control points $\hat{x}_n$ are not the extrema anymore, however the errors at $\hat{x}_n$ are still alternating in sign. Therefore the new zeros $\bar{x}_n$ (needed to find the new control points by bisection) can be found by bisection of the intervals $(\hat{x}_n, \hat{x}_{n+1})$.

[31] This becomes kind of intuitive after considering Chebyshev polynomials as *some kind* of minimax approximations of the zero constant function $f(x) \equiv 0$ on the interval $[-1, 1]$.

Restrictions and variations

Often it is desired to obtain a function which is odd or even, or has some other restrictions. This can be done by simply fixing the respective a_n, thereby reducing the number of control variables a_n and reducing the number of control points $\hat{x}_n$ and zero crossings $\bar{x}_n$ accordingly.

Remez algorithm can also be easily modified to accommodate a weight function in the minimax norm (10.67):

$$E = \max_{[a,b]} \left(W(x) \cdot \left| \tilde{f}(x) - f(x) \right| \right) \qquad W(x) > 0$$

The error function therefore turns into $W(x)(\tilde{f}(x) - f(x))$, while the minimax equations (10.70) turn into

$$\tilde{f}(\hat{x}_n) + (-1)^n W^{-1}(\hat{x}_n)\varepsilon = f(\hat{x}_n) \qquad n = 0, \ldots, N+1$$

(where $W^{-1}(x)$ is the reciprocal of $W(x)$).

Remez algorithm for rational approximation

Instead of using a polynomial $\tilde{f}(x)$, better approximations can be often achieved by rational $\tilde{f}(x)$:

$$\tilde{f}(x) = \frac{\sum_{n=0}^{N} a_n x^n}{1 + \sum_{n=1}^{M} b_n x^n} \tag{10.71}$$

Besides being able to deliver better approximations in certain cases, rational functions can be often useful for obtaining approximations on infinite intervals such as $[a, +\infty)$, because by varying the degrees of the numerator and denominator the asymptotic behavior of $\tilde{f}(x)$ at $x \to \infty$ can be controlled.

For a rational $\tilde{f}(x)$ defined by (10.71) the minimax equations (10.70) become nonlinear in respect to the unknowns ε and b_n, although they are still linear in respect to the unknowns a_n:

$$\sum_{i=0}^{N} a_i \hat{x}_n^i + (-1)^n \left(1 + \sum_{i=1}^{M} b_i \hat{x}_n^i \right) \varepsilon = \left(1 + \sum_{i=1}^{M} b_i \hat{x}_n^i \right) f(\hat{x}_n) \qquad n = 0, \ldots, N+M+1 \tag{10.72}$$

Notice that the number of degrees of freedom is now $N+M+1$. The equations (10.72) can be solved using different numeric methods for nonlinear equation solution, however there is one simple trick.[32] Rewrite (10.72) as

$$\sum_{i=0}^{N} a_i \hat{x}_n^i + (-1)^n \varepsilon \sum_{i=1}^{M} b_i \hat{x}_n^i + (-1)^n \varepsilon = \left(1 + \sum_{i=1}^{M} b_i \hat{x}_n^i \right) f(\hat{x}_n)$$

[32] This trick is adapted from the *boost* library documentation and sources.

Now we pretend we don't know the free term ε, but we do know the value of ε before the sum of $b_i \hat{x}_n^i$:

$$\sum_{i=0}^{N} a_i \hat{x}_n^i + (-1)^n \varepsilon_0 \sum_{i=1}^{M} b_i \hat{x}_n^i + (-1)^n \varepsilon = \left(1 + \sum_{i=1}^{M} b_i \hat{x}_n^i \right) f(\hat{x}_n) \qquad (10.73)$$

where ε_0 is this "known" value of ε. The value of ε_0 can be estimated e.g. as the average absolute error at the control points $\hat{x}_n$. Then (10.73) are linear equations in respect to a_n, b_n and ε and can be easily solved. Having obtained the new a_n and b_n, we can obtain a new estimation for ε_0 and solve (10.73) again. We repeat until the errors $\tilde{f}(\hat{x}_n) - f(\hat{x}_n)$ at the control points $\hat{x}_n$ become equal in absolute vlaue up to a necessary precision. At this point we can consider the solution of (10.72) as being obtained to a sufficient precision and proceed with the usual Remez algorithm routine (find the new $\bar{x}_n$, new $\hat{x}_n$ etc.)

Here are some further notes.

- In principle the solution of (10.72) doesn't need to be obtained to a very high precision, except in the final step of the Remez algorithm. However, in order to know whether the current step is the final one or not, we need to know the true control points, so that we can estimate how well the equiripple condition is satisfied. Ultimately, this is a question of the computational expense of finding the new control points vs. computing another iteration of (10.73).

- Sometimes, if the equations are strongly nonlinear, the trick (10.73) may fail to converge. In this case one could attempt to use the discussed below more general Newton–Raphson approach (10.79), where the damping parameter may be used to mitigate the convergence problems.

- In regards to the problem of choice of the initial $\tilde{f}(x)$ for the rational Remez approximation, notice that the zero error equations (10.69) take the form

$$\sum_{n=0}^{N} a_n \bar{x}^n = f(\bar{x}_n) \left(1 + \sum_{n=1}^{M} b_n \bar{x}^n \right)$$

  which is fully linear in respect to a_n and b_n, and can be easily solved.

Other kinds of approximating functions

In certain cases one could use even more complicated forms of $\tilde{f}(x)$, which are neither polynomial nor rational. In the general case such function $\tilde{f}(x)$ is controlled by a number of parameters a_n:

$$\tilde{f}(x) = \tilde{f}(x, a_1, a_2, \ldots, a_N)$$

(notice that this time the numbering of a_n is starting at one, so that there are N parameters in total, giving N degrees of freedom). The minimax equations (10.70) become

$$\tilde{f}(\hat{x}_n, a_1, a_2, \ldots, a_N) + (-1)^n \varepsilon = f(\hat{x}_n) \qquad n = 0, \ldots, N \qquad (10.74)$$

Introducing functions

$$\phi_n(a_1, a_2, \ldots, a_N, \varepsilon) = \tilde{f}(\hat{x}_n, a_1, a_2, \ldots, a_N) + (-1)^n \varepsilon - f(\hat{x}_n)$$

we rewrite the equations (10.74) as

$$\phi_n(a_1, a_2, \ldots, a_N, \varepsilon) = 0 \qquad n = 0, \ldots, N \tag{10.75}$$

Introducing vector notation

$$\mathbf{\Phi} = \begin{pmatrix} \phi_0 & \phi_1 & \ldots & \phi_N \end{pmatrix}^\mathsf{T}$$
$$\mathbf{a} = \begin{pmatrix} a_1 & a_2 & \ldots & a_N & \varepsilon \end{pmatrix}^\mathsf{T}$$

we rewrite (10.75) as

$$\mathbf{\Phi}(\mathbf{a}) = 0 \tag{10.76}$$

Apparently, (10.76) is a vector form of (10.70), except that now we consider it as a generally nonlinear equation. Both the function's argument $\mathbf{a}$ and the function's value $\mathbf{\Phi}(\mathbf{a})$ have the dimension $N+1$, therefore the equation (10.76) is fully defined.

Different numeric methods can be applied to solving (10.76). We will be particularly interested in the application of multidimensional Newton–Raphson method. Expanding $\Phi(\mathbf{a})$ into Taylor series at some fixed point $\mathbf{a}_0$ we transform (10.76) into:

$$\mathbf{\Phi}(\mathbf{a}_0) + \frac{\partial \mathbf{\Phi}}{\partial \mathbf{a}}(\mathbf{a}_0) \cdot \Delta\mathbf{a} + o(\Delta\mathbf{a}) = 0 \tag{10.77}$$

where $\partial\mathbf{\Phi}/\partial\mathbf{a}$ is the Jacobian matrix and $\mathbf{a} = \mathbf{a}_0 + \Delta\mathbf{a}$. By discarding the higher order terms $o(\Delta\mathbf{a})$, the equation (10.77) is turned into

$$\Delta\mathbf{a} = -\left(\frac{\partial \mathbf{\Phi}}{\partial \mathbf{a}}(\mathbf{a}_0)\right)^{-1} \cdot \mathbf{\Phi}(\mathbf{a}_0) \tag{10.78}$$

The equation (10.78) implies the Newton–Raphson iteration scheme

$$\mathbf{a}_{n+1} = \mathbf{a}_n - \alpha \cdot \left(\frac{\partial \mathbf{\Phi}}{\partial \mathbf{a}}(\mathbf{a}_n)\right)^{-1} \cdot \mathbf{\Phi}(\mathbf{a}_n) \tag{10.79}$$

where the damping factor α is either set to unity, or to a lower value, if the nonlinearity of $\mathbf{\Phi}(\mathbf{a})$ is too strong and prevents the iterations from convergening. The initial value $\mathbf{a}_0$ is obtained from the initial settings of the parameters a_n and the estimated initial value of ε. As for the rational $\tilde{f}(x)$, the initial value of ε can be estimated e.g. as the average error at the control points.

Similarly to the rational approximation case, the solution of (10.76) doesn't need to be obtained to a very high precision during the intermediate steps of the Remez algorithm. However the same tradeoff between computing the iteration step (10.79) and finding the new control points applies.

The choice of the initial $\tilde{f}(x)$ can be done based on the same principles. The zero error equations (10.69) turn into

$$\phi_n(a_1, a_2, \ldots, a_N, 0) = 0 \qquad n = 1, \ldots, N$$

(notice that compared to (10.75) we have set ε to zero and we have N rather than $N+1$ equations). Letting

$$\bar{\mathbf{\Phi}} = \begin{pmatrix} \phi_1 & \phi_2 & \ldots & \phi_N \end{pmatrix}^\mathsf{T}$$

$$\bar{\mathbf{a}} = \begin{pmatrix} a_1 & a_2 & \dots & a_N \end{pmatrix}^{\mathsf{T}}$$

we have an N-dimensional nonlinear equation

$$\bar{\mathbf{\Phi}}(\bar{\mathbf{a}}) = 0$$

which can be solved by the same Newton–Raphson method:

$$\bar{\mathbf{a}}_{n+1} = \bar{\mathbf{a}}_n - \alpha \cdot \left(\frac{\partial \bar{\mathbf{\Phi}}}{\partial \bar{\mathbf{a}}}(\bar{\mathbf{a}}_0) \right)^{-1} \cdot \bar{\mathbf{\Phi}}(\bar{\mathbf{a}}_0) \tag{10.80}$$

10.14 Numerical construction of phase splitter

For the sake of a demonstration example we are now going to use Remez algorithm to build an approximation of the ideal 90° allpass phase shifter defined by (10.48), while deliberately staying away from the entire framework of elliptic functions. The obtained results shall be identical to the ones previously obtained analytically.

We will retain the mentioned allpass property in the approximation, therefore let $H(s)$ denote the allpass which should approximate the ideal phase shifter (10.48). Using serial decomposition, $H(s)$ can be decomposed into series of 2- and 1-pole allpasses. Since we aim to have $H(s)$ with as flat (actually, constant in the range of interest) phase response as possible, 2-poles seem to be less useful than 1-poles, due to steeper phase responses of the former (Figs. 10.35 and 10.36).

Restricting ourselves to using just 1-poles we have:

$$H(s) = \prod_{n=1}^{N} A_n(s) = \prod_{n=1}^{N} \frac{\omega_n - s}{\omega_n + s} \tag{10.81}$$

where ω_n are the cutoffs of the 1-pole allpasses $A_n(s)$. Notice that the specific form of specifying $H(s)$ in (10.81) ensures $H(0) = 1\ \forall N$, that is we wish to have a 0° rather than −180° phase response at $\omega = 0$.

Now the idea is the following. Suppose $N = 0$ in (10.81) (that is we have no 1-pole allpasses in the serial decomposition yet). Adding the first allpass A_1 at the cutoff ω_1 we make the phase response of (10.81) equal to the one of a 1-pole allpass (Fig. 10.35). From $\omega = 0$ to $\omega = \omega_n$ the phase response is kind of what we expect it to be: it starts at $\arg H(0) = 0$ and then decreases to $\arg H(j\omega_n) = -\pi/2$. However, after $\omega = \omega_n$ it continues to decrease, which is not what we want. Therefore we insert another allpass A_2 with a *negative cutoff* $-\omega_2$:

$$H(s) = \frac{\omega_1 - s}{\omega_1 + s} \cdot \frac{-\omega_2 - s}{-\omega_2 + s} \qquad 0 < \omega_1 < \omega_2$$

Clearly, A_2 is unstable. However, we already know that unstable components of $H(s)$ are not a problem, since they simply go into the H_+^{-1} part of the phase splitter.

The phase response of a negative-cutoff allpass (Fig. 10.37) is the inversion of Fig. 10.35. Therefore, given sufficient distance between ω_1 and ω_2, the phase response of H will first drop below $-\pi/2$ (shortly after $\omega = \omega_1$) and then at some point turn around and grow back again (Fig. 10.38). Then we insert another

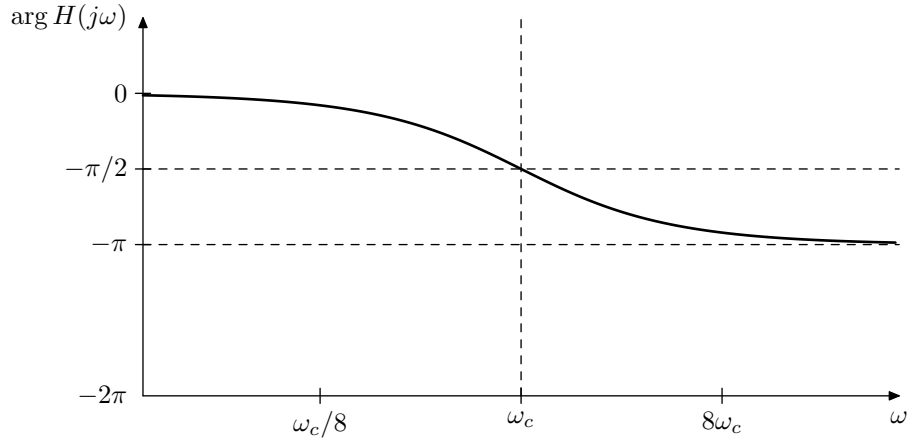


Figure 10.35: Phase response of a 1-pole allpass filter.

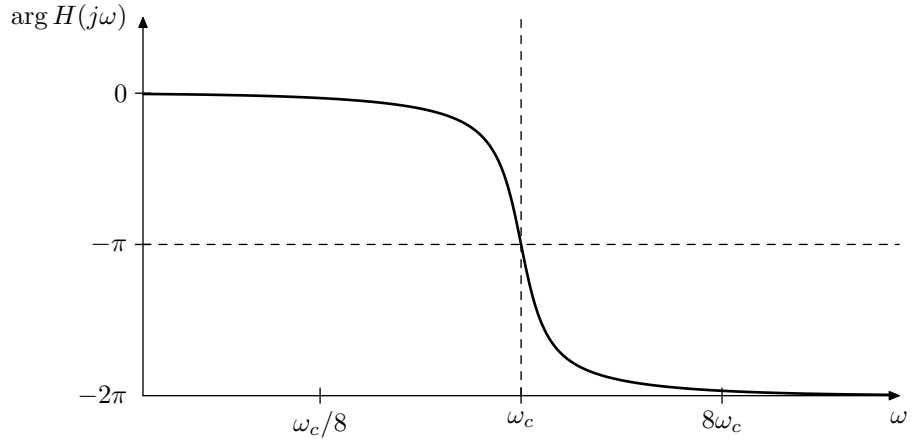


Figure 10.36: Phase response of a 2-pole allpass filter.

positive-cutoff allpass A_3, then a negative-cutoff allpass A_4 etc., obtaining if not an equiripple approximation of $-90°$ phase response, then something of a very similar nature (Fig. 10.39).

The curve in Fig. 10.39 has two obvious problems. The ripple amplitude is way too large. Furthermore, in order to obtain this kind of curve, we need to position the cutoffs ω_n pretty wide apart (4 octaves between the neighboring cutoffs is a safe bet). We would like to position the cutoffs closer together, thereby reducing the ripple amplitude, however the uniform spacing of the cutoffs doesn't work very well for denser spacings of the cutoffs. We need to find a way to identify the optimum cutoff positions.

Using cutoffs of alternating signs, we rewrite the transfer function expression (10.81) as

$$H(s) = \prod_{n=1}^{N} A_n(s) = \prod_{n=1}^{N} \frac{(-1)^{n+1}\omega_n - s}{(-1)^{n+1}\omega_n + s} \qquad 0 < \omega_1 < \omega_2 < \ldots < \omega_N \quad (10.82)$$

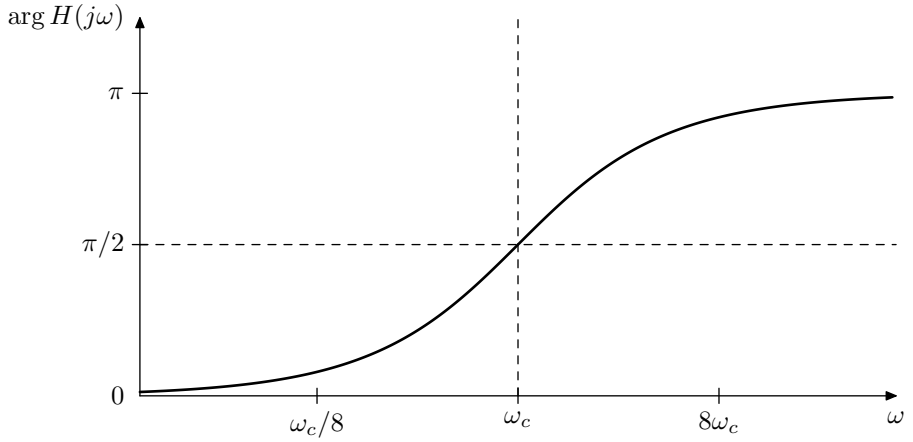


Figure 10.37: Phase response of a negative-cutoff 1-pole allpass filter.

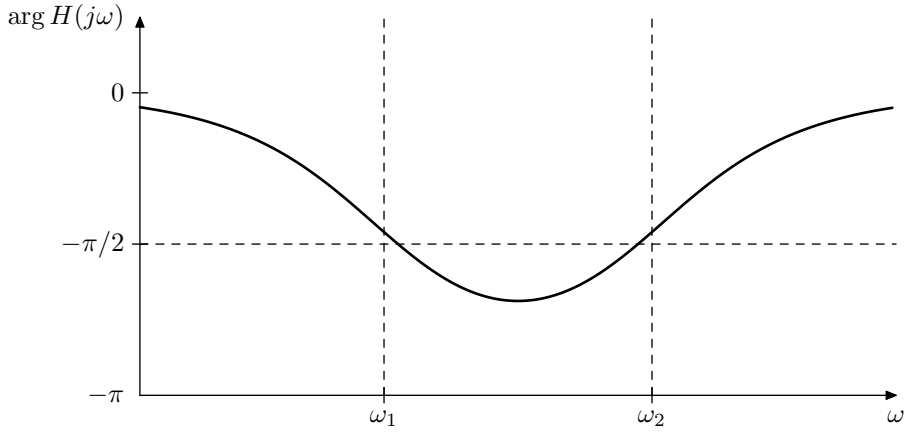


Figure 10.38: Phase response of a pair of a positive-cutoff and a negative-cutoff 1-pole allpass filters. Frequency scale is logarithmic.

(the cutoff of A_1 needs to be positive in order for the phase response of H to have a negative derivative at $\omega = 0$). Considering that the phase response of a 1-pole allpass with cutoff ω_c is

$$H(j\omega) = -2 \arctan \frac{\omega}{\omega_c}$$

the phase response of the serial decomposition (10.82) is

$$\varphi(x) = \arg H(j\omega) = 2 \sum_{n=1}^{N} (-1)^n \arctan \frac{\omega}{\omega_n} = 2 \sum_{n=1}^{N} (-1)^n \arctan e^{x - a_n} \quad (10.83)$$

$$\omega = e^x$$

$$\omega_n = e^{a_n}$$

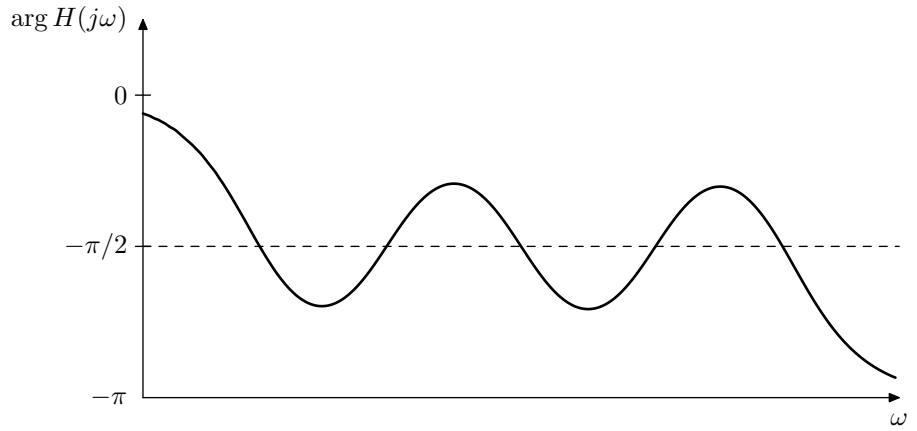


Figure 10.39: Phase response of a series of alternating positive-cutoff and negative-cutoff 1-pole allpass filters. Frequency scale is logarithmic.

where x and a_n are the logarithmic scale counterparts of ω and ω_n (essentially these are the pitch-scale values, we have just used e rather than 2 as the base to simplify the expressions of the derivatives of φ). The reason to use the logarithmic scale in (10.83) is that the phase responses of 1-pole allpasses are symmetric in the logarithmic scale, therefore the entire problem gets certain symmetry and uniformity.

Now we are in a position to specify the minimax approximation problem of construction of the phase shifter H_{-90}. We wish to find the minimax approximation of $f(x) \equiv -\pi/2$ on the specified interval $x \in [x_{\min}, x_{\max}]$, where the approximating function $\varphi(x)$ needs to be of the form (10.83).

The approximating function $\varphi(x)$ has N parameters:

$$\varphi(x) = \varphi(x, a_1, a_2, \ldots, a_N)$$

which can be found by using the Remez algorithm for approximations of general form. Notably, for larger N and smaller intervals $[x_{\min}, x_{\max}]$ the problem becomes more and more nonlinear, requiring smaller damping factors α in (10.79) and (10.80). The damping factors may be chosen by restricting the lengths $|\mathbf{a}_{n+1} - \mathbf{a}_n|$ and $|\bar{\mathbf{a}}_{n+1} - \bar{\mathbf{a}}_n|$ in (10.79) and (10.80).

In order to further employ the logarithmic symmetry of the problem (although this is not a must), we may require $x_{\min} + x_{\max} = 0$ corresponding to $\omega_{\min}\omega_{\max} = 1$. Then the following applies.

- Due to the symmetry $\omega_{\min}\omega_{\max} = 1$ the obtained cutoffs ω_n will also be symmetric: $\omega_n\omega_{N+1-n} = 1$. (Actually they will be symmetric relatively to $\sqrt{\omega_{\min}\omega_{\max}}$ no matter what the $\omega_{\min}$ and $\omega_{\max}$ are, but it's convenient to have this symmetry more explicitly visible.)

- Using this symmetry the number of cutoff parameters can be halved (for odd N the middle cutoff $\omega_{(N+1)/2}$ is always at unity and therefore can be also excluded from the set of varying parameters). Essentially we simply restrict $\varphi(x)$ to be an odd (for odd N) or even (for even N) function of x.

- The obtained symmetric range $[\omega_{\min}, \omega_{\max}]$ can be scaled by an arbitrary constant A by scaling the allpass cutoffs by the same constant:

$$[\omega_{\min}, \omega_{\max}] \leftarrow [A\omega_{\min}, A\omega_{\max}]$$
$$\omega_n \leftarrow A\omega_n$$

Figs. 10.40 and 10.41 contain example approximations of $H_{-90}(s)$ obtained by cutoff optimization (for the demonstration purposes, the approximation orders have been chosen relatively low, giving the phase ripple amplitude of an order of magnitude of 1°). The readers are encouraged to compare these pictures (qualitatively, since the specified filter orders and bandwidths do not match) to Fig. 10.28.

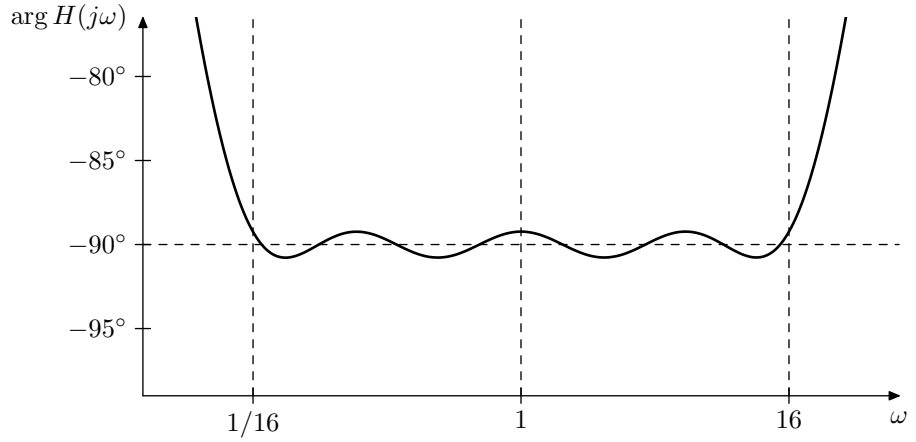


Figure 10.40: 8th-order minimax approximation of the ideal $H_{-90}(s)$.

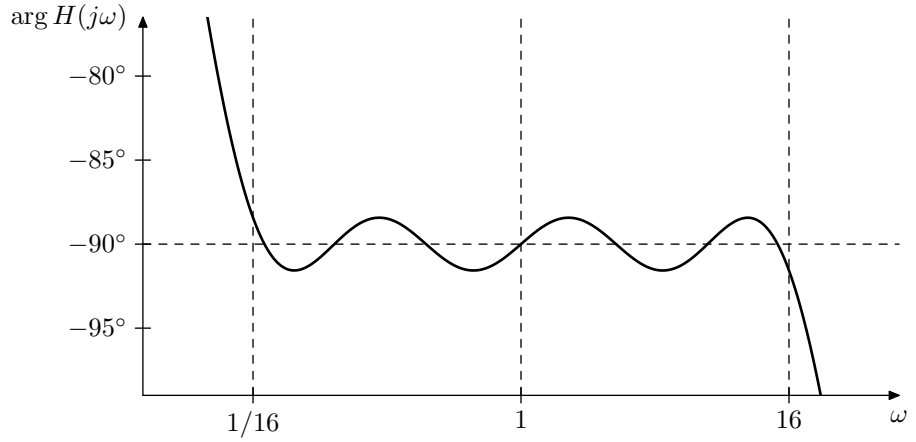


Figure 10.41: 7th-order minimax approximation of the ideal $H_{-90}(s)$.

Instead of solving the initial approximation equation (10.80) there is a different approach, which generally results in the nonlinearity of $\mathbf{\Phi}(\mathbf{a})$ not so strongly

affecting the algorithm convergence. We could take the manually constructed (10.82) with 4-octave spaced cutoffs $\omega_{n+1} = 16\omega_n$ as our initial approximation. The formal range of interest could contain two additional octaves on each side: $\omega_{\min} = \omega_1/4$, $\omega_{\max} = 4\omega_N$. Employing the logarithmic symmetry, we center the whole range around $\omega = 1$, so that $\omega_{\min}\omega_{\max} = 1$.

Using (10.79) (in the logarithmic scale x) we refine the initial approximation to the ripples of equal amplitude. Then we simply shrink the range a little bit. An efficient shrinking substitution is using the geometric averages:

$$\begin{aligned} \omega_{\min} &\leftarrow \sqrt{\omega_{\min}\omega_1} \\ \omega_{\max} &\leftarrow \sqrt{\omega_{\max}\omega_N} \end{aligned} \tag{10.84}$$

The substitution (10.84) doesn't affect the control points $\hat{x}_n$ or the zeros $\bar{x}_n$ of the Remez algorithm. Therefore after the substitution the Remez algorithm can be simply run again. Then the substitution is performed again, and so on, until we shrink the interval $[\omega_{\min}, \omega_{\max}]$ to the exact desired range.[33]

Notice that the approximations on the intermediate ranges $[\omega_{\min}, \omega_{\max}]$ do not need to be obtained with a very high precision, since their only purpose is to provide a starting point for the next application of the Remez algorithm on a smaller range. It is only the Remez algorithm on the exact desired range, which needs to be run to a high precision. This can noticeably improve the algorithm's running time.

SUMMARY

We have discussed various approaches to the construction of shelving filters, crossovers and Hilbert transformers. The basis for the construction happened to be mostly EMQF filters, with 1st-kind Butterworth as their limiting case. The slope control in higher-order shelving filters was implemented using 2nd-kind Butterworth filters, although EMQF filters can also be used here with the drawback of having ripples in the pass and shelving bands.

Further reading

S.J.Orfanidis, *Lecture notes on elliptic filter design* (available on the author's webpage).

M.Kleehammer, *Mathematical development of the elliptic filter* (available in QSpace online repository).

Elliptic filter (Wikipedia artile).

L.M.Milne-Thomson, *Jacobian elliptic functions and theta functions* and *Elliptic Integrals* (contained in *Handbook of mathematical functions* by M.Abramowitz and I.A.Stegun, available on the internet).

[33] Of course at the last step we simply set $\omega_{\min}$ and $\omega_{\max}$ to the desired values, rather than perform the substitution (10.84).

Chapter 11

Multinotch filters

Multinotch filters have various uses. One of their most common applications is in phaser and flanger effects, which are built by modulating the parameters (in the simplest and the most common case just the cutoff) of the respective multinotch by an LFO. The main difference between a phaser and a flanger is that in the former the multinotch filter is based around a chain of differential allpass filters, while in the latter the allpass chain is replaced by a delay (thus making a comb filter).

11.1 Basic multinotch structure

Let $G(s)$ be an arbitrary allpass:

$$|G(j\omega)| = 1$$
$$\arg G(j\omega) = e^{j\varphi(\omega)}$$

where $\varphi(\omega)$ is the allpass's phase response, and consider the transfer function of the form

$$H(s) = \frac{1 + G(s)}{2} \tag{11.1}$$

corresponding to the system in Fig. 11.1.

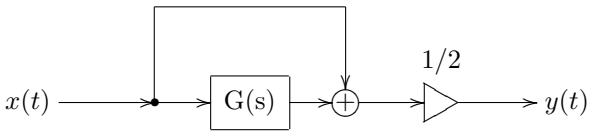


Figure 11.1: A basic multinotch. $G(s)$ is an allpass.

Writing out the amplitude response of $H(s)$ we have

$$|H(j\omega)|^2 = \left|\frac{1 + e^{j\varphi}}{2}\right|^2 = \left|\frac{1 + \cos\varphi + j\sin\varphi}{2}\right|^2 =$$
$$= \frac{(1 + \cos\varphi)^2 + \sin^2\varphi}{4} = \frac{2 + 2\cos\varphi}{4} = \frac{1 + \cos\varphi}{2} = \cos^2\frac{\varphi}{2}$$

and

$$|H(j\omega)| = \left|\cos\frac{\varphi(\omega)}{2}\right|$$

Thus

$$\begin{aligned} |H(j\omega)| = 1 \quad &\Longleftrightarrow \quad \varphi = 2\pi n \\ |H(j\omega)| = 0 \quad &\Longleftrightarrow \quad \varphi = \pi + 2\pi n \end{aligned} \qquad (n \in \mathbb{Z})$$

The points $|H(j\omega| = 1$ where the amplitude response of $H(s)$ is maximal are referred to as *peaks* and the points $|H(j\omega| = 0$ where the amplitude response of $H(s)$ is minimal are referred to as *notches*. So the peaks occur where the phase response of the allpass $G(s)$ is zero and the notches occur where the phase response of the allpass $G(s)$ is 180°. This is also fully intuitive: when the phase response of $G(s)$ is zero, both mixed signals add together, when the phase response of $G(s)$ is 180°, both mixed signals cancel each other.

The filters whose phase response contains several notches are referred to as *multinotch* filters. Apparently the filter in Fig. 11.1 is a multinotch.

11.2 1-pole-based multinotches

The allpass $G(s)$ can be arbitrary. However there are some commonly used options. One of such options is to use a chain of identically tuned 1-pole allpasses:

$$G(s) = G_1^N(s) \qquad G_1(s) = \frac{1-s}{1+s}$$

The phase response of a 1-pole allpass according to (2.13) is

$$\arg G_1(j\omega) = -2\arctan\omega$$

Respectively

$$\varphi(\omega) = \arg G(j\omega) = N\arg G_1(j\omega) = -2N\arctan\omega$$

(Fig. 11.2). The symmetry of the graph of $\varphi(\omega)$ in the logarithmic frequency scale is apparently due to the same symmetry of the phase response of the 1-pole allpass.

So the peaks occur whenever

$$\varphi = -2N\arctan\omega = -2\pi n \quad \Longleftrightarrow \quad \omega = \tan\frac{2\pi n}{2N} = \tan\frac{\pi n}{N}$$

and the notches occur whenever

$$\varphi = -2N\arctan\omega = -\pi - 2\pi n \quad \Longleftrightarrow \quad \omega = \tan\frac{\pi + 2\pi n}{2N} = \tan\frac{\frac{\pi}{2} + \pi n}{N}$$

Or, combining peaks and notches together, we have

$$\varphi = -2N\arctan\omega = -\pi n \quad \Longleftrightarrow \quad \omega = \tan\frac{\pi n}{2N}$$

Since we need $0 \le \omega \le +\infty$, the range of values of n is obtained from

$$0 \le \frac{\pi n}{2N} \le \frac{\pi}{2}$$

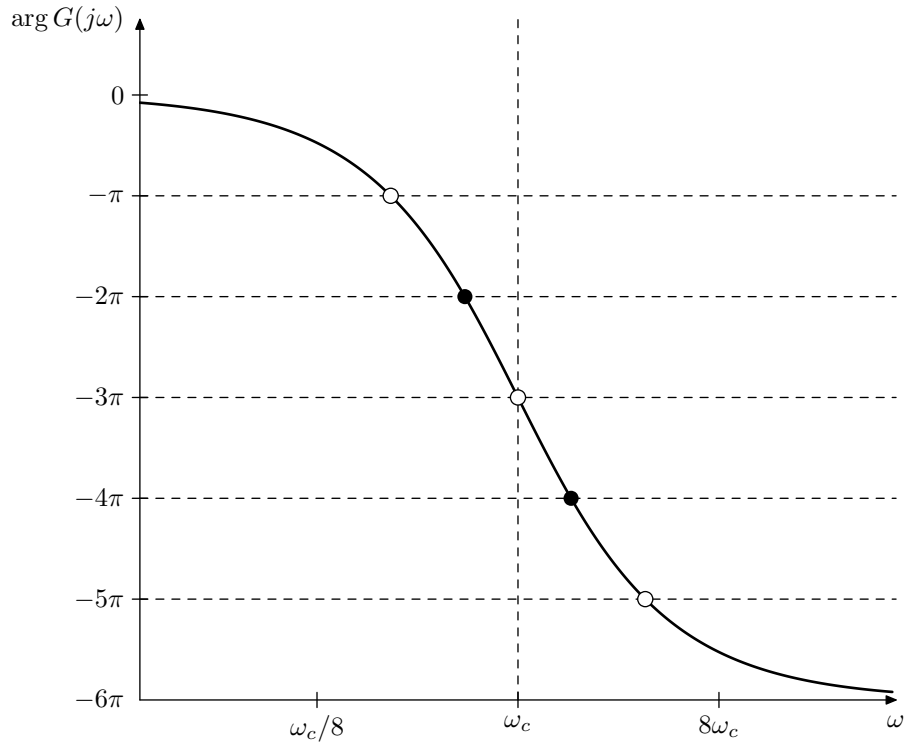


Figure 11.2: Phase response of a chain of 6 identical 1-pole allpasses. Black dots correspond to multinotch's peaks, white dots correspond to multinotch's notches.

giving

$$0 \le n \le N$$

Thus the total count of peaks plus notches is $N + 1$. Noticing that the peaks correspond to even values of n and notches correspond to odd values of n we have the following pictures:

If N is even there are $N/2 + 1$ peaks (including the ones at $\omega = 0$ and $\omega = +\infty$) and $N/2$ notches. Figs. 11.3 and 11.4 illustrate.

If N is odd there are $(N+1)/2$ peaks (starting at the one at $\omega = 0$) and $(N+1)/2$ notches (the last notch occuring at $\omega = +\infty$). Fig. 11.5 illustrates.

Odd counts are less commonly used due to unsymmetric shape of the amplitude response.

11.3 2-pole-based multinotches

Instead of 1-pole allpasses we could use 2-pole allpasses:

$$G(s) = G_2^N(s) \qquad G_2(s) = \frac{1 - 2Rs + s^2}{1 + 2Rs + s^2}$$

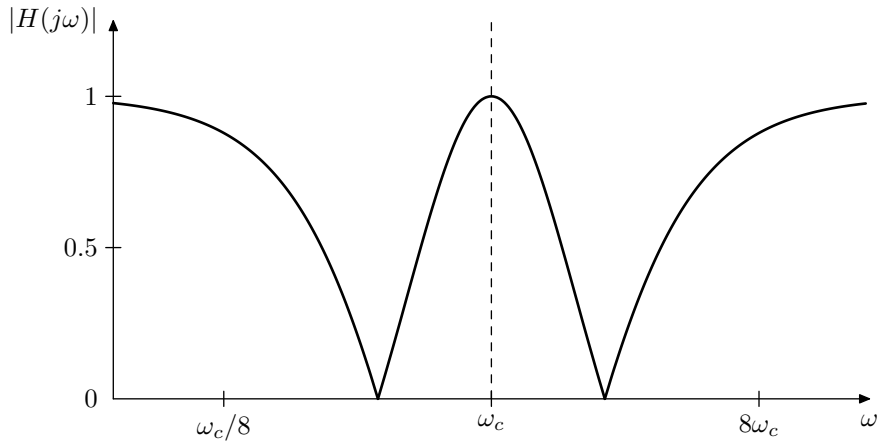


Figure 11.3: Amplitude response of a multinotch built around a chain of 4 identical 1-pole allpasses.

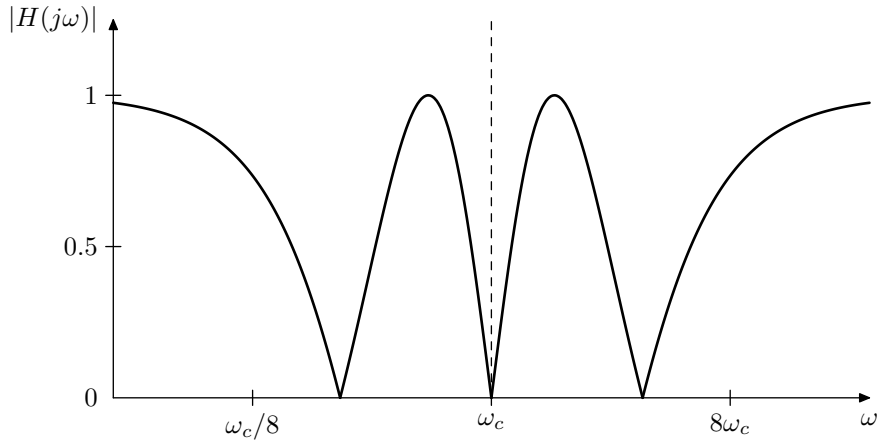


Figure 11.4: Amplitude response of a multinotch built around a chain of 6 identical 1-pole allpasses.

Note that at $R = 1$ we obtain an equivalent of a chain of $2N$ 1-pole allpasses.

According to (4.24) and (4.5) the phase response of a 2-pole allpass is

$$\arg G_2(j\omega) = -2 \operatorname{arccot} \frac{\omega^{-1} - \omega}{2R}$$

or, in terms of logarithmic frequency scale (where we also use (4.6))

$$\arg G_2(je^x) = -2 \operatorname{arccot} \frac{-\sinh x}{R}$$

Thus

$$\varphi(\omega) = \arg G(j\omega) = N \arg G_2(j\omega) = -2N \operatorname{arccot} \frac{\omega^{-1} - \omega}{2R}$$

$$\varphi(e^x) = \arg G(je^x) = N \arg G_2(je^x) = -2N \operatorname{arccot} \frac{-\sinh x}{R}$$

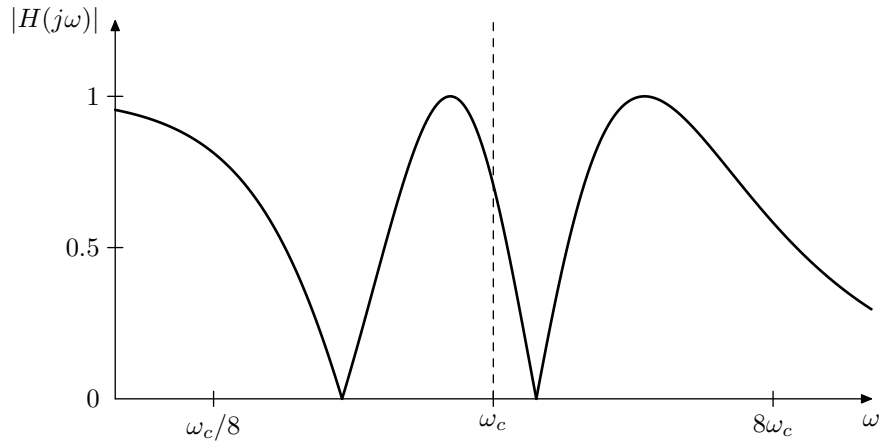


Figure 11.5: Amplitude response of a multinotch built around a chain of 5 identical 1-pole allpasses.

Thus this time $\varphi(\omega)$ is going from 0 to $-2\pi N$, which means that we obtain only symmetric amplitude responses, similar to the ones which we were getting for even numbers of 1-pole allpasses. Fig. 11.6 illustrates. By adjusting the value of R we change the steepness of the phase response and thereby the distance between the notches.

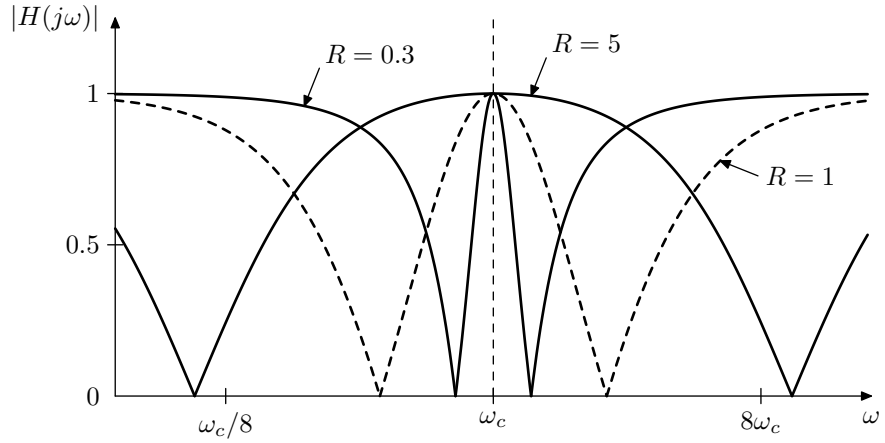


Figure 11.6: Amplitude response of a multinotch built around a chain of 2 identical 2-pole allpasses (at different damping values).

The first notch occurs at $\varphi = -\pi$, that is

$$-2N \operatorname{arccot} \frac{-\sinh x}{R} = -\pi$$

or

$$\sinh x = -R \cot \frac{\pi}{2N}$$

from where we can obtain the logarithmic position of the first notch

$$x = -\sinh^{-1} \left(R \cot \frac{\pi}{2N} \right) < 0$$

The logarithmic position of the last notch is respectively $-x$ and the logarithmic bandwidth (in base e) is therefore $-2x$, while the respective bandwidth in octaves is $-2x/\ln 2$:

$$\Delta = \frac{2}{\ln 2} \sinh^{-1}\left(R \cot \frac{\pi}{2N}\right)$$

Notice the obvious similarly of the above formula to (4.19).

11.4 Inversion

By multiplying an allpass filter's output by -1 we obtain another allpass. At frequencies where the phase response was $0°$ we thereby obtain $180°$ and vice versa. This means that if such allpass is used as a core of the multinotch in Fig. 11.1, inverting the allpass's output will swap the peak and notch positions (compare Fig. 11.7 vs. Fig. 11.4).

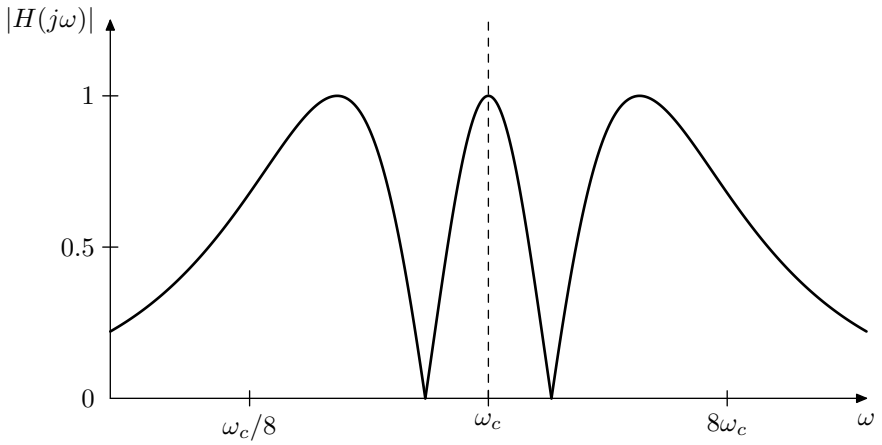


Figure 11.7: Amplitude response of a multinotch built around a chain of 6 identical 1-pole allpasses with inversion (compare to Fig. 11.4).

The structure of Fig. 11.1 can be modified as shown in Fig. 11.8 to accomodate optional inversion.

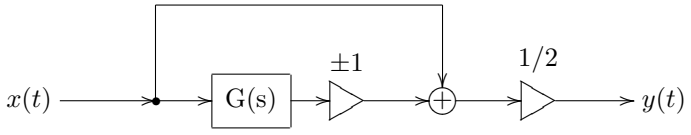


Figure 11.8: Multinotch from Fig. 11.1 with optional inversion.

11.5 Comb filters

A delay is also an allpass. It is not a differential allpass, since it's not based on integrators, but it is still an allpass. Indeed, taking the delay equation

$$y(t) = x(t - T)$$

where T is delay time and letting $x(t) = Ae^{st}$ we have

$$y(t) = Ae^{s(t-T)} = e^{-sT} \cdot Ae^{st} = e^{-sT} \cdot x(t)$$

Since the delay is linear (in the sense that a delayed linear combination of two signals is equal to the same linear combination of these signals delayed separately) we could apply (2.7) which means that the transfer function of the delay is

$$H(s) = e^{-sT}$$

Apparently

$$|H(j\omega)| = 1$$
$$\arg H(j\omega) = -\omega T$$

and thus the delay is an allpass.

Therefore we can use the delay as the allpass core of the multnotch filter in Fig. 11.1. Letting $G(s) = e^{-sT}$ we have $\varphi(\omega) = -\omega T$ (Fig. 11.9). The peak/notch equation is respectively

$$-\omega T = -\pi n$$

from where

$$\omega = \frac{\pi n}{T} = 2\pi \cdot \frac{n}{2T}$$

or, in ordinary frequency scale

$$f = \frac{n}{2T}$$

The peaks and notches are therefore harmonically spaced with a step of $1/2T$ Hertz (Fig. 11.10). The amplitude response in Fig. 11.10 looks like a comb. Hence this kind of multinotch filters are referred to as *comb filters*.

Since the peaks and notches of the comb filter's amplitude response occur at $f = n/2T$, the frequency $1/2T$ is the fundamental frequency of this harmonic series. It is convenient to use this frequency as comb's filter formal cutoff $f_c = 1/2T$.

If there is no inversion, then (excluding the DC peak at $f = 0$) the peaks of the amplitude response are lcoated at frequencies $2f_c$, $4f_c$, $6f_c$, etc. This makes the perceived fundamental frequency of the comb filter (especially in the case of a strong resonance[1]) rather be $2f_c$. However in the case of inversion the peaks are located at f_c, $3f_c$, $5f_c$, etc., giving an impression (which is stronger in the case of a strong resonance) of an odd-harmonics-only signal at frequency f_c.

[1] Resonating multinotches will be discussed later in this chapter.

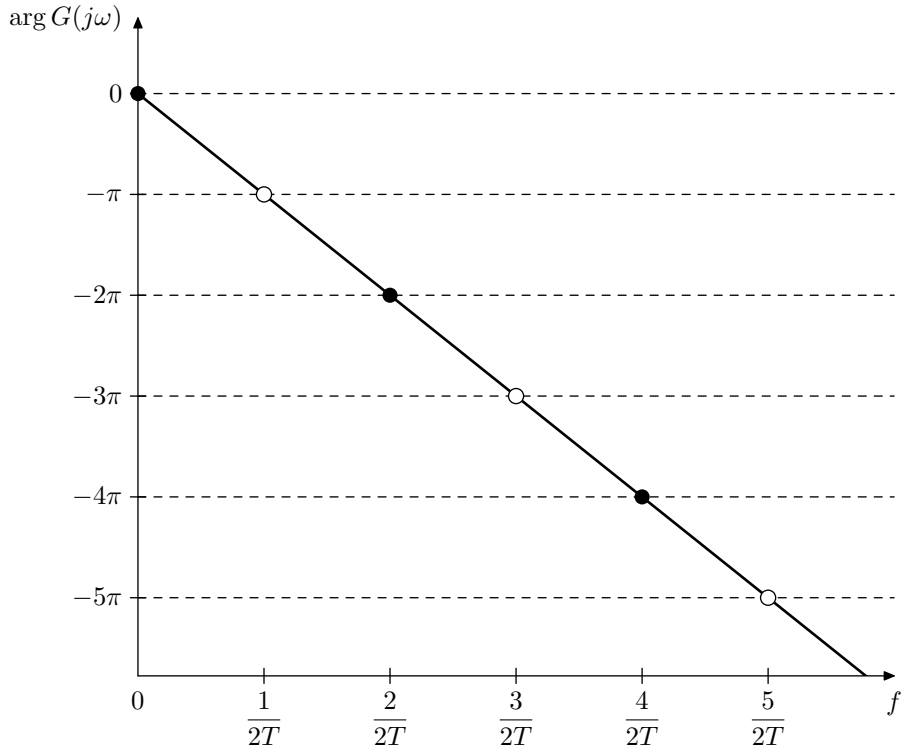


Figure 11.9: Phase response of a delay. Black dots correspond to multinotch's peaks, white dots correspond to multinotch's notches. The frequency scale is linear!

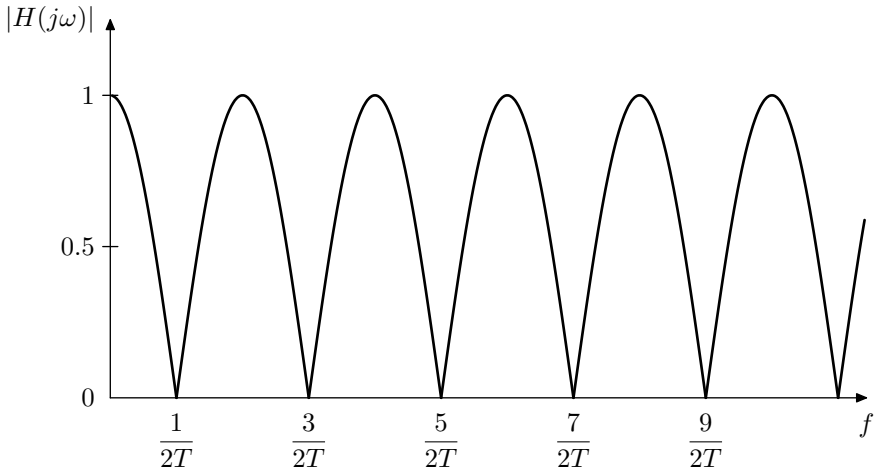


Figure 11.10: Amplitude response of a multinotch built around a delay (comb filter). The frequency scale is linear!

11.6 Feedback

Suppose we introduce feedback into the structure of Fig. 11.1 as shown Fig. 11.11. Now the output of the allpass $G(s)$ is not anymore purely the allpassed input signal. Let's introduce the notation $\tilde{y}(t)$ for the post-allpass signal (as shown in Fig. 11.11) and $\tilde{G}(s)$ for the respective transfer function (in the sense of $\tilde{y} = G(s)x$ for complex exponential x). We also introduce the pre-allpass signal $\tilde{x}(t)$, but we are not going to use it for now. Then we are having

$$\tilde{G}(s) = \frac{G(s)}{1 - kG(s)}$$

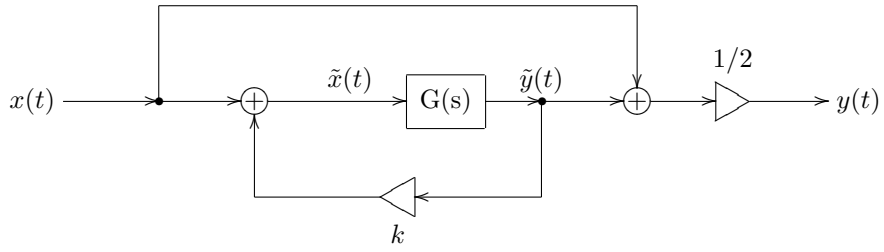


Figure 11.11: Multinotch from Fig. 11.1 with added feedback. Note that this figure is showing a poor mixing option.

The transfer function of the entire multinotch thereby turns into

$$H(s) = \frac{1 + \tilde{G}(s)}{2} = \frac{1}{2} \cdot \frac{1 - kG(s) + G(s)}{1 - kG(s)} = \frac{1}{2} \cdot \frac{1 + (1 - k)G(s)}{G(s)}$$

or, in frequency response terms

$$H(j\omega) = \frac{1}{2} \cdot \frac{1 + (1 - k)e^{j\varphi}}{1 - ke^{j\varphi}}$$

We can immediately notice that as soon as $k > 0$ the numerator of the frequency response doesn't turn to zero anymore, respectively we are not having fully deep notches in the amplitude response (Fig. 11.12).

Instead of mixing $\tilde{y}(t)$ with $x(t)$ let's mix it with $\tilde{x}(t)$, as shown in Fig. 11.13. The transfer function corresponding to the signal $\tilde{x}(t)$ in Fig. 11.11 is

$$\frac{\tilde{G}(s)}{G(s)} = \frac{1}{1 - kG(s)}$$

and thus we obtain

$$H(s) = \frac{1}{2} \cdot \left(\frac{1}{1 - kG(s)} + \frac{G(s)}{1 - kG(s)} \right) = \frac{1}{2} \cdot \frac{1 + G(s)}{1 - kG(s)}$$

This transfer funtion looks much better, since it preserves fully deep notches. The frequency response turns into

$$H(j\omega) = \frac{1}{2} \cdot \frac{1 + e^{j\varphi}}{1 - ke^{j\varphi}}$$

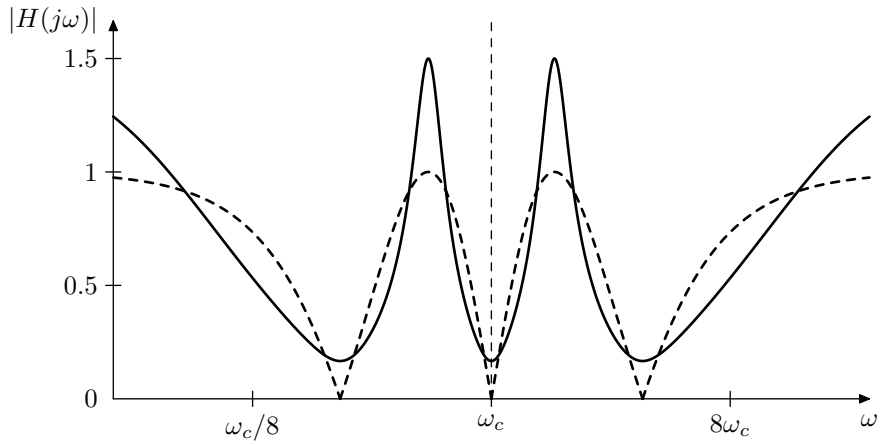


Figure 11.12: Amplitude response of the multinotch in Fig. 11.11 built around a chain of 6 identical 1-pole allpasses at $k = 0.5$. Dashed curve corresponds to $k = 0$ (the same response as in Fig. 11.4).

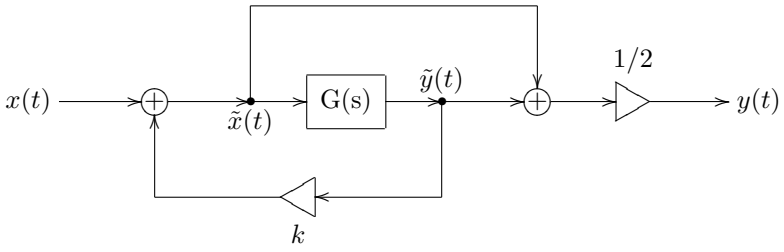


Figure 11.13: Multinotch from Fig. 11.1 with added feedback and corrected mixing.

which varies between

$$H(j\omega) = \frac{1}{2} \cdot \frac{1+1}{1-k} = \frac{1}{1-k} \qquad \text{when } \varphi = 2\pi n \tag{11.2a}$$

and

$$H(j\omega) = \frac{1}{2} \cdot \frac{1-1}{1-k} = 0 \qquad \text{when } \varphi = \pi + 2\pi n \tag{11.2b}$$

The amplitude response is then

$$\begin{aligned} |H(j\omega)|^2 &= \frac{1}{4} \cdot \left| \frac{1+e^{j\varphi}}{1-ke^{j\varphi}} \right|^2 \frac{1}{4} \cdot \left| \frac{1+\cos\varphi+j\sin\varphi}{1-k\cos\varphi-jk\sin\varphi} \right|^2 = \\ &= \frac{1}{4} \cdot \frac{(1+\cos\varphi)^2+\sin^2\varphi}{(1-k\cos\varphi)^2+k^2\sin^2\varphi} = \frac{1}{4} \cdot \frac{2+2\cos\varphi}{1+k^2-2k\cos\varphi} = \\ &= \frac{1}{2} \cdot \frac{1+\cos\varphi}{1+k^2+2k-2k(1+\cos\varphi)} = \frac{\cos^2\dfrac{\varphi}{2}}{(1+k)^2-4k\cos^2\dfrac{\varphi}{2}} \end{aligned}$$

Again one can see that $H(j\omega)| = 1/(1-k)$ when $\cos^2(\varphi/2) = 1$ and $H(j\omega) = 0$ when $\cos^2(\varphi/2) = 0$. Thus the effect of the feedback in Fig. 11.13 is that the peaks become $1/(1-k)$ times higher (given $0 < k < 1$) and notches stay intact. Fig. 11.14 illustrates. Observe that the peaks become higher and narrower.[2]

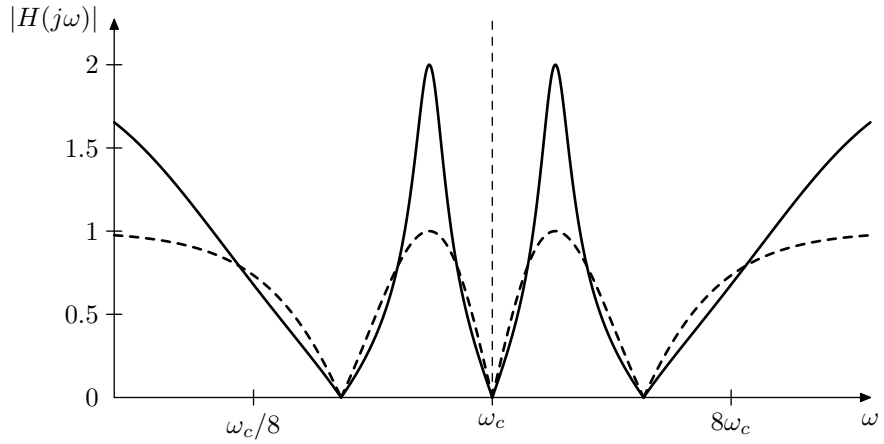


Figure 11.14: Amplitude response of the multinotch in Fig. 11.13 built around a chain of 6 identical 1-pole allpasses at $k = 0.5$. Dashed curve corresponds to $k = 0$ (the same response as in Fig. 11.4).

We could combine the feedback (Fig. 11.13) and the inversion (Fig. 11.8), as shown in Fig. 11.15. Apparently the inversion only adds another 180° to $\varphi(\omega)$, swapping peaks and notches. Therefore the results of the previous discussion of Fig. 11.13 equally apply to Fig. 11.15.

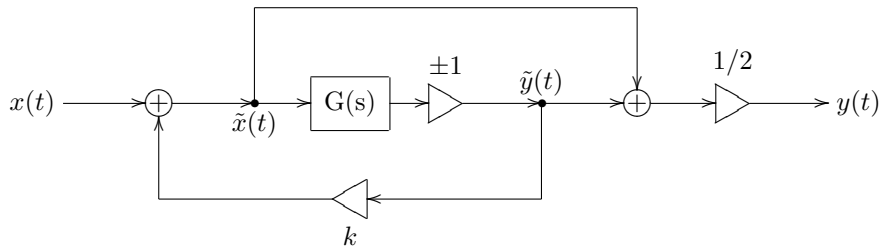


Figure 11.15: Multinotch with feedback and inversion.

As we should recall from the discussion of ladder filters, the feedback becomes unstable when the total gain across the feedback loop, computed at a

[2]Since $\tilde{x} = x + k\tilde{y}$, instead of simple averaging $y = (\tilde{x} + \tilde{y})/2$ we could have had

$$y = \frac{x + k\tilde{y} + \tilde{y}}{2} = \frac{1}{2}x + \frac{1+k}{2}\tilde{y}$$

however this doesn't seem to give any benefits compared to the previous option, while we need to adjust the mixing coefficient for $\tilde{y}$ depending on the feedback amount, which is rather a drawback.

frequency where the total phase shift across the feedback loop is zero, exceeds 1. Apparently in the case of Fig. 11.15 the zero total phase shift is occurring exactly at the frequencies where the multinotch has peaks, while the total feedback loop gain at these frequencies is simply k. Therefore the multinotch filter becomes unstable at $k = 1$ and the suggested range of k is $0 \leq k < 1$.[3] Note that the presence of the inversion doesn't really change the stable range of k, since the allpass $G(s)$ is anyway delivering all possible phase shifts across the frequency range $0 \leq \omega < +\infty$, and there always will be frequencies at which the total feedback loop phase shift is zero (thereby producing amplitude response peaks), regardless of whether the inversion is on or off. Thus the feedback loop will be stable as long as $|k| < 1$.

Feedback shaping

Being essentially a ladder allpass, the multnotch in Fig. 11.15 can accomodate feedback shaping, as discussed in Section 5.4. Notably, as long as the amplitude responses of the shaping filters do not exceed 1, neither will the total feedback loop gain (since in the absence of shaping filters the feedback loop gain is exactly 1 at all frequencies). This means that the stability of the feedback loop for $|k| < 1$ will not be destroyed, no matter what the phase responses of the shaping filters are.

11.7 Dry/wet mixing

So far we have been mixing the allpass-processed signal and the input signal (or, if we are using feedback, the pre-allpass signal $\tilde{x}(t)$ with the post-allpass signal $\tilde{y}(t)$) in equal amounts:

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

Let's crossfade the multinotch filter output signal with the input signal:

$$y = a\frac{\tilde{x} + \tilde{y}}{2} + (1 - a)x \tag{11.3}$$

If the multinotch is being used as a part of a phaser or flanger effect, the input signal is commonly referred to as the *dry signal* while the multinotch output signal $(\tilde{x} + \tilde{y})/2$ is referred to as the *wet signal*.[4]

According to (11.2a) the phase response of the feedback multinotch at the peak is zero, therefore the peak, having the height $1/(1-k)$ should mix naturally with the input signal (correspoding to the transfer function equal to 1 everywhere), producing a smooth crossfade between $1/(1-k)$ and 1 in the amplitude

[3] Negative values of k lower the amplitude response peaks below 1, simultaneously making them wider and respectively making the notches narrower. Being narrower, such notches become less audible, even if we compensate for the amplitude loss by multiplying the signal by $1 - k$, thus the case of $k < 0$ is less common.

[4] Sometimes just the allpass output signal $\tilde{y}$ is referred to as the wet signal. Such terminology is however more appropriate for an effect such as e.g. chorus, where the main idea of the effect is pitch detuning produced by delay modulation. In comparison e.g. in a flanger the main idea of the effect is the appearance of the notches, while pitch detuning, if present at all, is rather a modulation artifact. Thus, in absence of strong modulation, the output of the flanger's delay will be hardly distinguishable by ear from the dry signal, not really being "wet".

response at this frequency. This is indeed the case and the amplitude response of a multinotch will nicely crossfade into a unity gain response (Fig. 11.16). The respective structure is shown in Fig. 11.17.

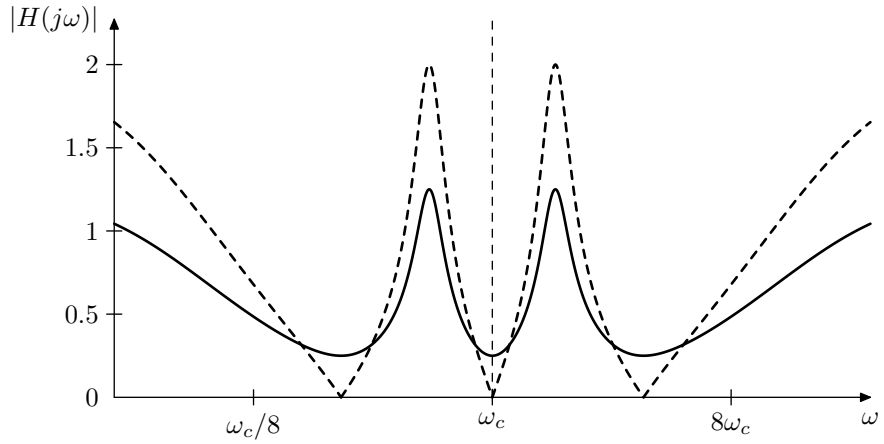


Figure 11.16: Amplitude response of the multinotch in Fig. 11.13 built around a chain of 6 identical 1-pole allpasses at $k = 0.5$ with a dry/wet mixing ratio of 50%. Dashed curve corresponds to a dry/wet mixing ratio of 100% (same response as in Fig. 11.14).

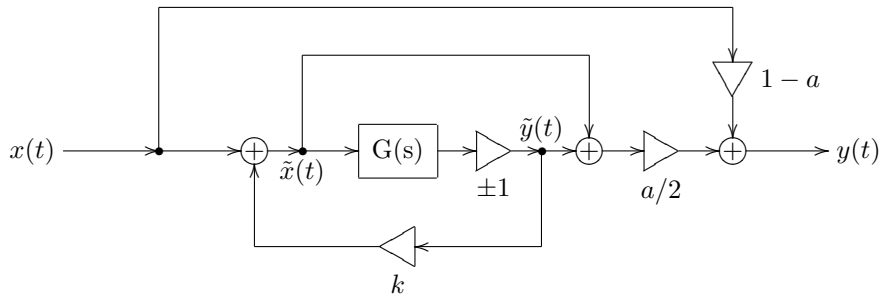


Figure 11.17: Multinotch with feedback, inversion and dry/wet mixing.

Since $\tilde{x} = x + k\tilde{y}$, we can rewrite (11.3) as

$$y = a\frac{x + k\tilde{y} + \tilde{y}}{2} + (1-a)x = \frac{a}{2}(x + (1+k)\tilde{y}) + (1-a)x = \\ = \left(1 - \frac{a}{2}\right)x + \frac{a}{2}(1+k)\tilde{y}$$

Thus, even though normally $0 \le a \le 1$, we could let a grow all the way to $a = 2$, in which case only the allpass output $\tilde{y}$ (albeit boosted by $1 + k$) will be present in the output signal.

11.8 Barberpole notches

Consider the frequency shifter in Fig. 10.31 and let's replace $\Delta\omega \cdot t$ with some fixed value $\Delta\varphi$, obtaining a similar structure shown in Fig. 11.18.[5]

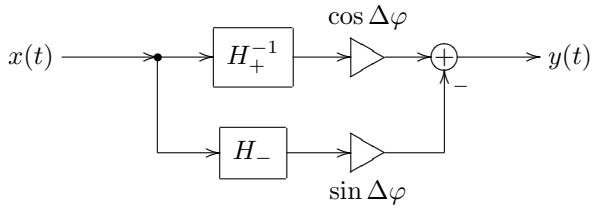


Figure 11.18: Barberpole allpass, obtained from the frequency shifter in Fig. 10.31.

Across the supported bandwidth of the frequency shifter the phase difference between the allpasses H_+^{-1} and H_- is 90°. That is

$$\varphi_+(\omega) - \varphi_-(\omega) = 90^\circ$$

where

$$\varphi_+(\omega) = \arg H_+^{-1}(j\omega)$$
$$\varphi_-(\omega) = \arg H_-(j\omega)$$

or simply

$$H_-(s) = -jH_+^{-1}(s)$$

The frequency response of the structure in Fig. 11.18 (within the supported bandwidth of the frequency shifter) is thereby

$$\begin{aligned} G(j\omega) &= H_+^{-1}(j\omega) \cdot \cos\Delta\varphi - H_-(j\omega) \cdot \sin\Delta\varphi = \\ &= H_+^{-1}(j\omega) \cdot \cos\Delta\varphi + jH_+^{-1}(j\omega) \cdot \sin\Delta\varphi = \\ &= H_+^{-1}(j\omega) \cdot (\cos\Delta\varphi + j\sin\Delta\varphi) = e^{j\Delta\varphi} \cdot H_+^{-1}(j\omega) \end{aligned}$$

from where we repsectively obtain

$$|G(j\omega)| = \left|e^{j\Delta\varphi}\right| \cdot \left|H_+^{-1}(j\omega)\right| = 1 \tag{11.4a}$$

$$\arg G(j\omega) = \arg e^{j\Delta\varphi} + \arg H_+^{-1}(j\omega) = \arg H_+^{-1}(j\omega) + \Delta\varphi \tag{11.4b}$$

That is, $G(s)$ is an allpass and by varying $\Delta\varphi$ we can arbitrarily offset its phase response! Of course, this holds only within the frequency shifter's bandwidth, but nevertheless it's a very remarkable property.

But what does the phase response of $G(s)$ actually look like? Apparently it depends on the details of H_+^{-1} and H_- implementations. If H_+^{-1} and H_- are built from 1-pole allpasses obtained by minimax optimization of the phase difference (e.g. by using formula eq:ellip:PhaseSplit:PolesZeros), the phase responses of H_+^{-1} and H_- will look like the ones in Fig. 11.19, where we first should concentrate on the phase responses shown by solid lines.

[5]The author was introduced to the approach of using the frequency shifter structure to implement barberpole phasers and flangers by Dr. Julian Parker.

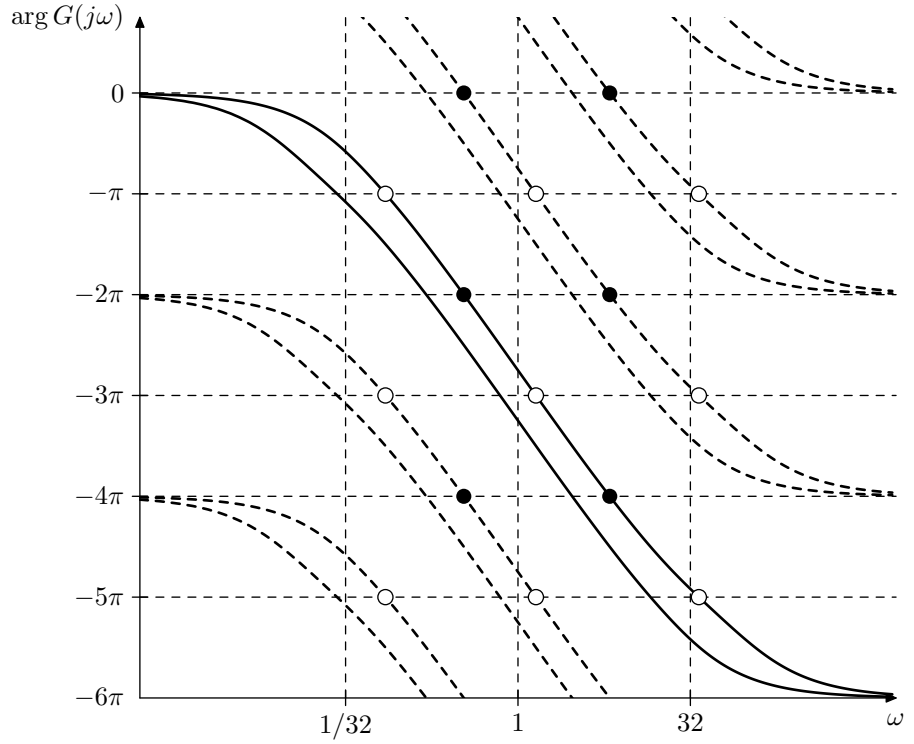


Figure 11.19: Phase responses of H_+^{-1} and H_- (each consisting of six 1-pole allpasses). The frequency shifter bandwidth is 10 octaves (bounded by vertical dashed lines at $\omega = 1/32$ and $\omega = 32$). Black dots correspond to multinotch's peaks arising out of H_+^{-1}, white dots correspond to the respective notches. Dashed curves show "aliased" phase responses.

Aside from being 90° apart across the frequency shifter bandwidth, the phase responses in Fig. 11.19 do not look much different from the phase responses we have been using earlier, such as e.g. in Fig. 11.2. Thus H_+^{-1} or H_- will provide a decent allpass to be used in a multinotch. By using (11.4b) we can obtain an offset phase response of H_+^{-1} as the phase response of $G(s)$, which will result in shifted peaks and notches of the multinotch (compared to their positions arising out of $H_+ - 1$).

However, recall that the phase is defined modulo 360°. That is a phase response of −20° is exactly the same as the phase response of −380° or of −740° etc. This has been shown by the dashed curves in Fig. 11.19, they represent alternative interpretations or "aliased" versions of the "principal" (solid-line) phase responses. Notice how the black and white dots on the aliased responses of H_+^{-1} correspond to exactly the same peak and notch frequencies as the ones arising out of the principal phase responses (reflecting the fact that it doesn't matter if we use a principal or an aliased phase response to determine peak and notch positions). By offsetting the phase response of H_+^{-1} (visually this corresponds to a vertical shifting of the responses in Fig. 11.19) we simultaneously offset all its aliases by the same amount.

Imagine that $\Delta\varphi$ is increasing, thus the principal and aliased responses of H_+^{-1} in Fig. 11.19 are continuously moving upwards, and the notches and peaks are continuously moving to the right.[6] In turn, each of the peaks and notches will disappear on the right at $\omega = +\infty$ simultaneously reappearing from the left at $\omega = 0$. Thus the peaks and notches will move "endlessly" from left to the right. Respectively if $\Delta\varphi$ is decreasing, they will move from right to the left. This is the so-called *barberpole effect*.

In reality, however, the peaks and notches will not move all the way to $\omega = +\infty$ or $\omega = 0$. At some point they will leave the frequency shifter bandwidth, at which moment (11.4b) will no longer hold. Particularly, the amplitude response of $G(s)$ will no longer stay allpass. At $\omega = 0$ we have $H_+^{-1}(0) = H_-(0) = 1$, which means that

$$G(0) = 1 \cdot \cos\Delta\varphi + 1 \cdot \sin\Delta\varphi = \cos\Delta\varphi + \sin\Delta\varphi = \sqrt{2} \cdot \cos\left(\Delta\varphi - \frac{\pi}{4}\right)$$

which means that the amplitude response of $G(s)$ at $\omega = 0$ can get as large as $\sqrt{2}$. The same situation occurs at $\omega = +\infty$. Respectively, if the multinotch contains feedback, it will explode at $k = 1/\sqrt{2}$. The explosion can be prevented by introducing low- and high-pass or -shelving filters into the feedback loop.[7]

Thus, we have built a *barberpole phaser*, where the peaks and notches can move endlessly to the left or to the right. The same technique cannot be directly used to build a barberpole flanger, since, while we have a phase splitter acting as a differential allpass, we do not have a phase splitter acting as a delay. This would not be even possible in theory, since the phase response of a delay must be proportional to the frequency (this is the property which ensures the harmonic spacing of comb filter's peaks and notches), but adding any constant to such phase response will destroy this property. What is however possible is using an allpass arising out of a serial connection of a delay and a barberpole allpass in Fig. 11.18. This would destroy the perfect harmonic spacing of flanger's peaks and notches, but one gets a barberpole effect in return, as the phase responses of the delay and the barberpole allpass add up.

SUMMARY

Multinotch filters can be build by mixing a signal with its allpassed version, where the allpass could be a differential allpass or a delay, the latter resulting in a comb filter. Inverting the allpass's output swaps the peaks and the notches. Adding feedback makes the peaks more prominent.

---

[6] In a practical implementation $\Delta\varphi$ would not be able to increase endlessly, as at some point it will leave the representable range of values. If floating point representation is used, precision losses will become intolerably large even before the value gets out of range. However, we don't really need to increase or decrease $\Delta\varphi$ endlessly, since what matters in the end (according to Fig. 11.18) are the values of its sine and cosine. Thus we could wrap $\Delta\varphi$ to the range $[-\pi, \pi]$, or work directly with sine and cosine values (in which case it's convenient to treat them as real and imaginary parts of a complex number $e^{j\Delta\varphi}$).

[7] Particularly, for the 1-pole lowpass (or any 1st kind Butterworth lowpass) we have $|H(j\omega)| \leq 1/\sqrt{2}\ \forall\omega \geq \omega_c$, while outside of that range we still have $|H(j\omega)| \leq 1$. Therefore such filter, placed at the upper boundary of the frequency shifter's bandwidth, will be guaranteed to mitigate the unwanted amplitude response boost in the high frequency range. A highpass of the same kind placed at the lower boundary of the frequency shifter's bandwidth will perform the same in the low frequency range.

History

The revision numbering is major.minor.bugfix. Pure bugfix updates are not listed here.

1.0.2 (May 18, 2012)

first public revision

1.1.0 (June 7, 2015)

- TSK filters
- frequency shifters
- further minor changes

2.0.0alpha (May 28, 2018)

- redone: TSK/SKF filters
- 8-pole ladder filters
- expanded: nonlinearities
- expanded: phasers and flangers (now found under the title *multinotch filters*)
- Butterworth transformations of the 1st and 2nd kinds
- classical signal processing filters (Butterworth, Chebyshev, elliptic)
- redone: shelving filters
- redone: Hilbert transformers
- crossovers
- state-space form
- transient responses
- many further smaller changes

Index