

8 Filter: lineare, zeit-invariante Systeme

Markus Lippitz

23. Mai 2022

Ziel Sie können einen Filter *designen* und *anwenden*

- Frequenzantwort, Impulsantwort
- analoge und digitale Filter

Weitere Aufgaben

- Filtern Sie ein Audio-Signal! (see PortAudio.jl und SampledSignals.jl)

Literatur Butz Kap. 5, Horowitz / Hill, Kap. 1.19

Überblick

Wir betrachten allgemeine Eigenschaften eines linearen, Zeit-invarianten Systems. Das System reagiert also *linear* auf das Eingangssignal ('doppelt so viel hilft doppelt so viel') und ist Zeit-invariant. Die Antwort heute unterscheidet sich also nicht von der gestern. Diese Anforderungen erfüllen viele Systeme, beispielsweise elektronische Filter, getriebenen Oszillatoren, dielektrische Materialien oder angeregte Moleküle.

Zeitliche Antwort

Auf einen Eingang $f(t)$ reagiert das System mit einer Antwort $y(t)$, für die gilt

$$y(t) = \int_{-\infty}^{+\infty} f(t') h(t - t') dt' \quad .$$

Dabei ist y also die Faltung von f mit h . Bei kausale Systemen darf die Antwort jetzt nicht von einem Eingang in der Zukunft abhängen. t' muss also immer kleiner als t sein. Dies erreicht man, in dem man fordert

$$h(t - t' < 0) = 0 \quad .$$

Dabei wird h die *Impulsantwort* genannt. Wenn nämlich der Eingang eine Delta-Funktion ist $f(t) = \delta(t)$ dann wird $y(t) = h(t)$. So lässt sich auch h messen, indem man einen möglichst delta-förmigen Impuls als Eingang verwendet und die Reaktion darauf aufzeichnet.

Frequenz-Antwort

Da $y(t)$ die Faltung von $f(t)$ mit $h(t)$ ist, gilt für die Fourier-transformierte Frequenzantwort

$$Y(\omega) = F(\omega) H(\omega)$$

wobei kleine Buchstaben durch FT in große Buchstaben übergehen. $H(\omega)$ nennt sich *Transferfunktion*. Man kann diese Funktion messen, in dem man beispielsweise die Antwort auf Eingänge mit verschiedenen Frequenzen misst. Alternativ kann man ein im Frequenzraum flaches, breites (= 'weißes') Spektrum anlegen und das Frequenz-Spektrum des Ausgangs messen.

Kramers-Kronig-Beziehung

Auch wenn $h(t)$ reelwertig ist, so wird $H(\omega)$ doch komplexwertig sein, da $h(t)$ nicht gerade (Kosinus-förmig) ist. Aus der Kausalität, also $h(t - t' < 0) = 0$ folgt allerdings ein Zusammenhang zwischen Real- und Imaginär-Teil von $H(\omega)$, die Kramers-Kronig-Beziehung. Da dielektrische Materialien linear und Zeit-invariant sind, gibt es so einen Zusammenhang zwischen Real- und Imaginär-Teil der dielektrischen Funktion bzw. des Brechungsindex

$$\epsilon'(\omega) = 1 + \frac{2}{\pi} \mathcal{P} \int_0^\infty \frac{\Omega \epsilon''(\Omega)}{\Omega^2 - \omega^2}$$

und analog auch in die andere Richtung. \mathcal{P} bezeichnet dabei das Cauchy'sche Hauptwert-Integral, bei dem der Pol bei $\Omega = \omega$ ausgenommen wird.

Analog kann man aus bekannten Amplitudenverlauf eines getriebenen Oszillator seinen Phasenverlauf bestimmen und umgekehrt.

Kaskadierte Systeme

Wenn mehrere lineare, Zeit-invariante Systeme hintereinander geschaltet sind, also der Ausgang des einen als Eingang des anderen dient, dann ist die Transferfunktion des Gesamtsystems das Produkt aller Transferfunktionen (und daher von der Reihenfolge der Systeme unabhängig). Die Gesamt-Impulsantwort ist also die Faltung aller Einzel-Impulsantworten.

Entfaltung

Limitierte Transferfunktionen bzw. nicht delta-förmige Impulsantworten sind unpraktisch, aber allgegenwärtig. Gerne würde man sie nachträglich rechnerisch aus der Messung beseitigen, um diese zu korrigieren. Das ist aber nur sehr begrenzt möglich. Man könnte die im Zeitraum gemessenen Größen Fourier-Transformieren und dann rechnen

$$F_{exp}(\omega) = \frac{Y_{exp}(\omega)}{H(\omega)} \quad .$$

Das Problem ist, dass bei limitiertem $H(\omega)$ dieses quasi Null ist für manche Frequenzen. An diesen Frequenzen ist dann auch das gemessene $Y(\omega)$ quasi Null und der wahre von Null verschiedene Wert von $F(\omega)$ lässt sich nicht rekonstruieren. Wenn die Information einmal weg ist, kann man sie nicht wieder beschaffen. In Bereichen, in denen H zwar klein, aber nicht Null ist, kann dieses Verfahren helfen, solange der gemessene Wert von Y gut genug, also rauscharm genug bekannt ist.

Was man an dieser Stelle typischerweise macht, ist nicht die Daten um die Messprozedur zu entfalten, sondern das Modell mit der Apparatur zu falten, und dann gemessene Y mit modellierten Y zu vergleichen.

Reihenentwicklung der Transferfunktion

An Stelle von

$$y(t) = \int_{-\infty}^{+\infty} f(t') h(t - t') dt' = \int_{-\infty}^{+\infty} f(t - t') h(t') dt'$$

hätten wir auch schreiben können (Butz, eq. 5.1)

$$y(t) = \sum_{j=-k}^k a_j f^{[j]}(t) \quad \text{mit} \quad f^{[j]}(t) = \frac{d^j f(t)}{dt^j}$$

wobei negatives j entsprechende Integration bedeutet. Ausgeschrieben ist das

$$y(t) = a_0 f(t) + a_1 \frac{df(t)}{dt} + a_2 \frac{d^2 f(t)}{dt^2} + \dots + a_{-1} \int f(\tau) d\tau + a_{-2} \iint f(\tau) d\tau d\tau' + \dots$$

Durch Fourier-Transformation wird daraus

$$Y(\omega) = \sum_{j=-k}^k a_j (i\omega)^j F(\omega)$$

oder

$$H(\omega) = \sum_{j=-k}^k a_j (i\omega)^j$$

Wir stellen also H als Polynom in ω dar und die a_j sind die Koeffizienten. Diese Koeffizienten bestimmen, aus welchen Ableitungen und Integralen von f der Ausgang y gebildet wird.

Diskrete Datenreihe

Jetzt gehen wir über zu diskreten Samples f_k im zeitlichen Abstand Δt . Wir können sowohl die Integral-Definition von y diskretisieren, also ausgehen von

$$y(t) = \int_{-\infty}^{+\infty} f(t') h(t - t') dt' = \int_{-\infty}^{+\infty} f(t - t') h(t') dt'$$

oder wir schreiben die Integrale und Differentiationen von

$$y(t) = \sum_{j=-k}^k a_j f^{[j]}(t)$$

als Summen und Differenzen von aufeinanderfolgenden Elementen f_k . Egal wie, wir bekommen

$$y_k = \sum_{l=-L}^L a_l f_{k+l} = \sum_{l=-L}^L a_l V^l f_k$$

mit dem Verschiebe-Operator V^l

$$V^n f_k = f_{k+n}$$

Die Größe L muss rein praktisch endlich sein, weil wir in diesem Universum nicht unendlich viele Werte f_k speichern können. Wenn h_n mit positiven n von Null verschieden sind, dann ist die Impulsantwort h *akausal*. Wir müssen in diesem Fall Werte von f in der Zukunft kennen, um y jetzt auszurechnen. Das geht technisch bedingt, in dem man beispielsweise L Werte speichert und y entsprechend verspätet ausgibt. Die Fourier-Transformation ist beispielsweise *akausal*.

Natürlich können wir das auch Fourier-Transformieren

$$Y_j = H_j F_j$$

mit

$$H_j = \sum_{l=-L}^L a_l e^{i\omega_j l \Delta t} \quad \text{und} \quad \omega_j = 2\pi j / (N \Delta t)$$

Die Anzahl N der Elemente im Datensatz ist typischerweise viel größer als die Länge L der Filterkoeffizienten. Daher wird sehr oft die diskrete Natur der ω_j ignoriert und ein kontinuierliches ω geschrieben, also

$$H(\omega) = \sum_{l=-L}^L a_l e^{i\omega l \Delta t}$$

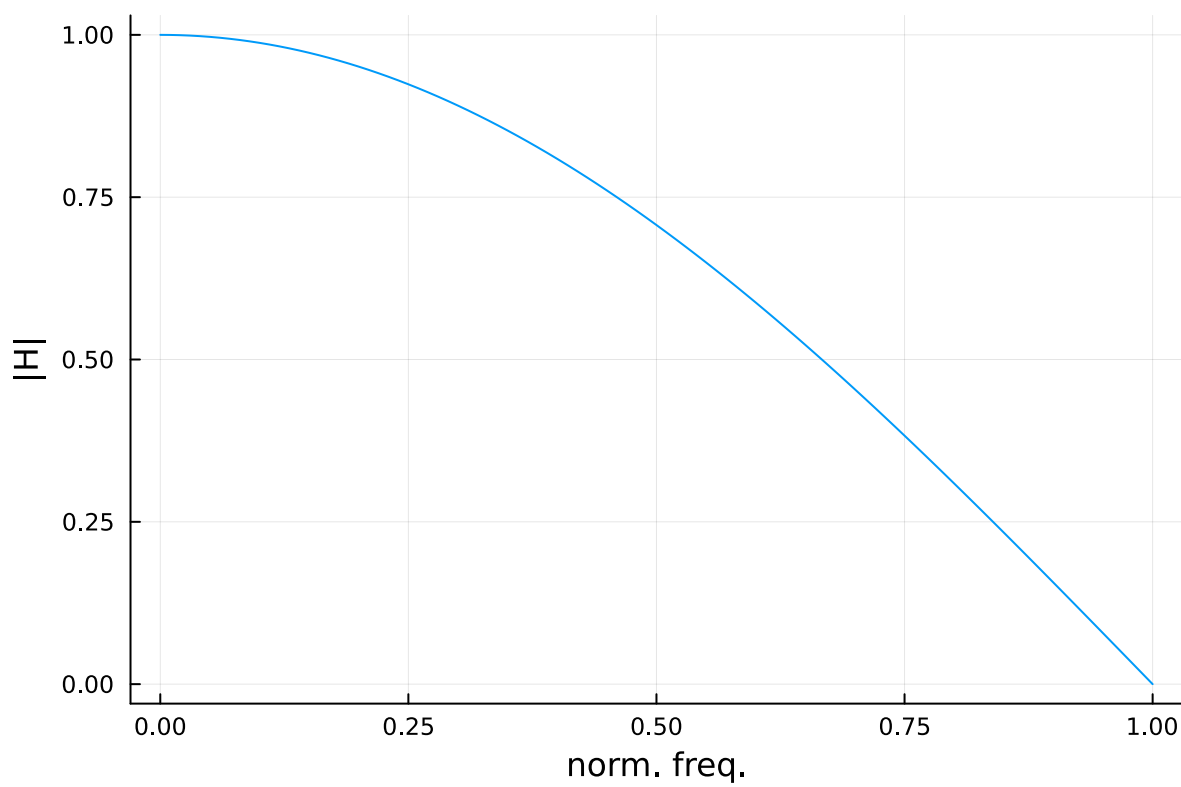
Beispiel: Tiefpass-Filter

Wir folgen weiterhin Butz und betrachten einen Tiefpass-Filter, der einfach durch Mittelwertbildung entsteht. Sei

$$y_k = \frac{1}{2}(f_k + f_{k+1})$$

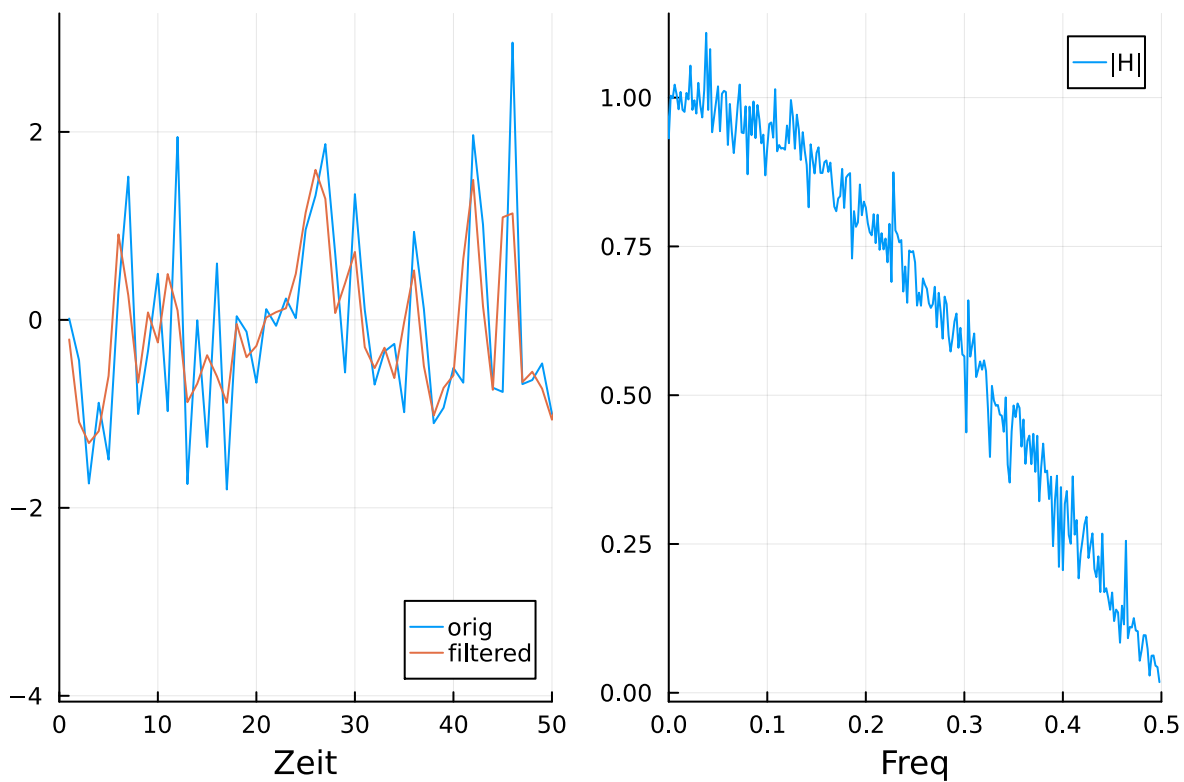
also $a_0 = a_1 = 1/2$. Damit bekommen wir

$$H(\omega) = \frac{1}{2} (1 + e^{i\omega \Delta t}) \quad \text{bzw.} \quad |H(\omega)| = \cos \frac{\omega \Delta t}{2} = \cos \frac{\pi}{2} \frac{\omega}{\Omega_{Nyq.}}$$



```
1 let
2   x=(0:0.01:1)
3   aside(embed_display(plot(x, cos.(π/2 *x), legend=false, xlabel="norm. freq.",
4                             ylabel="|H|"))))
4 end
```

```
1 using FFTW
```

```

1  let
2      f = randn(500)
3      y = zeros(length(f))
4      for k = 1:(length(f)-1)
5          y[k] = 0.5 * (f[k] + f[k+1])
6      end
7
8      F = fftshift(fft(f))
9      Y = fftshift(fft(y))
10     H = Y ./ F
11     omega = fftshift(fftfreq(length(f)))
12
13     plot(f, label="orig", layout=2)
14     plot!(y, label="filtered", xlabel="Zeit", subplot=1, xlims=(0, 50))
15
16     plot!(omega, abs.(H), subplot=2, xlims=(0, 0.5), label="|H|", xlabel="Freq")
17 end

```

Der Filter filtert zwar wie erwartet, hat aber die unangenehme Eigenschaft, aus reelwertigen Eingaben komplexwertige Ausgaben zu machen. Das f_{k+1} führt zu einer Phasenverschiebung. Wir können das symmetrischer machen, indem wir auch das Element f_{k-1} mitnehmen, also

$$y_k = \frac{1}{3}(f_{k-1} + f_k + f_{k+1})$$

und

$$H(\omega) = \frac{1}{3} \left(e^{-i\omega\Delta t} + 1 + e^{i\omega\Delta t} \right) = \frac{1}{3} (1 + 2 \cos \omega\Delta t)$$

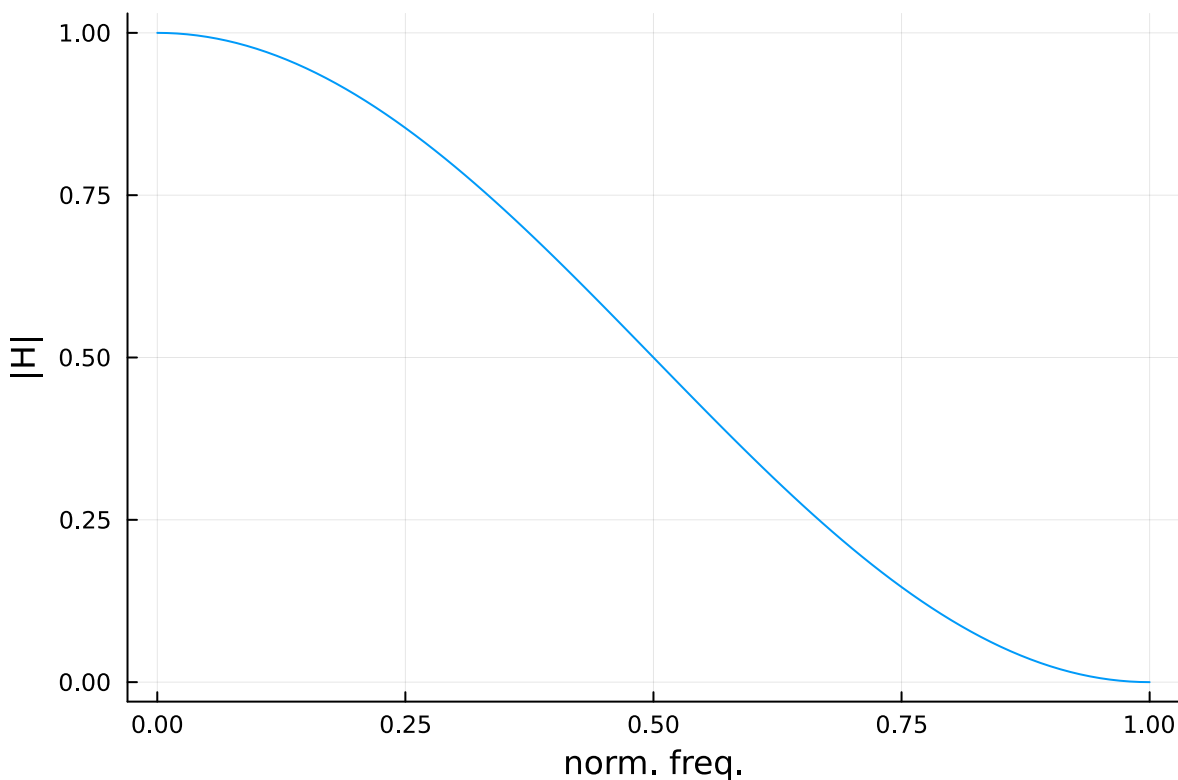
Jetzt ist H zwar rein reelwertig, aber leider negativ bei hohen Frequenzen. Schöner ist das mit

$$y_k = \frac{1}{4}(f_{k-1} + 2f_k + f_{k+1})$$

und

$$H(\omega) = \frac{1}{4} \left(e^{-i\omega\Delta t} + 2 + e^{i\omega\Delta t} \right) = \cos^2 \frac{\omega\Delta t}{2}$$

Dies ist die 'offizielle' Form des Tiefpassfilters.



```

1 let
2   x=(0:0.01:1)
3   aside(embed_display(plot(x, (cos.(pi/2 *x)).^2, legend=false, xlabel="norm.
4     freq.", ylabel="|H|"))))
5 end

```

Der -3dB-Punkt, also $H(\omega) = 1/2$ liegt bei $\omega = \Omega_{Nyq}/2$

Hochpass-Filter

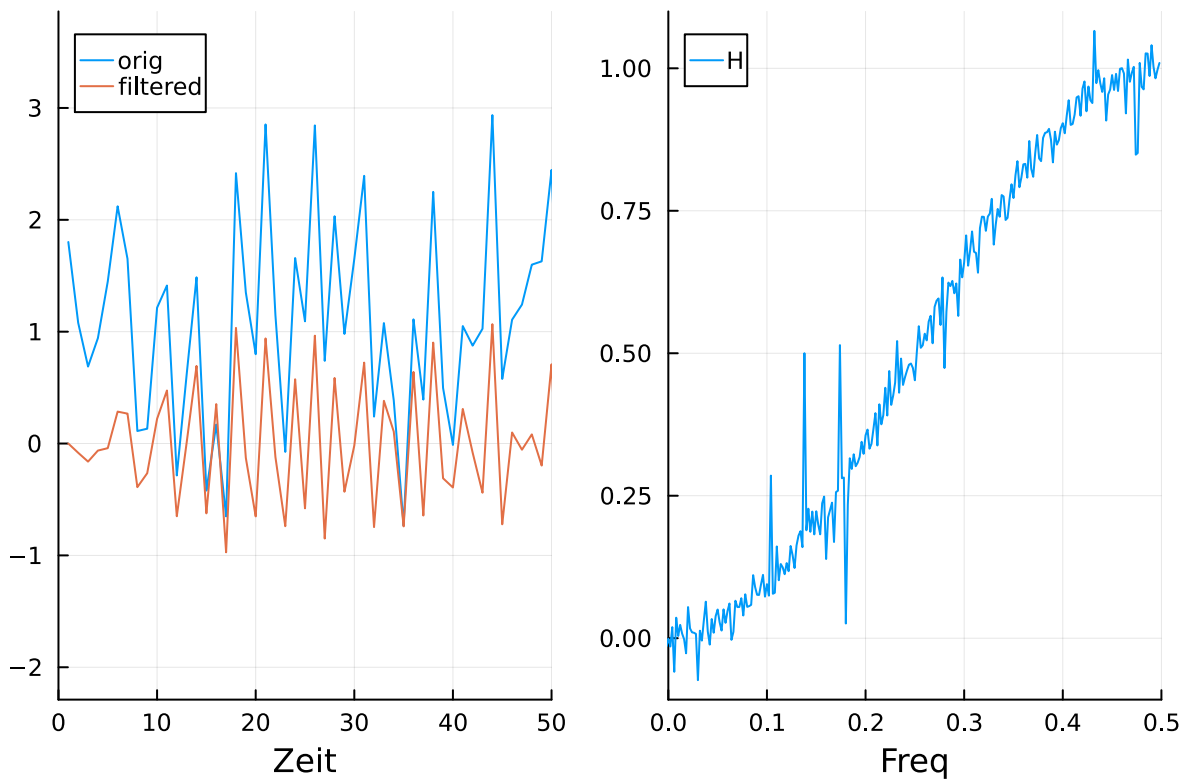
Analog kann man einen Hochpass-Filter erzeugen über

$$y_k = \frac{1}{4}(-f_{k-1} + 2f_k - f_{k+1})$$

und

$$H(\omega) = \frac{1}{4}(-e^{-i\omega\Delta t} + 2 - e^{i\omega\Delta t}) = 1 - \cos^2 \frac{\omega\Delta t}{2}$$

Eine konstanten Funktion $f_k = c$ (für alle k) liefert eine Null am Ausgang, also $y_k = 0$. Nur hohe Frequenzen werden durchgelassen.



```

1  let
2      f = 1 .+ randn(500)
3      y = zeros(length(f))
4      for k = 2:(length(f)-1)
5          y[k] = 0.25 * (- f[k-1] + 2 * f[k] - f[k+1])
6      end
7
8      F = fftshift(fft(f))
9      Y = fftshift(fft(y))
10     H = Y ./ F
11     omega = fftshift(fftfreq(length(f)))
12
13     plot(f, label="orig", layout=2)
14     plot!(y, label="filtered", xlabel="Zeit", subplot=1, xlims=(0, 50))
15
16     plot!(omega, real.(H) , subplot=2, xlims=(0, 0.5), label="H", xlabel="Freq")
17 end

```

Bandpass und Bandsperre

Bandpass-Filter lassen nur Frequenzen in einem Intervall durch. Bandsperren sind das Gegenteil, sperren also nur ein Frequenzband. Einfache Formen sind für den **Bandpass**

$$y_k = \frac{1}{16}(-f_{k-2} + 2f_k - f_{k+2})$$

und die **Bandsperre**

$$y_k = \frac{1}{16}(f_{k-2} + 14f_k + f_{k+2})$$

Rekursive Filter

Die bis hier betrachteten Filter waren *nicht-rekursiv*, weil die berechneten y_k nicht auch auf der Eingangsseite des Filters verwendet wurden. Diese Klasse der Filter heist auch *finite impulse response* oder FIR, weil eben die Impulsantwort h immer finit (kleiner unendlich) bleibt. Die Filter sind angenehm, aber es ist manchmal aufwendig (brauch viele Elemente a_l), um eine gewünschte Transferfunktion H zu erzeugen. In diesen Fällen sind *rekursive* Filter, also solche mit *infinite impulse response* (IIR) einfacher. Die Idee ist, y_k auch auf der rechten Seite der Gleichung zu verwenden, also eine Art Rückkopplung einzubauen. Dies führt aber unter Umständen zu einer Resonanz, einem instabilen Filter, oder eben Rückkopplungs-Pfeifen.

Mathematisch ist das

$$y_k = \sum_{l=-L}^L a_l V^l f_k - \sum_{m=-M}^M b_m V^m y_k$$

wobei $m = 0$ in der zweiten Summe ausgeschlossen ist, weil das ja schon links steht. Umsortiert ist das

$$\sum_{m=-M}^M b_m V^m y_k = \sum_{l=-L}^L a_l V^l f_k$$

was jetzt auch $m = 0$ beinhaltet.

Durch Fourier-Transformation erhalten wir

$$B_j Y_j = A_j F_j$$

mit den A_j definiert wie oben und den B_j analog. Die Transferfunktion ist dann (wieder mit kontinuierlichem ω)

$$H(\omega) = \frac{A(\omega)}{B(\omega)}$$

Die Rückkopplung erzeugt Nullstellen (oder Pole) in $B(\omega)$ und damit eine Resonanz in H . Den Resonanz-freien Fall von oben erhalten wir durch $b_m = 0$ für $m \neq 0$, also $B(\omega) = \text{const.}$

Bsp. nochmal Tiefpass

Wir (bzw. Butz) verbessern den Tiefpass von oben durch etwas Rückkopplung

$$y_k = \frac{1}{2} y_{k-1} + \frac{1}{4} (f_{k-1} + 2f_k + f_{k+1})$$

bzw.

$$\left(1 - \frac{1}{2} V^{-1}\right) y_k = \frac{1}{4} (V^{-1} + 2 + V^{+1}) f_k$$

Das ergibt

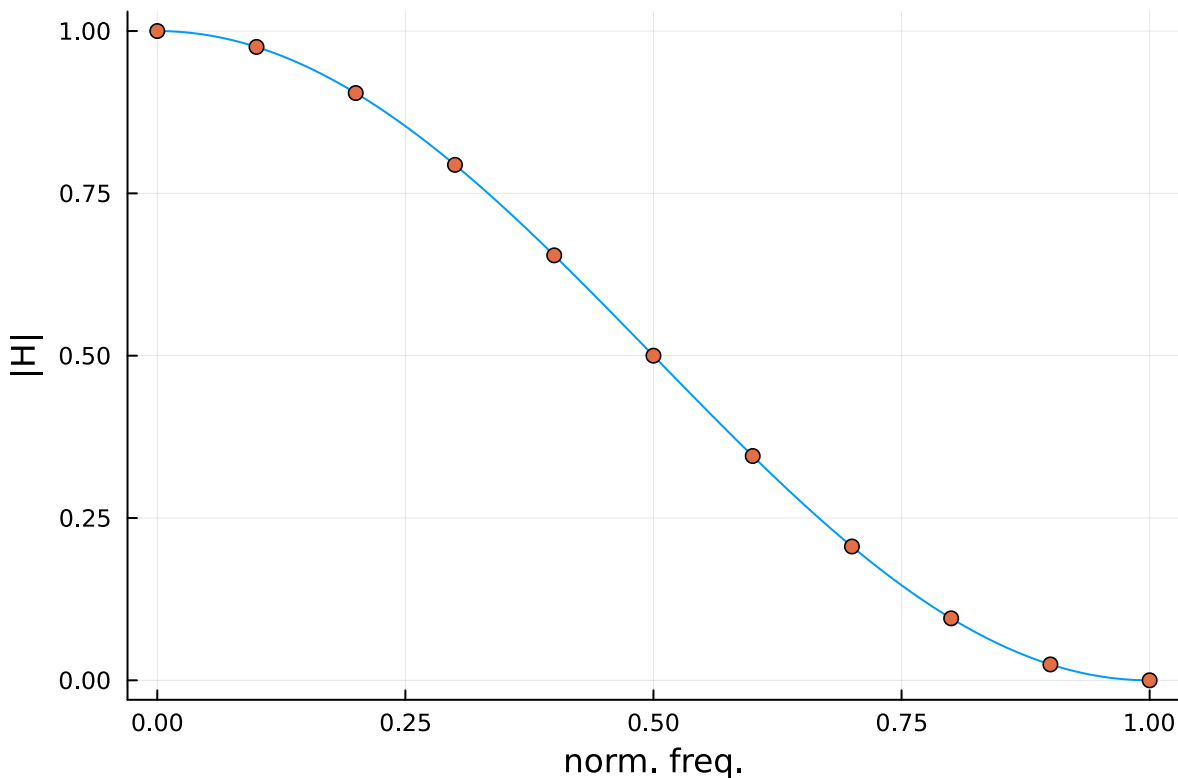
$$H(\omega) = \frac{\cos^2 \omega \Delta t / 2}{1 - \frac{1}{2} e^{-i\omega \Delta t}}$$

Die grafische Darstellung machen wir jetzt nicht von Hand, sondern nehmen eine Bibliothek.

Filter in Julia via DSP.jl

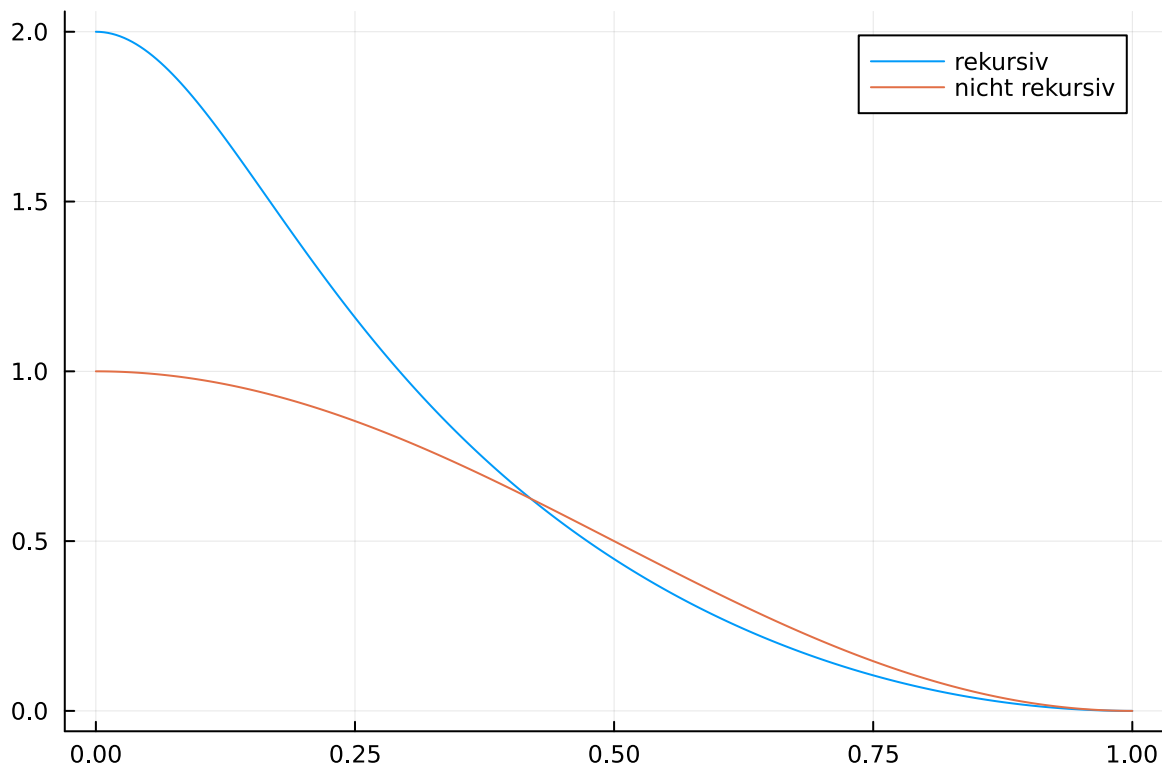
Das Paket `DSP.jl` implementiert alles was wir brauchen, um Filter darzustellen und anzuwenden. Werte in der Zukunft kennen wir natürlich nicht. Daher wird alles so verschoben, dass nie Indizes in die Zukunft verweisen und man gibt die Koeffizienten in der Reihenfolge a_0, a_{-1}, a_{-2} , etc. an (b analog). Unser nicht-rekursiver Tiefpass von oben mit $a_0 = 2$ und $a_{\pm 1} = 1$ wird hier also $a = [1, 2, 1]$ und $b = [1]$, also nur y_k kommt vor. Durch das Verschieben ist dann in `DSP.jl` allerdings H in diesem Fall nicht mehr reelwertig, sondern erhält einen Phasenfaktor. Der wird hier durch $e^{i\omega\Delta t}$ korrigiert.

```
1 using DSP
```



```
1 let
2     f = DSP.Filters.PolynomialRatio([1, 2, 1]./4, [1])
3     H, w = freqresp(f)
4     plot(w ./ pi, real.(H .* exp.(1im .* w)))
5
6     x = (0:0.1:1)
7     scatter!(x, (cos.(pi/2 * x)).^2, legend=false, xlabel="norm. freq.",
8             ylabel="|H|")
9 end
```

Nun ist es einfach, das rekursive Filter zu plotten. Wir sehen, dass der -3dB-Punkt bei niedrigeren Frequenzen liegt und der Verlauf steiler ist.



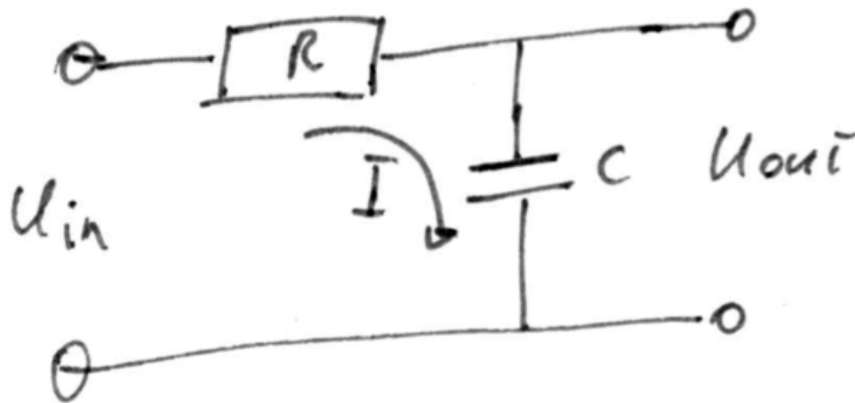
```

1 let
2   f = DSP.Filters.PolynomialRatio([1, 2, 1]./4, [1, -1/2])
3   H, w = freqresp(f)
4   plot(w ./ pi, abs.(H), label="rekursiv")
5
6   f = DSP.Filters.PolynomialRatio([1, 2, 1]./4, [1])
7   H, w = freqresp(f)
8   plot!(w ./ pi, abs.(H), label="nicht rekursiv")
9 end

```

RC-Tiefpass-Filter

Nun verbinden wir das mit physikalischen, also elektrischen Filtern. Wir betrachten auf verschiedene Weisen einen unbelasteten RC-Tiefpass-Filter. Es wird also nur eine Spannung U_{out} gemessen und kein Strom I_{out} fließt.



Schaltbild eines RC Filters

Komplexe Widerstände

Der RC-Filter bildet einen Spannungsteiler aus zwei komplexen Widerständen $Z_1 = R$ und $Z_2 = 1/(i\omega C)$. Damit ergibt sie die Transferfunktion H zu

$$H = \frac{U_{out}}{U_{in}} = \frac{Z_2}{Z_1 + Z_2} = \frac{\frac{1}{i\omega C}}{R + \frac{1}{i\omega C}} = \frac{\frac{1}{i\omega RC}}{1 + \frac{1}{i\omega RC}}$$

Zeitliche Integration

Alternativ kann man die Spannung U_{out} auch aus der Ladung Q des Kondensators berechnen

$$U_{out}(t) = \frac{Q(t)}{C} = \frac{1}{C} \int_{\tau < t} I(\tau) d\tau = \frac{1}{RC} \int_{\tau < t} U_{in}(\tau) - U_{out}(\tau) d\tau$$

Für kleine Δt mit quasi konstanten U ergibt sich so

$$U_{out}(t + \Delta t) = U_{out}(t) + \frac{\Delta t}{RC} (U_{in}(t) - U_{out}(t))$$

Nun seien x_n und y_n die Zeitreihe der Spannungen, also

$$x_n = U_{in}(n\Delta t) \quad \text{und} \quad y_n = U_{out}(n\Delta t)$$

also

$$y_{n+1} = y_n + \frac{\Delta t}{RC} (x_n - y_n)$$

oder umgeformt , inkl. Verschieben des Indexes um 1

$$y_n + y_{n-1} \left(1 + \frac{\Delta t}{RC} \right) = \frac{\Delta t}{RC} x_{n-1}$$

Mit dem Verschiebe-Operator V^l lässt sich das schreiben als

$$\left(1 + V^{-1} \left(1 + \frac{\Delta t}{RC} \right) \right) y_n = \frac{\Delta t}{RC} V^{-1} x_n$$

Die einzigen von Null verschiedenen Filter-Koeffizienten sind also

$$\begin{aligned} a_{-1} &= \frac{\Delta t}{RC} \\ b_0 &= 1 \\ b_{-1} &= 1 - \frac{\Delta t}{RC} \end{aligned}$$

Die $A(\omega)$ und $B(\omega)$ sind diese Koeffizienten multipliziert mit $\exp(i\omega\Delta t)$ also

$$A(\omega) = \sum a_n \exp(n i\omega\Delta t) = \frac{\Delta t}{RC} \exp(-i\omega\Delta t)$$
$$B(\omega) = \sum b_n \exp(n i\omega\Delta t) = 1 + \left(1 - \frac{\Delta t}{RC}\right) \exp(-i\omega\Delta t)$$

Mit der Näherung $\exp(-i\omega\Delta t) = 1/(1 + i\omega\Delta t)$ ergibt sich für die Transferfunktion $H(\omega)$

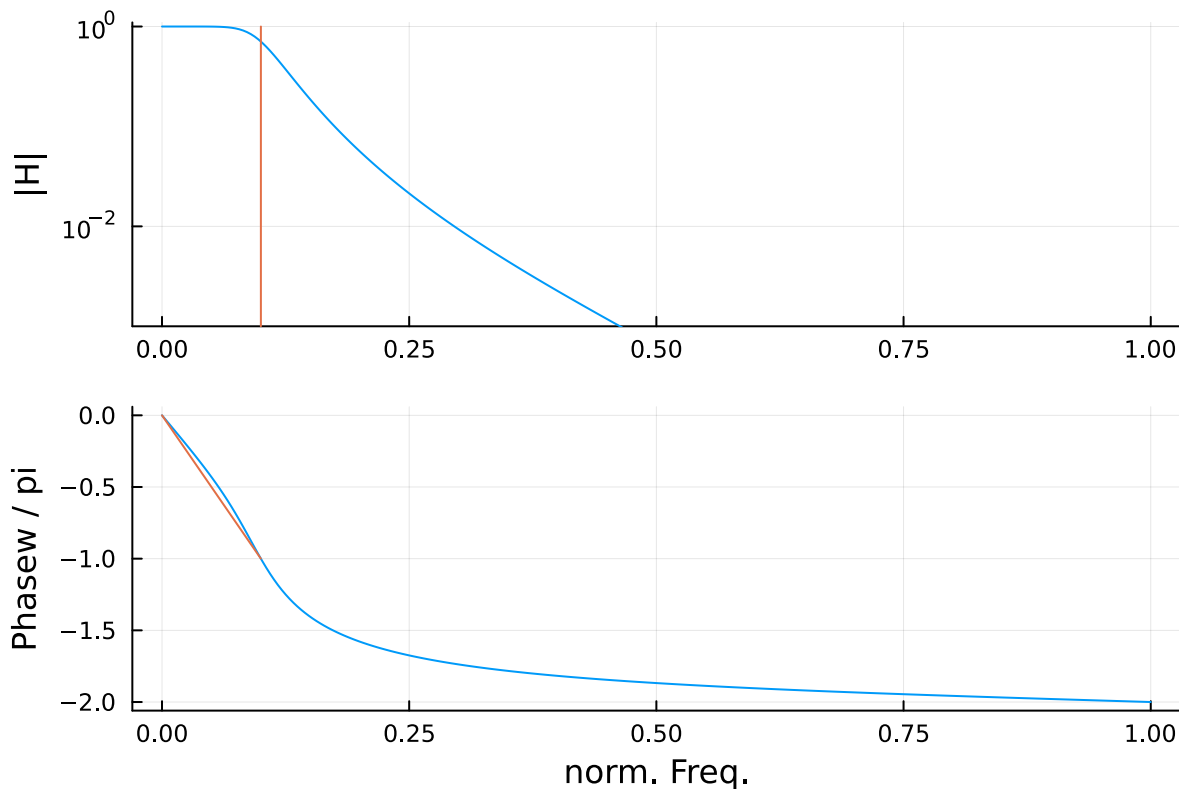
$$H(\omega) = \frac{A(\omega)}{B(\omega)} = \frac{\frac{\Delta t}{RC} \cdot \frac{1}{1+i\omega\Delta t}}{1 + \left(1 - \frac{\Delta t}{RC}\right) \cdot \frac{1}{1+i\omega\Delta t}} \approx \frac{\frac{1}{i\omega RC}}{1 + \frac{1}{i\omega RC}}$$

Die letzte Näherung erfordert, dass $\Delta t/RC \ll 1$, also sehr kleines Δt . Das Ergebnis stimmt also mit dem über die komplexen Widerstände überein.

Filter-Design

Wie kommt man auf die Koeffizienten a_i und b_i bzw. wie verschaltet man Spulen, Kondensatoren und Widerstände, so dass sich eine gewünschte Filterfunktion einstellt. Dazu gibt es 'Kochrezepte', die Sie hier ausprobieren können. Relevante Größen sind die Steilheit der Flanke (roll-off) und die Welligkeit des transmittierten bzw. geblockten Bereichs (pass-band / stop-band ripple). Auch die Phasenfunktion ist unterschiedliche. Eine konstante Ableitung der Phase mit der Frequenz ergibt eine konstante Gruppengeschwindigkeit, also werden Pulse nicht verzerrt.

Filter-Typ Butterworth ▼



```

1 begin
2     responsetype = Lowpass(0.1)
3     if (filtertyp == "Butterworth")
4         designmethod = Butterworth(4)
5     elseif (filtertyp == "Chebyshev")
6         designmethod = Chebyshev2(4, 30)
7     else
8         designmethod = Elliptic(4, 2, 30)
9     end
10
11     myfilter = digitalfilter(responsetype, designmethod)
12
13     H, w = freqresp(myfilter)
14     plot(w ./ pi, abs.(H), layout=(2,1), subplot=1, leg=false, yaxis=(log, (1e-3, 1.1)), ylabel = "|H|")
15     plot!([0.1, 0.1], [1e-3, 1], subplot=1)
16
17     phi, w = phaseresp(myfilter)
18     plot!(w ./ pi, phi ./ pi, subplot=2, leg=false, xlabel="norm. Freq.", ylabel="Phasew / pi")
19     plot!([0, 0.1], [0, -1], subplot=2)
20
21 end

```

Dies sind die zugehörigen Koeffizienten a und b

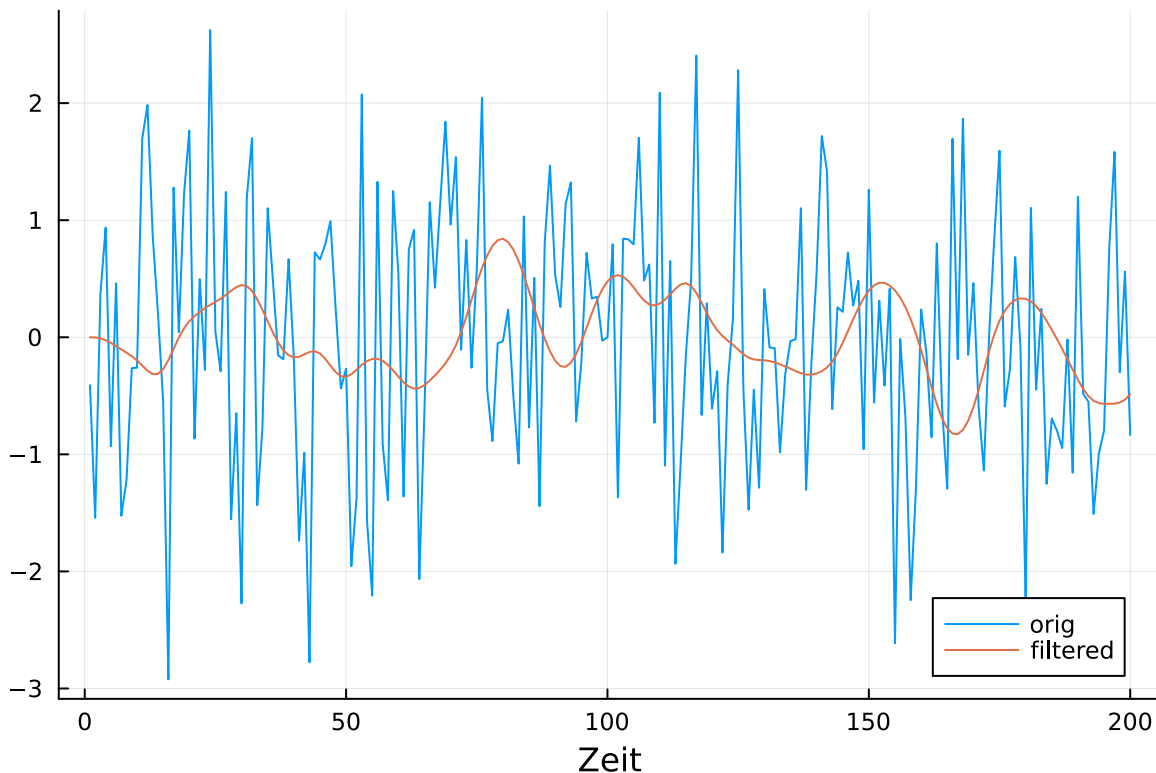
```

▶ ([0.000416599, 0.0016664, 0.0024996, 0.0016664, 0.000416599], [1.0, -3.18064, 3.86119,
1 begin
2   tf = convert(PolynomialRatio, myfilter)
3   numerator_coefs = coefb(tf)
4   denominator_coefs = coefa(tf)
5   (numerator_coefs, denominator_coefs)
6 end

```

Filter anwenden

Die Funktion 'filt' wendet den Filter an



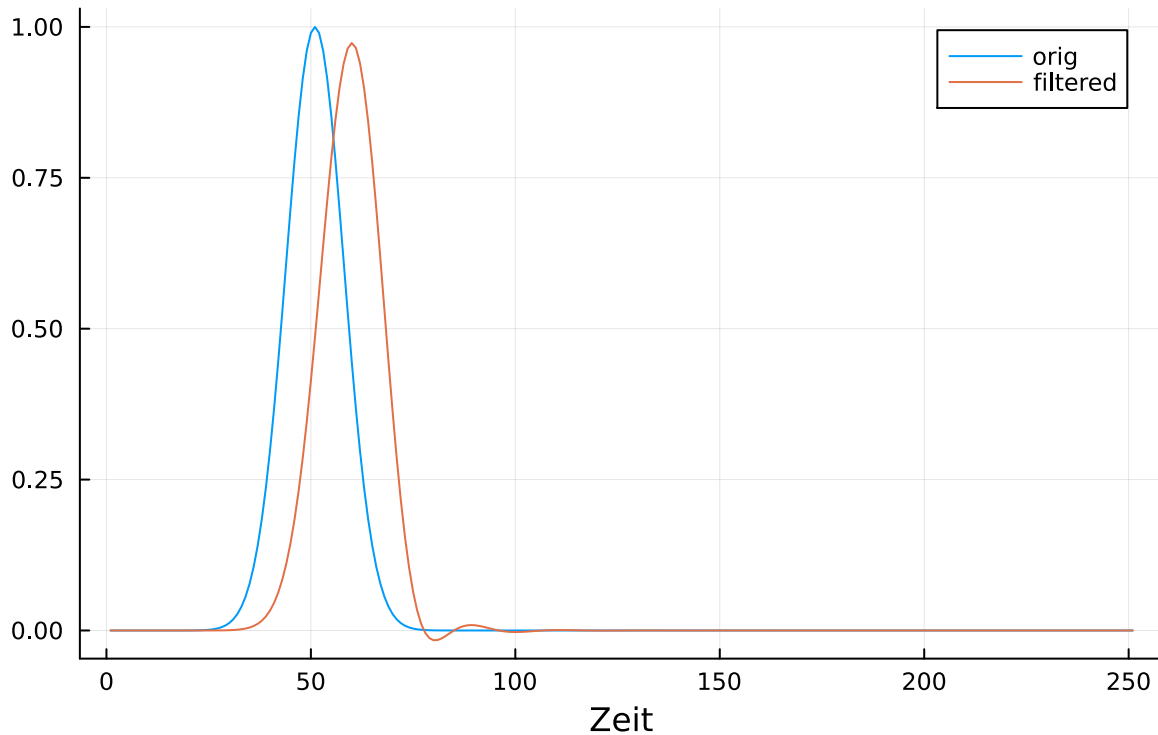
```

1 let
2   f = randn(200)
3   y = filt(myfilter, f)
4   plot(f, label="orig")
5   plot!(y, label="filtered", xlabel="Zeit")
6 end

```

Hier sieht man auch, wie der Phasenverlauf den Puls verzerren kann (dazu weiter oben den Filter ändern).

Gauss gefiltert mit Butterworth



```
1 let
2   f = exp.(-1 .* (-50:200) .^ 2 ./ (10 .^ 2) )
3   y = filt(myfilter, f)
4   plot(f, label="orig")
5   plot!(y, label="filtered", xlabel="Zeit", title="Gauss gefiltert mit
6     $(filtertyp)")
6 end
```

```
1 using PlutoUI, Plots
```

```
1 aside(x) = PlutoUI.ExperimentalLayout.aside(x);
```

☰ Inhalt

Überblick

Zeitliche Antwort

Frequenz-Antwort

- Kramers-Kronig-Beziehung
- Kaskadierte Systeme
- Entfaltung

Reihenentwicklung der Transferfunktion

Diskrete Datenreihe

Beispiel: Tiefpass-Filter

- Hochpass-Filter
- Bandpass und Bandsperre

Rekursive Filter

- Bsp. nochmal Tiefpass

Filter in Julia via DSP.jl

RC-Tiefpass-Filter

- Komplexe Widerstände
- Zeitliche Integration

Filter-Design

Filter anwenden

1 `TableOfContents(title="Inhalt")`