

R. Brigola, TH Nürnberg Georg Simon Ohm, 2014

Mathematica - Notebooks als Bonusmaterial zum Lehrbuch

[1] Rolf Brigola *Fourier-Analyse und Distributionen,
Eine Einführung mit Anwendungen,
edition swk, Hamburg 2013*

Beispiel zum Entwurf von diskreten linearen Filtern nach der Methode der Bilinearen Transformation mit *Mathematica*

Referenzen zu Kapiteln, Abschnitten, Seitenzahlen beziehen sich im Folgenden meist auf das genannte Lehrbuch des Autors. Einige wenige weitere Referenzen findet man am Ende des Notebooks.

Die URL aller meiner *Mathematica*-Notebooks zur Fourier-Analyse ist:
<http://www.stiftung-swk.de/mathematica/>

In diesem Notebook wird gezeigt, wie man diskrete lineare Filter nach der Methode der bilinearen Transformation entwirft. *Dabei folgen wir der Beschreibung in [1], Kap.10.*

Wir beginnen damit, ein Beispiel eines diskreten Filters zunächst eigenständig zu berechnen und danach anzusehen, was die bereits vorhandenen *Mathematica*-Routinen zum Thema anbieten.

Diskrete IIR – Filter

Eine naheliegende Idee, ein diskretes Filter zu entwerfen, ist es, *vom Frequenzgang eines zu einem gegebenen Dämpfungsplan passenden Analogfilters auszugehen* und daraus ein entsprechendes diskretes Filter zu berechnen. Es gibt verschiedene Möglichkeiten dies zu bewerkstelligen. Ich werde hier die **Methode der “Bilinearen Transformation”** am Beispiel eines Butterworth-Tiefpassfilters vorführen.

Für diesen Filtertyp haben wir in einem vorangehenden Notebook schon gesehen, wie man ihn zu einem gegebenem Dämpfungsplan als Analogfilter entwerfen kann. Wir konstruieren mit der bilinearen Transformation ein sog. IIR-Filter. Auf diskrete FIR-Filter gehe ich dann in einem anderen Notebook ein.

Betrachtet wird jeweils ein zeitinvariantes diskretes lineares System, bei dem der System-Output durch Faltung des Inputs mit der Impulsantwort des Systems gegeben ist. Beispiele, dass dies trotz eines weit verbreiteten Glaubens (*und sämtlich falscher angeblicher “Beweise” dafür*) nicht notwendigerweise immer so sein muss, finden Sie in [1], Kap 10.

Es ist daher nützlich, sich Gedanken darüber zu machen, zwischen welchen Signalmäumen der

lineare Operator, der das System beschreibt, arbeitet und welche Topologien (Konvergenzbegriffe) man in diesen Räumen verwendet.

Ich verwende im Folgenden als *Signalraum für Inputs und Outputs den Raum l_d^∞ aller Impulsfolgen $x = \sum_{n=-\infty}^{+\infty} x_n \delta(t-nA)$ mit beschränkter Koeffizientenfolge (x_n) für $n \in \mathbb{Z}$ mit der sup-Norm $\|x\| = \sup\{|x_n|, n \in \mathbb{Z}\}$. Damit das sinnvoll ist, muss die Impulsantwort, mit der gefaltet werden soll, absolut-summierbare Koeffizienten haben, was im Beispiel so sein wird. Das Filter ist dann stabil, der Frequenzgang stetig. Für andere Signalräume und mehr Details sei auf [1] verwiesen.*

$\delta(t-nA)$ sind **Dirac-Distributionen**, t ein Zeitparameter, A ein fester Zeitschritt, der die Abtast-rate charakterisiert, mit der das diskrete System arbeitet. Mit diesen Vorgaben gehören alle vorkommenden Impulsfolgen auch zu S' und sind damit auch Fourier-transformierbar. Für die Theorie steht damit alles zur Verfügung, was in vorangehenden Notebooks über Distributionen und die Fouriertransformation schon demonstriert wurde. Wir konstruieren diskrete Filter so, dass der Zusammenhang zwischen Input und Output durch einen Faltungsoperator $L: l_d^\infty \rightarrow l_d^\infty$ gegeben ist, d.h. es gilt:

$$Lx = h * x \text{ mit der Impulsantwort } h=L\delta$$

1. Die bilineare Transformation zur Abbildung des Frequenzgangs eines Analogfilters auf den eines diskreten Filters

1.1 Übertragungsfunktion und Frequenzgang diskreter linearer Filter, z-Transformation

Der Frequenzgang ist die Fouriertransformierte der Impulsantwort h . Hat h die Koeffizienten h_k , dann ist der **Frequenzgang die Fourierreihe $\sum_{k=-\infty}^{+\infty} h_k e^{-ik\omega A}$** , A der fixierte Zeittakt des Systems. Die Laurentreihe $H(z) = \sum_{k=-\infty}^{+\infty} h_k z^{-k}$ heißt **Übertragungsfunktion** des Filters. Sie ist die **z-Transformierte der Impulsantwort. Der Frequenzgang ist $H(e^{i\omega A})$.**

1.2 Bilineare Transformation

Aus verschiedenen Gründen sind Filter mit rationalen Frequenzgängen Standard in der Übertragungstechnik. Gründe sind z.B. die Tatsache, dass man im Analogfall solche Filter durch lineare DGL'en mit konstanten Koeffizienten beschreiben kann, dass die Filteroperatoren dann Faltungsoperatoren sind, dass die Fouriertransformation rationaler Funktionen einfach möglich ist und einfache schaltungstechnische Realisierungsmöglichkeiten vorliegen.

Für Analogfilter (mit einer Variablen) setzen wir also gewöhnliche DGL'en mit konstanten Koeffizienten, die den Input-Output-Zusammenhang beschreiben, voraus. Sind die Filter stabil - und gerade solche sind in technischen Anwendungen erwünscht - dann ist die Übertragungs-

funktion $R(s)$, $s \in \mathbb{C}$, gegeben durch $R(s) = 1/P(s)$, P das charakteristische Polynom der DGL. Alle Pole von R müssen dann negative Realteile besitzen (Stabilität) und der zugehörige Frequenzgang ist $R(i\omega)$.

Gegeben sei nun der Frequenzgang $R(i\omega)$ für ein stabiles analoges lineares Filter. Um daraus den **$2\pi/A$ -periodischen Frequenzgang** eines entsprechenden diskreten Filters zu gewinnen, bildet man die Frequenzachse des Analogfilters bijektiv auf das Intervall $]-\pi/A, \pi/A[$ ab. Eine Abbildung T , die das leistet, ist

$$(1) \quad T(\omega) = \frac{2}{A} \arctan\left(\frac{\omega A}{2V}\right) \quad \text{mit der Umkehrabbildung} \quad T^{-1}(\Omega) = \frac{2V}{A} \tan\left(\frac{\Omega A}{2}\right) = \omega$$

Ω bezeichnet die Kreisfrequenz im diskreten Fall, $1/A$ die Abtastfrequenz des diskreten Systems, V ist ein Faktor, mit dem eine "Vorverzerrung" gewählt werden kann, so dass z.B. eine gewünschte Grenzkreisfrequenz ω_g ein Fixpunkt bei dieser Abbildung ist. Damit kann man nun den Frequenzgang $H(e^{i\Omega A})$ des gewünschten diskreten Filters definieren durch

$$(2) \quad H(e^{i\Omega A}) = R(i T^{-1}(\Omega)) \quad \text{für } \Omega \in]-\pi/A, \pi/A[.$$

Aus Additionstheoremen für die tan-Funktion folgt (vgl. [1], Abschnitt 10.6) mit $z = e^{i\Omega A}$

$$(3) \quad H(e^{i\Omega A}) = H(z) = R\left(i \frac{2V}{A} \tan\left(\frac{\Omega A}{2}\right)\right) = R\left(\frac{2V}{A} \frac{1-z^{-1}}{1+z^{-1}}\right) = R(B(z))$$

Die Abbildung $B(z) = \frac{2V}{A} \frac{1-z^{-1}}{1+z^{-1}} = s$ für $z \in \mathbb{C}$, $B(-1) = \infty$, $B(\infty) = \frac{2V}{A}$ von der kompaktifizierten komplexen Ebene in sich ist bijektiv mit der Umkehrabbildung $B^{-1}(s) = \frac{2V/A+s}{2V/A-s} = z$.

Diese Möbius-Abbildung heißt bilineare Transformation B.

Die Gleichung (3) zeigt, dass $H(z)$ rational in z ist. Warum ist das von Bedeutung?

Antwort: Wenn das analoge Filter kausal ist, dann ist auch das wie oben entworfene Filter kausal und wird "im Zeitbereich" mit der Normierung $b_0 = 1$ durch eine Differenzengleichung der Form

$$(4) \quad y_n = a_0 x_n + \sum_{m=1}^N (a_m x_{n-m} - b_m y_{n-m})$$

beschrieben. Damit kann man das Filter sofort als Software realisieren. Die x_n sind dabei die Input-Samples, die y_n die Output-Samples. Ich komme weiter unten darauf zurück. Die a_m und b_m sind einfach die Koeffizienten der Polynome in $H(z)$, (b_0 auf $b_0=1$ normiert).

Da der Zusammenhang zwischen Übertragungsfunktion und Differenzengleichung einfach und theoretisch geklärt ist (vgl. [1]), brauchen wir im Folgenden eine z-Transformation oder ihre Inverse nirgendwo explizit durchzuführen.

Zunächst nun ein paar konkrete Vorgaben für ein Tiefpassfilter, das wir in der Folge berechnen wollen; damit veranschaulichen wir, wie die Abbildung T die Frequenzachse

eines Analogfilters auf das Intervall $]-\pi/A, \pi/A[$ abbildet, und wie die Transformationen B und B^{-1} die kompaktifizierte komplexe Ebene abbilden: Mit B^{-1} wird die imaginäre Achse auf die Einheitskreislinie abgebildet, die linke Halbebene in das Innere des Einheitskreises.

Wir wählen:

$1/A = 10^4$ Hz als Abtastfrequenz für das diskrete System

$f_g = 100$ Hz als Grenzfrequenz

V so gewählt, dass f_g ein Fixpunkt bei der Frequenzgangtransformation wird

Bemerkung: Auch wenn man die Größe A im Argument der arctan-Funktion kürzen könnte, führe ich sie hier lieber mit, damit man deutlich sieht, dass die Argumente der Funktionen "physikalisch einheitenfrei" (wie $\omega_g \cdot A$ etwa) sind, da ja Funktionen, die wir verwenden, wie die exp-Funktion als Potenzreihen nur Sinn machen, wenn die Argumente eben einheitenfrei sind. Außerdem kann man deutlich sehen, welchen Einfluss die Abtastrate auf die Bandbreite der Filter hat, wo sie eingeht (in der Vorverzerrung V etwa) oder eben nicht. Ich sehe es als gute Rechenkontrolle, es macht keine Mühe, sie mitzunehmen und man muss dann auch nicht mehr darüber nachdenken, wo evtl. Normierungen wieder rückgängig zu machen sind, wenn man mit konkreten Daten arbeitet.

1) Veranschaulichung der Transformation der Frequenzachse

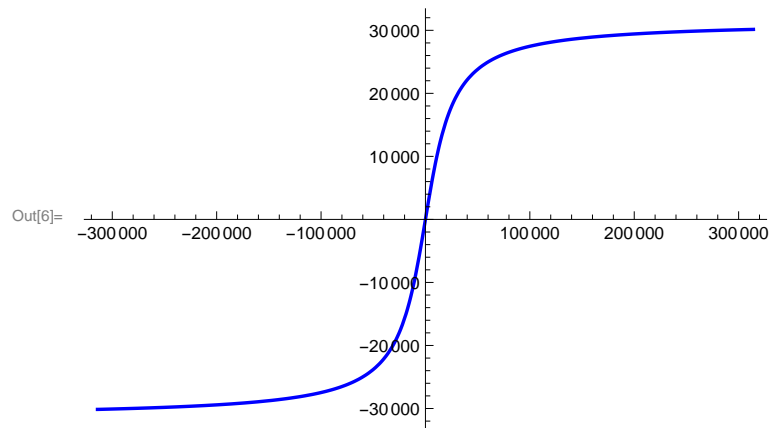
```
In[1]:= ClearAll["Global`*"]; Remove["Global`*"];
A := 10^-4;  $\omega_g = 200 \pi$ ; V =  $\omega_g A / 2 \cot[\omega_g A / 2]$ ;
N[ $\pi / A$ ] (* zeigt den Wert der oberen Asymptote von T *)
T[ $\omega_$ ] = 2 / A ArcTan[ $\omega A / (2 V)$ ] (* Abbildung von  $\mathbb{R}$  auf  $]-\pi/A, \pi/A [$  *)
T[200  $\pi$ ] (* zeigt den Fixpunkt  $f_g=200\pi$  der Abbildung *)
Plot[T[ $\omega$ ], { $\omega$ , -100 000  $\pi$ , 100 000  $\pi$ }, PlotRange -> All,
PlotStyle -> Directive[Blue, Thickness[0.005]]]
(* zeigt den Graph der Abbildung T *)
```

Remove::rmnsm : There are no symbols matching "Global`*". >>

Out[3]= 31 415.9

Out[4]= $20\,000 \operatorname{ArcTan}\left[\frac{\omega \operatorname{Tan}\left[\frac{\pi}{100}\right]}{200 \pi}\right]$

Out[5]= 200 π



2) Veranschaulichung zur bilinearen Transformation

Hier zur Veranschaulichung, wie die bilineare Transformation wirkt: Die Halbebene von \mathbb{C} aller z mit negativen Realteilen wird durch B^{-1} in das Innere, die rechte Halbebene in das Äußere des Einheitskreises abgebildet, die imaginäre Achse auf die Kreislinie.

Konsequenz: *Alle Pole der Übertragungsfunktion $H(z)$ eines stabilen so konstruierten diskreten Filters müssen im Inneren des Einheitskreises liegen.* In [1] wird gezeigt, dass dies generell für jedes stabile diskrete Filter mit rationaler Übertragungsfunktion so sein muss. Soll das Filter überdies stabil invertierbar sein - wichtig für die Rekonstruktion gefilterter Daten - dann müssen auch alle Nullstellen von $H(z)$ innerhalb des Einheitskreises liegen.

```

In[7]:= Binv[x_, y_] := (2 V / A + (x + i y)) / (2 V / A - (x + i y))
Binv[0, 0] (* B-1(0) *)
Limit[Binv[0, y], y → ∞] (* B-1(∞) *)
Limit[Binv[0, y], y → -∞] (* B-1(∞) =
  B-1(+∞): Die imaginäre Achse wird zur Einheitskreislinie zusammengebogen *)
N[Abs[Binv[-3, 3]]]
(* Betrag des Bilds einer komplexen Zahl mit negativem Realteil *)
N[Abs[Binv[1, 1]]]
(* Betrag des Bilds einer komplexen Zahl mit positivem Realteil *)
plot1 = ParametricPlot[{Re[Binv[0, y]], Im[Binv[0, y]]},
  {y, -500 000, 50 000}, PlotRange → All,
  PlotStyle → Directive[Blue, Thickness[0.005]], PlotPoints → 200]
(* Bild eines Ausschnitts der imaginären Achse unter B-1 *)

```

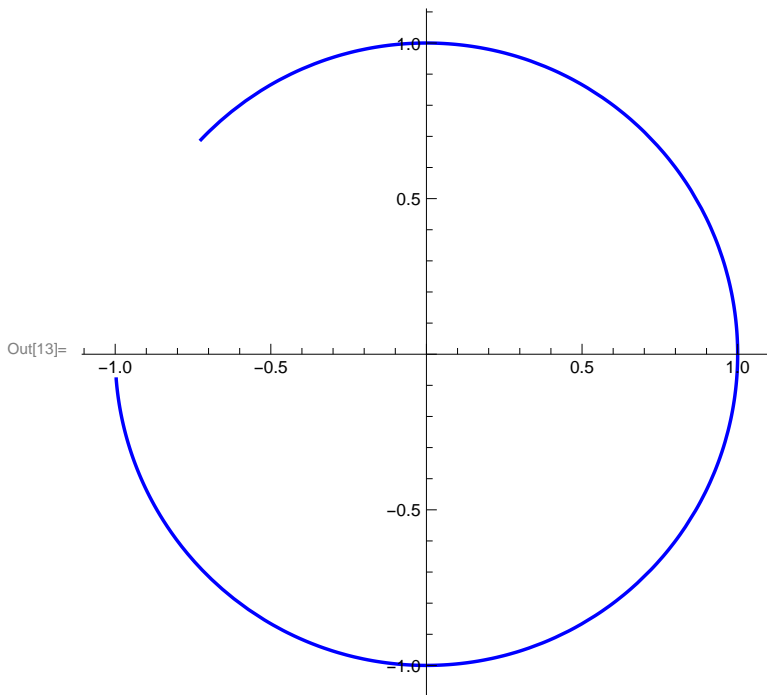
Out[8]= 1

Out[9]= -1

Out[10]= -1

Out[11]= 0.9997

Out[12]= 1.0001



In[14]:=

2. Beispiel für den Entwurf eines diskreten Tiefpassfilters mit der bilinearen Transformation

Wir berechnen nach der vorgestellten Methode die Koeffizienten der rationalen Übertragungsfunktion $H(z) = \frac{\sum_{m=0}^N a_m z^{-m}}{\sum_{m=0}^N b_m z^{-m}}$ des resultierenden diskreten Filters. Wir sehen, dass das berechnete Filter kausal ist, im Fall stabiler Analogfilter, die wir heranziehen, ebenfalls stabil, und dass man allein mit den Koeffizienten a_k , b_k unmittelbar eine Realisierung mit der zugehörigen Differenzgleichung nach Normierung des Koeffizienten b_0 auf $b_0 = 1$ (vgl. ggf. [1], S. 304) programmieren kann:

$$(4) \quad y_n = a_0 x_n + \sum_{m=1}^N (a_m x_{n-m} - b_m y_{n-m})$$

für Input-Samples x_n und Output-Samples y_n und verschwindende Anfangsbedingungen.

Beispiel. Butterworth-Tiefpassfilter 3. Ordnung

Gegeben sei der Frequenzgang R eines Butterworth-Tiefpassfilters 3. Ordnung, d.h.

$$R(\omega) = K / (1 + 2s + 2s^2 + s^3) \text{ mit } s = i\omega / \omega_g \text{ für die Grenzkreisfrequenz } \omega_g.$$

Wir berechnen die Koeffizienten von Zähler- und Nenner-Polynom des diskreten Filters, das durch bilineare Transformation entsteht. Die Vorverzerrung wird so gewählt, dass

$$R(\omega_g) = H(e^{i\omega_g A}) \text{ gilt. In beiden Fällen erhalten wir dann bei } \omega_g \text{ eine 3 dB-Dämpfung.}$$

Für den nachfolgenden Plot wählen wir $K=1$, $\omega_g=200\pi$, $A=1/500$ und setzen im Folgenden zur Übersichtlichkeit $L = \cot(\omega_g A/2)$.

Nun zur Berechnung:

Um eine möglichst übersichtliche Darstellung mit *Mathematica* zu erreichen,

substituiere ich z^{-1} durch w und verwende den *Mathematica*-Befehl **Cancel**:

Das Ergebnis ist dann eine in w rationale Funktion aus der wir mit *Mathematica*

leicht die benötigten Koeffizienten im Zähler und im Nenner als Listen extrahieren können.

```

In[15]= ClearAll["Global`*"]; Remove["Global`*"];
R[ω_] := K / (1 + 2 i ω / wg + 2 (i ω / wg)2 + (i ω / wg)3);
(* Butterworth-Frequenzgang mit Grenzkreisfrequenz wg*)
B[w_] = wg L  $\frac{1-w}{1+w}$ ; (* bilineare Transformation, w=z-1,
L=cot(wg A/2); 2V/A mit V=ωgA/2cot[wg A/2] gekürzt *)
HH[w_] = Cancel[R[B[w]/i]]
(* in w rationale Übertragungsfunktion des diskreten Filters *)
zaehlerkoeff = CoefficientList[Numerator[%], w]
(* Unnormierte Koeffizientenliste des Zählerpolynoms *)
nennerkoeff = CoefficientList[Denominator[%%], w]
(* Unnormierte Koeffizientenliste des Nennerpolynoms *)
Out[18]= -((K (1 + w)3) / (-1 - 2 L - 2 L2 - L3 - 3 w - 2 L w + 2 L2 w +
3 L3 w - 3 w2 + 2 L w2 + 2 L2 w2 - 3 L3 w2 - w3 + 2 L w3 - 2 L2 w3 + L3 w3))
Out[19]= {-K, -3 K, -3 K, -K}
Out[20]= {-1 - 2 L - 2 L2 - L3, -3 - 2 L + 2 L2 + 3 L3, -3 + 2 L + 2 L2 - 3 L3, -1 + 2 L - 2 L2 + L3}

```

Nun Plots mit den konkreten Daten von oben

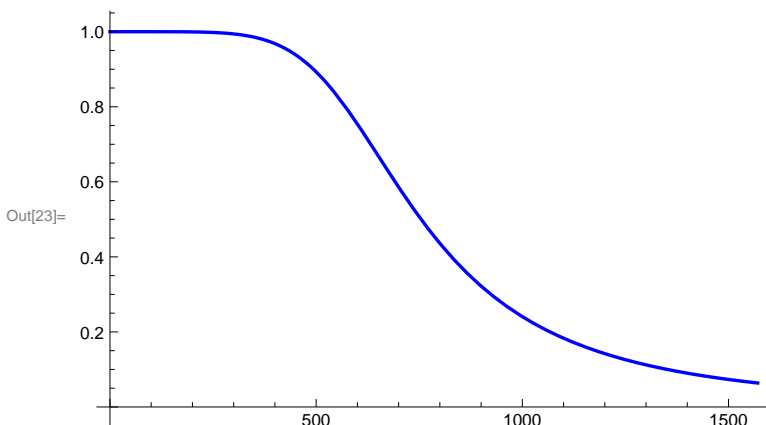
a) Der Amplitudengang des Analogfilters und die Dämpfung in dB bei der Grenzfrequenz

```

In[21]= K := 1; wg := 200 π; A := 1 / 500; L := Cot[wg A / 2]; (* Vorgaben *)
-20 Log10[N[Abs[R[ wg]]]]
(* Dämpfung in dB bei der Grenzkreisfrequenz wg *)
plot1 = Plot[Abs[R[ ω]], {ω, 0, 2.5 wg},
PlotStyle → Directive[Blue, Thickness[0.005]]
(* Analog-Amplitudengang als Funktion der Kreisfrequenz ω *)

```

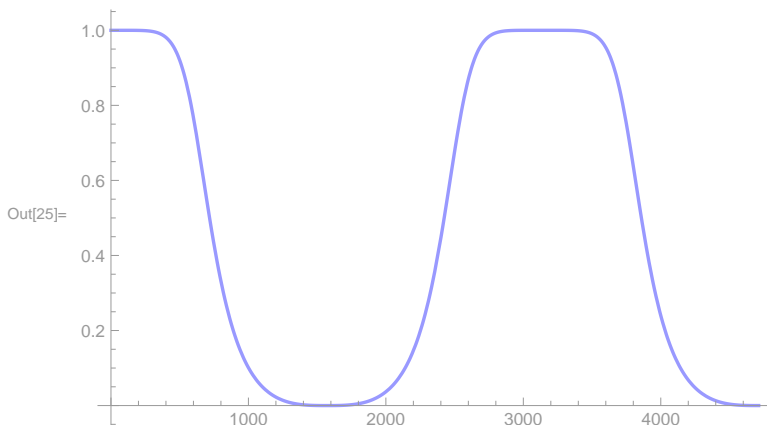
Out[22]= 3.0103



b) Der $2\pi/A$ -periodische Amplitudengang des diskreten Filters und die Dämpfung in dB bei der Grenzfrequenz

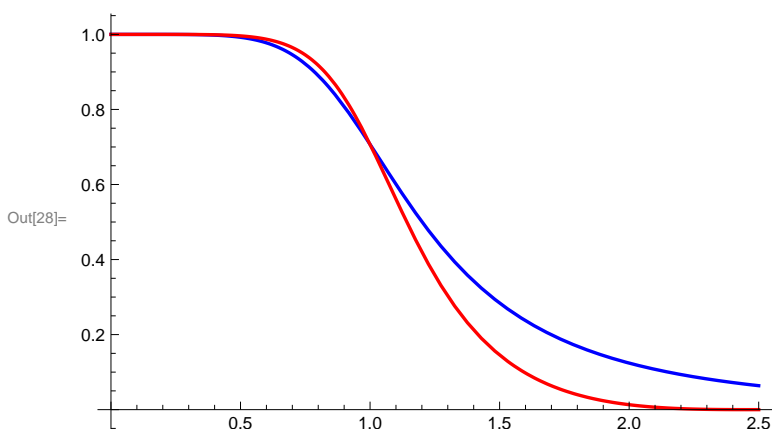
```
- 20 Log10 [N[Abs[HH[Exp[-I wg A]]]] ]
(* Dämpfung des diskreten Filters bei der Grenzfrequenz *)
plot2 = Plot[Abs[HH[Exp[-I Ω A]]], {Ω, 0, 1500 π},
  PlotStyle → Directive[Blue, Thickness[0.005]] ]
(* 2π/A-periodischer Amplitudengang des diskreten Filters *)
```

Out[24]= 3.0103



Hier nochmal in einem gemeinsamen Bild zum Vergleich der Analog-Amplitudengang (blau) und der des diskreten Filters (rot) im Frequenzband $[0, 2.5 \text{ wg}]$, wg auf Eins normiert

```
In[26]= plot1a :=
  Plot[Abs[R[s wg]], {s, 0, 2.5}, PlotStyle → Directive[Blue, Thickness[0.005]]]
plot3 := Plot[Abs[HH[Exp[-I s wg A]]], {s, 0, 2.5},
  PlotStyle → Directive[Red, Thickness[0.005]] ]
Show[{plot1a, plot3}]
```



und die Phasengänge des Analogfilters (blau) und des entworfenen diskreten Filters (rot) von 0 bis π/A ($=500\pi=2.5 \text{ wg}$ im Beispiel)


```
In[36]:= y = RecurrenceFilter[{b, a}, x]
```

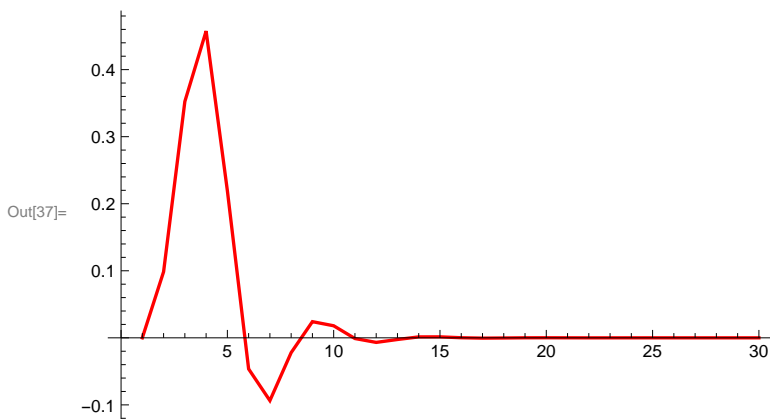
(* hier die zugehörige - um 1A verzögerte - Impulsantwort *)

```
Out[36]= {0., 0.0985312, 0.35247, 0.457494, 0.219495, -0.0464205, -0.0936204, -0.0221049,
0.0241146, 0.017973, -0.00104097, -0.00682407, -0.00248823, 0.00138339,
0.00146388, 0.000121432, -0.000469468, -0.000239802, 0.0000664282, 0.000113061,
0.0000237445, -0.0000302416, -0.0000211067, 1.90857 × 10-6, 8.30174 × 10-6,
2.79884 × 10-6, -1.77852 × 10-6, -1.73978 × 10-6, -9.655 × 10-8, 5.77959 × 10-7}
```

```
In[37]:= ListLinePlot[y, PlotRange → All, PlotStyle → Directive[Red, Thickness[0.005]]]
```

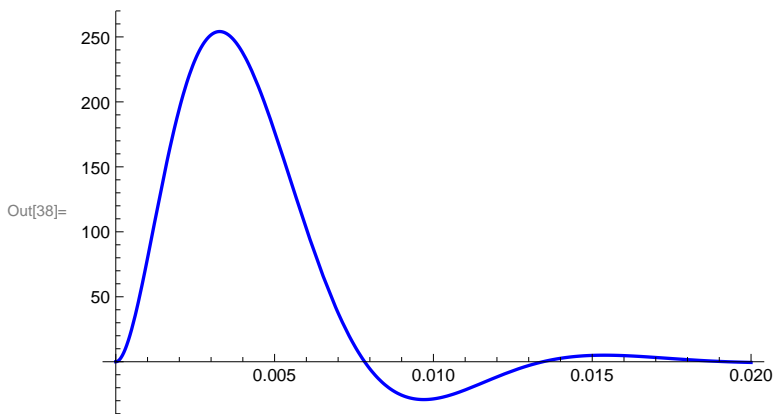
(* die Werte durch einen Polygonzug verbunden,

1 Einheit auf der horizontalen Achse entspricht der Wahl von $A=1/500$ s *)



Zum Vergleich die Impulsantwort des analogen Butterworthfilters

```
In[38]:= Plot[InverseFourierTransform[R[w], w, t, FourierParameters → {1, -1}],
{t, 0, 0.02}, PlotStyle → Directive[Blue, Thickness[0.005]]]
```



3. Vergleich mit der vorhandenen *Mathematica*-Routine

Wir lassen Mathematica zunächst mit dem **Befehl ButterworthFilterModel** ein analoges Butterworth-Tiefpassfilter der Ordnung 3 zu unserer oben gewählten Grenzfrequenz berechnen. Ich überlasse es Ihnen, zu prüfen, dass dies den gleichen analogen Frequenzgang ergibt wie bei uns oben.

Dann verwenden wir die **Mathematica-Routine ToDiscretTimeModel**, um daraus mit bilinearer Transformation ein diskretes Filter zu berechnen.

Ergebnis: Es gibt leichte Unterschiede zu unserem Filter oben in 2. Die Dämpfung des "Mathematica-Filters" an der Grenzfrequenz ist wesentlich höher als beim vorangehenden Beispiel, das Passband etwa kleiner; die Grenzfrequenz kann im Mathematica-Befehl allerdings auch nicht als Fixpunkt der Frequenzachsen-Transformation T ausgewählt werden, wie wir dies getan haben. Die Unterschiede sonst sind nicht sehr groß.

Nachfolgend die Daten und ein Vergleichsplot der beiden Amplitudengänge, der in 2. berechnete in rot, der nach dem Mathematica-Modell in blau, wieder die Grenzkreisfrequenz auf Eins normiert.

In[39]=

```
tf = N[ButterworthFilterModel[{3, wg}]]
```

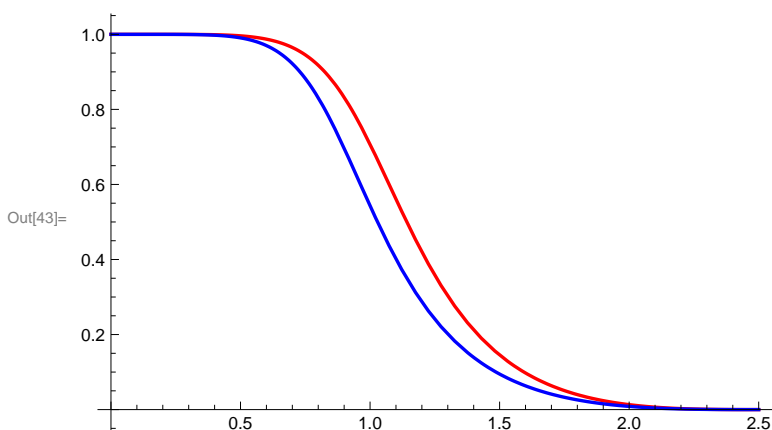
Out[39]=
$$\left(\frac{2.4805 \times 10^8}{\left((314.159 - 544.14 i) + s \right) \left((314.159 + 544.14 i) + s \right) (628.319 + s)} \right) \mathcal{T}$$

In[40]= `df = ToDiscreteTimeModel[tf, 1/500, z, Method -> {"BilinearTransform"}]`

Out[40]=
$$\left(\frac{1.9844 (1. + z)^3}{-2.27905 + (13.5836 + 4.44089 \times 10^{-16} i) z - 21.7833 z^2 + 26.354 z^3} \right) \mathcal{T}_{\frac{1}{500}}$$

```
In[41]= plot4 := Plot[Abs[df[Exp[I s wg A]]],
  {s, 0, 2.5}, PlotStyle -> Directive[Blue, Thickness[0.005]]]
-20 Log10[Abs[df[Exp[I wg A]]]] [[1, 1]]
Show[{plot3, plot4}]
```

Out[42]= 5.30263



Hier nochmal die Übertragungsfunktion des Mathematica - Modells als Funktion $dfh[z]$ von z .

```
In[44]:= dfh[z_] = Chop[df[[1, 1]][[1, 1]] / df[[1, 2]][[1, 1]] ]
Out[44]= (1.9844 (1. + z)^3) / (-2.27905 + 13.5836 z - 21.7833 z^2 + 26.354 z^3)
```

Es gibt eine ganze Reihe weiterer Methoden, aus analogen Filtern diskrete zu generieren. Nehmen Sie diese Feststellung als Anregung, hierzu mehr erfahren und lernen zu wollen, etwa durch Ausprobieren mit den fertigen *Mathematica*-Programmen, dann aber auch durch **Lektüre von Fachliteratur**. Ich gebe am Ende ein paar Referenzen zum Einstieg.

Sammeln Sie, wenn möglich, eigene Erfahrungen möglichst mit konkreten Daten aus Audio- oder Bild-Beispielen. Wenn Sie etwa auf der Seite mit der URL

<http://reference.wolfram.com/language/guide/SignalProcessing.html>
und in den *Mathematica*-Help-Pages unter tutorial/DigitalFilterDesign

starten, finden Sie schnell reichhaltiges weiteres Material.

Um dieses Notebook nicht zu lang werden zu lassen, habe ich ein Beispiel mit einem Notch-Filter, das wie hier mit bilinearer Transformation entworfen wird, und dessen Anwendung auf ein Audio-File in ein Folge-Notebook ausgelagert. Ein Sound-File zum Test ist dort auch zum Download zu finden.

Auf den Entwurf diskreter FIR-Filter gehe ich ebenfalls in einem Folge-Notebook (unter der gleichen URL wie dieses zu finden) ein.

Literatur: Neben [1] empfehle ich zur Vertiefung

- [2] A.V. Oppenheim, Zeitdiskrete Signalverarbeitung,
R.W.Schafer Oldenbourg, 1999
- [3] H.W. Schüßler Digitale Signalverarbeitung 1,
Springer, 2008
- [4] H. Wupper Einführung in die digitale Signalverarbeitung,
Hüthig, 1998