R. Brigola, TH Nürnberg Georg Simon Ohm, Februar 2015

Mathematica - Notebooks als Bonusmaterial zum Lehrbuch

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

Beispiel zum Entwurf von diskreten linearen FIR-Filtern mit linearer Phase 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-Analysis ist:
http://www.stiftung-swk.de/mathematica/

In diesem Notebook wird gezeigt, wie man **diskrete lineare Filter mit endlicher Impulsantwort** (FIR-Filter, engl. finite impulse response) entwerfen kann. *Dabei folgen wir der Beschreibung in [1], Kap.10.*

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

Diskrete FIR – Filter

Eine naheliegende Idee, ein diskretes FIR-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. In einem vorangehenden Notebook habe ich bereits die **Methode der "Bilinearen Transformation"** am Beispiel eines Butterworth-Tiefpassfilters vorgestellt.
Dabei wurde ein IIR-Filter erzeugt. Hier nun sollen sog. FIR-Filter mit *Mathematica* demonstriert werden. Sie spielen in der digitalen Übertragungstechnik eine große Rolle und müssen dort für Verfahren, die Phasenmodulationen beinhalten, einen linearen Phasengang besitzen.
Ich greife das Beispiel von S. 310 in [1] auf. Dort wird ein solches Filter durch Approximation eines idealen Tiefpassfilters mit Hilfe von Fensterfunktionen entworfen. Ich demonstriere zunächst dieses Beispiel.

Betrachtet wird stets 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 Signalräumen 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* $x = \sum_{n=-\infty}^{+\infty} x_n \delta(t\text{-}nA)$ *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.

δ (t-nA) sind **Dirac-Distributionen**, t ein Zeitparameter, A ein fester Zeitschritt, der die Abtastrate 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 $\mathbf{L: l_d^\infty \longrightarrow l_d^\infty}$ gegeben ist, d.h. es gilt:

$$\mathbf{L\ x = h * x \quad mit\ der\ Impulsanwort\ h = L\delta}$$

Approximation des Frequenzgangs eines idealen Tiefpass-Filters mit HIlfe von Fensterfunktionen

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

2. Entwurf durch Fourierreihen-Entwicklung für einen idealen Tiefpass

Unsere Grenzkreisfrequenz für das Tiefpassfilter sei ω_g. Wir betrachten dann den Ω-periodischen Frequenzgang eines idealen Tiefpasses mit Grenzkreisfrequenz ω_g, $0 < \omega_g < \Omega/2 = \pi/a$. Die Größe 1/a steht für die Abtastfrequenz, die unser diskretes System

dafür verwendet. Die Periode hängt also von der Abtastfrequenz 1/a ab, mit der das diskrete System arbeitet: Ω=2π/a. Für den Frequenzgang tp des idealen Tiefpassfilters gelte also

(1) tp[ω] = 1 für $-\omega_g \le \omega \le \omega_g$, tp[ω] = 0 für $\omega_g < |\omega| < \Omega/2$.

Wir berechnen zunächst die Übertragungsfunktion eines Näherungsfilters mit endlicher Impulsantwort durch Fourierreihenentwicklung von tp und geben konkrete Vorgaben für Ω, a und ω_g.

Wegen seiner Periodizität ist ein diskretes Tiefpassfilter nur bis zur Frequenz 1/(2a) sinnvoll nutzbar. Wir entwerfen ein kausales FIR-Tiefpassfilter mit 39 Koeffizienten und linearem Phasengang mit konstanter Gruppenlaufzeit (M-1)a.

In[1]:=
```
ClearAll["Global`*"]; Remove["Global`*"];
```

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

In[2]:=
```
a = 1 / 300     (* Abtastfrequenz 300 Hz *)
Ω = 2 π / a     (* Filterperiode in Abhängigkeit von der Abtastfrequenz *)
ωg = 2 π 100; (* Grenzkreisfrequenz 100 Hz,
Periode Ω 300 Hz, Abtastfrequenz a = 2π/Ω *)
M = 20;      (* Wir konstruierne ein FIR-Filter der Länge 2M-1= 39 *)
tp[ω_] = UnitStep[ω + ωg] - UnitStep[ω - ωg]
  (* Ideales Tiefpassfilter mit Bandbreite ωg *)
```

Out[2]= $\frac{1}{300}$

Out[3]= $600\,\pi$

Out[6]= $-\text{UnitStep}[-200\,\pi + \omega] + \text{UnitStep}[200\,\pi + \omega]$

Die gesuchten FourierKoeffizienten von tp sind:

In[7]:=
```
g[n_] = a / (2 π) ∫_(-π/a)^(π/a) tp[ω] e^(i n ω a) dω
```

Out[7]= $\frac{\text{Sin}\left[\frac{2\,n\,\pi}{3}\right]}{n\,\pi}$

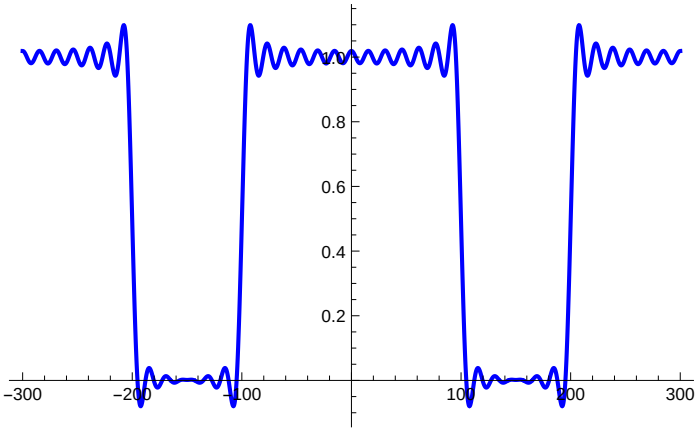
Übertragungsfunktion des nicht-kausalen FIR-Filters mit Koeffizienten g[n]

In[8]:=
```
H[z_] = Sum[Limit[g[x], x → n] z^-n, {n, 1-M, M-1}]
```

Out[8]=
$$\frac{2}{3} + \frac{\sqrt{3}}{38\,\pi\,z^{19}} - \frac{\sqrt{3}}{34\,\pi\,z^{17}} + \frac{\sqrt{3}}{32\,\pi\,z^{16}} - \frac{\sqrt{3}}{28\,\pi\,z^{14}} + \frac{\sqrt{3}}{26\,\pi\,z^{13}} - \frac{\sqrt{3}}{22\,\pi\,z^{11}} + \frac{\sqrt{3}}{20\,\pi\,z^{10}} - \frac{\sqrt{3}}{16\,\pi\,z^{8}} +$$
$$\frac{\sqrt{3}}{14\,\pi\,z^{7}} - \frac{\sqrt{3}}{10\,\pi\,z^{5}} + \frac{\sqrt{3}}{8\,\pi\,z^{4}} - \frac{\sqrt{3}}{4\,\pi\,z^{2}} + \frac{\sqrt{3}}{2\,\pi\,z} + \frac{\sqrt{3}\,z}{2\,\pi} - \frac{\sqrt{3}\,z^{2}}{4\,\pi} + \frac{\sqrt{3}\,z^{4}}{8\,\pi} - \frac{\sqrt{3}\,z^{5}}{10\,\pi} +$$
$$\frac{\sqrt{3}\,z^{7}}{14\,\pi} - \frac{\sqrt{3}\,z^{8}}{16\,\pi} + \frac{\sqrt{3}\,z^{10}}{20\,\pi} - \frac{\sqrt{3}\,z^{11}}{22\,\pi} + \frac{\sqrt{3}\,z^{13}}{26\,\pi} - \frac{\sqrt{3}\,z^{14}}{28\,\pi} + \frac{\sqrt{3}\,z^{16}}{32\,\pi} - \frac{\sqrt{3}\,z^{17}}{34\,\pi} + \frac{\sqrt{3}\,z^{19}}{38\,\pi}$$

In[9]:=
```
Plot[H[Exp[I 2 Pi f a]], {f, -300, 300}, PlotStyle → Directive[
   Blue, Thickness[0.006]], PlotRange → All] (*Frequenzgang des nicht-
 kausalen FIR-Filters in Abhängigkeit von der Frequenz in Hz *)
```

Out[9]=



Hier sehen wir also schon den Frequenzgang eines diskreten linearen Filters mit endlicher Impulsantwort der Länge 2M-1. Die Verwendung einer Partialsumme der Fourierreihenentwicklung von tp[ω] hat das **Gibbssche Phänomen** zur Folge. Außerdem ist das **Filter noch nicht kausal**.

Um zu einem kausalen Filter mit abgeschwächtem Gibbs-Phänomen zu kommen, verwendet man zur Glättung an den Sprungstellen Gewichtsfunktionen im Spektralbereich, gemeinhin **Fensterfunktionen** genannt, und erhält durch **Verzögerung**, d.h. hier Multiplikation der Übertragungsfunktion mit z^{1-M} ein kausales FIR-Filter mit einer Impulsantwort der Länge 2M-1 mit linearer Phase, hier mit konstanter Gruppenlaufzeit (M-1)a (vgl. [1], S. 309-310).

In der Praxis sind viele verschieden Fensterfunktionen - je nach Zweck - in Gebrauch (vgl. auch [1], Satz von Fejér für Fourierreihen etc.). Ich verwende zur Demonstration unten das sog. **Blackman-Fenster**.

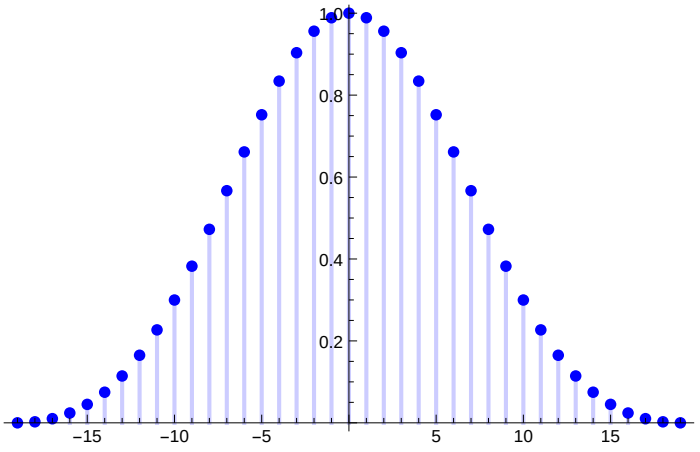
Zunächst das Blackman-Fenster, dann die Funktion G(z), in der die Übertragungsfunktion H mit dem Fenster “gewichtet” wird und schließlich die durch Verzögerung daraus entstehende Übertragungsfunktion HFIR des gewünschten kausalen FIR-Tiefpassfilters der Länge 2M-1.

Ich zeige die Übertragungsfunktion HFIR[z] , plotte die Gewichte des Blackman-Fensters, und gebe die Koeffizientenliste von HFIR und die Länge dieser Liste aus. Danach sehen wir noch den Amplitudengang über der Frequenz und noch einmal in dB für $0 \leq \omega \leq \Omega/2$.

In[10]:=
```
blackman[n_] = 0.42 + 0.5 Cos[n π / (M - 1)] + 0.08 Cos[2 n π / (M - 1)];
(* das Blackman-Fenster *)
G[z_] = Sum[blackman[n] Limit[g[x], x → n] z^-n, {n, 1 - M, M - 1}];
(* nicht-kausales FIR-Filter mit Blackman-Fenster gewichtet*)
HFIR[z_] = Together[z^(1-M) G[z]]  (* Kausalität durch Verzögerung um (M-1)a ==>
 kausales FIR-Filter mit linearer Phase*)
DiscretePlot[blackman[n], {n, 1 - M, M - 1}, PlotStyle → Directive[
   Blue, Thickness[0.006]], PlotRange → All] (* Plot Balckman-Fenster *)
coeffsFIR = CoefficientList[Numerator[HFIR[z]], z]
(* Koeffizientenliste des kausalen FIR-Filters *)
Length[coeffsFIR]
(* Länge der Koeffizientenliste = Länge der Impulsantwort *)
```

Out[12]=
$$\frac{1}{z^{38}}\left(-2.01348\times10^{-19}-0.000165765\,z^2+0.000413824\,z^3-0.00147191\,z^5+0.00242564\,z^6-0.00568619\,z^8+0.00826446\,z^9-0.0162775\,z^{11}+0.0223156\,z^{12}-0.0414618\,z^{14}+0.0574904\,z^{15}-0.131773\,z^{17}+0.27259\,z^{18}+0.666667\,z^{19}+0.27259\,z^{20}-0.131773\,z^{21}+0.0574904\,z^{23}-0.0414618\,z^{24}+0.0223156\,z^{26}-0.0162775\,z^{27}+0.00826446\,z^{29}-0.00568619\,z^{30}+0.00242564\,z^{32}-0.00147191\,z^{33}+0.000413824\,z^{35}-0.000165765\,z^{36}-2.01348\times10^{-19}\,z^{38}\right)$$

Out[13]=

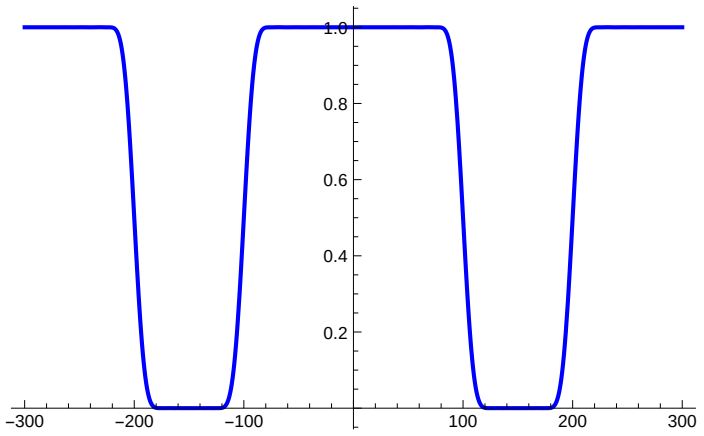


Out[14]=
$$\{-2.01348\times10^{-19},\ 0,\ -0.000165765,\ 0.000413824,\ 0,\ -0.00147191,\ 0.00242564,\ 0,\ -0.00568619,\ 0.00826446,\ 0,\ -0.0162775,\ 0.0223156,\ 0,\ -0.0414618,\ 0.0574904,\ 0,\ -0.131773,\ 0.27259,\ 0.666667,\ 0.27259,\ -0.131773,\ 0,\ 0.0574904,\ -0.0414618,\ 0,\ 0.0223156,\ -0.0162775,\ 0,\ 0.00826446,\ -0.00568619,\ 0,\ 0.00242564,\ -0.00147191,\ 0,\ 0.000413824,\ -0.000165765,\ 0,\ -2.01348\times10^{-19}\}$$

Out[15]= 39

In[16]:=
```
Plot[Abs[HFIR[Exp[I 2 Pi f a]]], {f, -Ω / (2 Pi), Ω / (2 Pi)}, PlotStyle → Directive[
   Blue, Thickness[0.006]], PlotRange → All]
(* Amplitudengang in Abhängigkeit von der Frequenz über 2 Perioden *)
```

Out[16]=

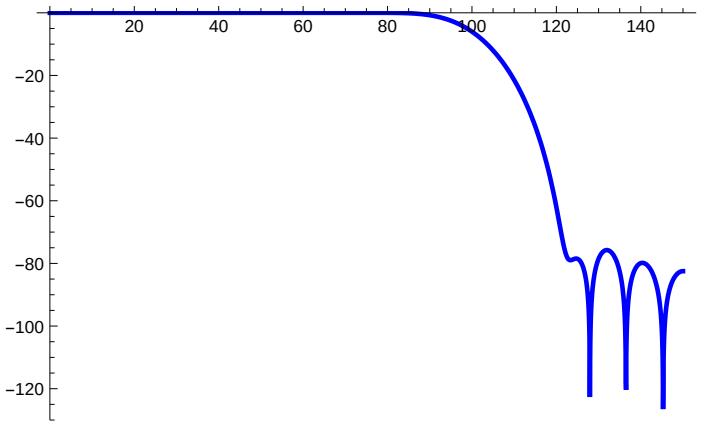


Hier nochmal der Amplitudengang in dB von 0 bis 150 Hz:

In[17]:=
```
Plot[20 Log[10, Abs[HFIR[Exp[I 2 Pi f a]]]],
 {f, 0, Ω / (4 Pi)}, PlotStyle → Directive[
   Blue, Thickness[0.007]], PlotRange → {-130, 0.1}]

(* Amplitudengang in dB als Funktion der Frequenz in Hz von 0 bis 150 Hz =
 1/2 Abtastfrequenz *)
```

Out[17]=



Dämpfung in dB bei der Grenzfrequenz:

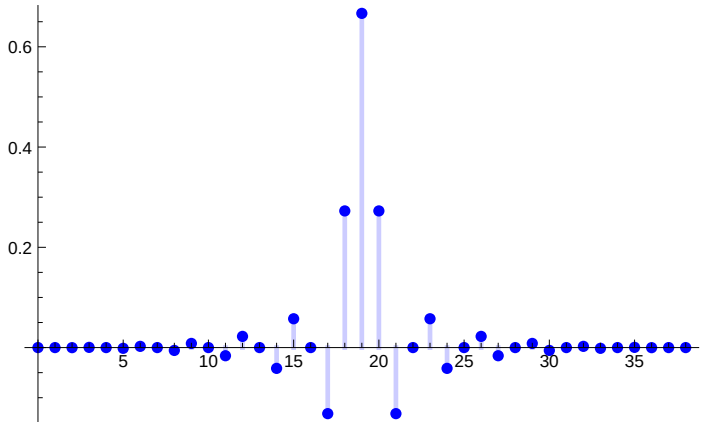
In[18]:=
```
-20 Log[10, Abs[HFIR[Exp[I 2 Pi 100 a]]]]
 (* Dämpfung in dB bei der Grenzfrequenz 100 Hz *)
```

Out[18]= 6.02054

Hier als DiscretePlot die Impulsantwort des Filters. Man erkennt deutlich die Verzögerung bei der Impulsantwort.

In[19]:=
```
plot1 = DiscretePlot[coeffsFIR[[n + 1]], {n, 0, 2 M - 2}, PlotStyle → Directive[
    Blue, Thickness[0.007]], PlotRange → All]
```

Out[19]=



Wir sehen noch die konstante Gruppenlaufzeit von (M-1)a , plotten den Phasengang und die Gruppenlaufzeit. Da *Mathematica* “keine Ableitungen der Argument-Funktion mag” (sie ist nicht analytisch), verwenden wir die **ArcTan-Funktion zur Phasenberechnung und deren Ableitung für die Gruppenlaufzeit**, dann klappt’s:

In[20]:=
```
N[(M - 1) a]  (* Gruppenlaufzeit *)
```

Out[20]=
```
0.0633333
```

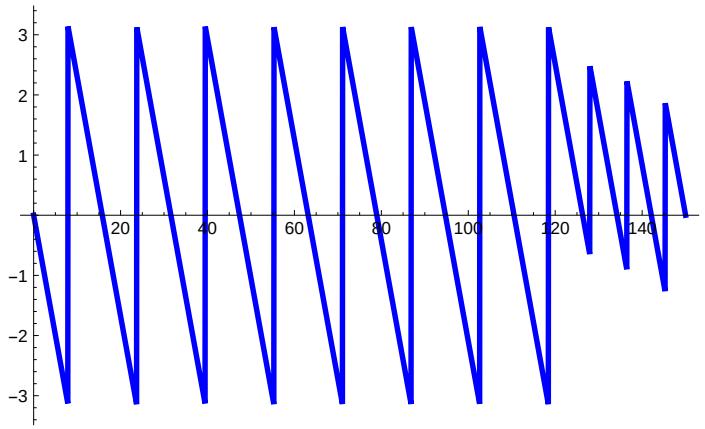
In[21]:=
```
x[ω_] = ComplexExpand[Re[HFIR[Exp[I ω a]]]];
y[ω_] = ComplexExpand[Im[HFIR[Exp[I ω a]]]];
phase[ω_] = ArcTan[x[ ω ], y[ ω ]];
groupdelay[ω_] = -phase'[ω];
groupdelay[2 Pi 50]
(* Nur als Test: Gruppenlaufzeit bei 50 Hz = (M-1)a siehe oben *)
```

Out[25]=
```
0.0633333
```

Jetzt noch zwei schöne Bilder zum Phasengang und der konstanten Gruppenlaufzeit, letztere mit Faktor 100

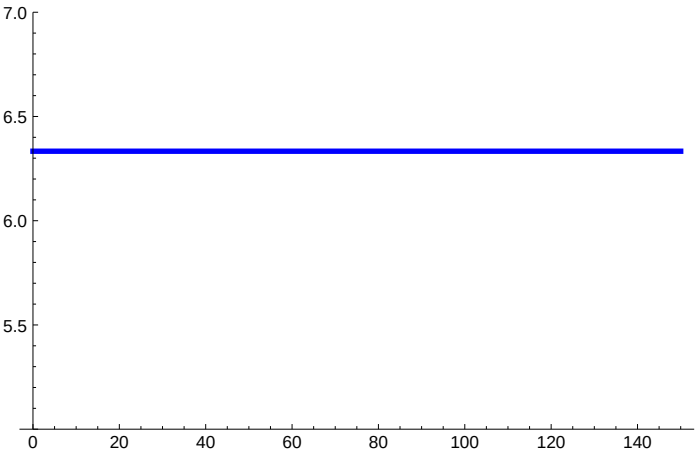
In[26]:=
```
Plot[phase[ 2 Pi f ], {f, 0, 150}, PlotStyle → Directive[
    Blue, Thickness[0.008]], PlotRange → All, ImageSize → Medium]
(* Phasengang in Abhängigkeit von der Frequenz *)
```

Out[26]=



In[27]:=
```
Plot[100 groupdelay[2 Pi f], {f, 0, 150}, PlotStyle → Directive[
    Blue, Thickness[0.008]], PlotRange → {5, 7}, ImageSize → Medium]
(* 100 × Gruppenlaufzeit 6.33333 s
  bei unserer geringen Abtastfrequenz 1/a=300 Hz *)
```

Out[27]=

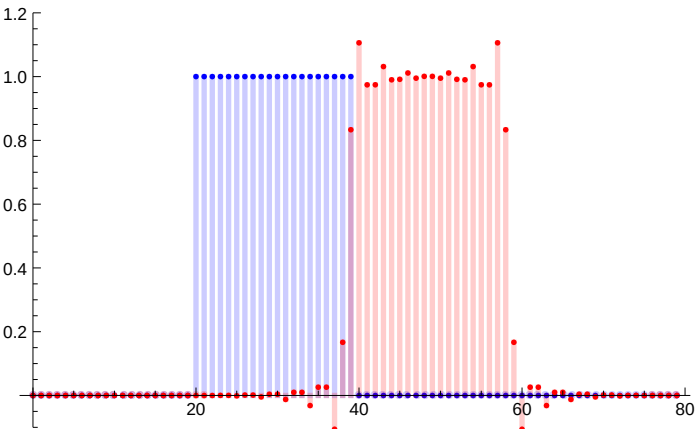


Filteranwendung auf eine diskrete Rechteckfunktion:

Ich verwende wie schon in vorherigen Notebooks einfach den *Mathematica*-Befehl RecurrenceFilter und falte damit die Impulsantwort mit den Input-Samples einer diskreten Rechteckfunktion “box”. Der Plot des Ergebnisses zeigt die Verzögerung und die Tiefpassfilterung, wodurch die senkrechten Kanten der Box durch Entfallen der hochfrequenten Anteile verschmiert werden und als Folge dann das Gibbssche Phänomen wieder zu beobachten ist (vgl. [1], Abschnitte über Fourierreihen).

In[28]:=
```
A = Table[0, {n, 1, 20}]; B = Table[1, {n, 1, 20}];
box = Flatten[Append[Append[Prepend[B, A], A], A]];
(* Definition der diskreten Box mit 20 Einsen ab Sample 21 *)
ergebnis = RecurrenceFilter[{{1}, coeffsFIR}, box];
(* Filterung mir RecurrenceFilter *)
plot1 = DiscretePlot[box[[n + 1]], {n, 0, 79}, PlotStyle → Directive[
     Blue, Thickness[0.008]], PlotRange → {-0.1, 1.2}] ;
plot2 = DiscretePlot[ergebnis[[n + 1]], {n, 0, 79}, PlotStyle → Directive[
     Red, Thickness[0.008]], PlotRange → {-0.1, 1.2}];
Show[{plot1, plot2}]
```

Out[33]=



3. In *Mathematica* bereits implementierte Routinen zum Entwurf diskreter FIR-Filter

Mathematica stellt ein ganzes Arsenal von implementierten Routinen zum Entwurf diskreter Filter bereit. Man siehe hierzu in den Help-Pages unter tutorial/DigitalFilterDesign.

Das gleiche FIR-Filter wie oben erhält man mit dem *Mathematica*-Befehl *LeastSquaresFilterKernel* und dem ebenfalls schon vorhandenen Blackman-Window.
Dabei ist zu beachten, dass - wie vielfach in der Literatur zur diskreten Signalverabeitung - mit normierten Größen a=1 und Ω=2π gerechnet wird.

Ich habe es oben vorgezogen, den direkten Zusammenhang zwischen Abtastfrequenz 1/a und Bandbreite des Filters sichtbar zu machen. Außerdem ziehe ich es vor, den Größen ihre physikalischen Einheiten zu lassen. Dann macht aber etwa für eine Größe wie ω in rad/s eine Potenzreihe wie $e^{i\omega}$ wenig Sinn, wohl aber $e^{i\omega a}$, weil dann der Exponent einheitenfrei ist. Es ist aber nicht weiter kompliziert, man muss halt die Normierung einfach berücksichtigen. Kurz gesagt: Statt oben $\omega_g = 200\,\pi$ und a=1/300 ist im Mathematica-Befehl $\omega_g a = 2\pi/3$ zu verwenden und wir erhalten das gleiche Filter. Um den Vergleich nicht mit numerischen Näherungsfehlern zu befrachten, verwende ich nachfolgend noch den Chop-Befehl. Damit werden alle Koeffizienten, die dem Betrag nach kleiner als 10^{-10} sind, Null gesetzt.

Nun also zuerst das Blackman-Fenster für die gewünschte Filterlänge, dann der **LeastSquaresFilterKernel** - wir hatten mit einer Partialsumme der Fourierreihe des idealen Tiefpasses ja eine L^2-Approximation mit minimalem quadratischen Fehler im Vergleich mit allen solchen trigonometrischen Polynomen vom Grad M-1 verwendet. Dann als Ausgabe die Filterkoeffizienten, die der *Mathematica*-Befehl errechnet. Zum Vergleich - ebenfalls mit Chop - nochmal die Koeffizienten unseres Filters von oben: Alle Koeffizienten stimmen überein, die Filter sind identisch.

In[34]:=
```
win = Array[BlackmanWindow, 2 M - 1, {-0.5, 0.5}]; (* Blackman-Fenster *)
h = LeastSquaresFilterKernel[{"Lowpass", 2 π / 3}, 2 M - 1];
(* Filterberechnung *)
h2 = Chop[win h] (* Gewichtung mit dem Blackman-
 Fenster und Ausgabe der Filterkoeffizienten*)
Chop[coeffsFIR] (* Nochmal die Filterkoeffizienten von oben mit Chop*)
```

Out[36]=
```
{0, 0, -0.000165765, 0.000413824, 0, -0.00147191, 0.00242564, 0, -0.00568619,
 0.00826446, 0, -0.0162775, 0.0223156, 0, -0.0414618, 0.0574904,
 0, -0.131773, 0.27259, 0.666667, 0.27259, -0.131773, 0, 0.0574904,
 -0.0414618, 0, 0.0223156, -0.0162775, 0, 0.00826446, -0.00568619,
 0, 0.00242564, -0.00147191, 0, 0.000413824, -0.000165765, 0, 0}
```

Out[37]=
```
{0, 0, -0.000165765, 0.000413824, 0, -0.00147191, 0.00242564, 0, -0.00568619,
 0.00826446, 0, -0.0162775, 0.0223156, 0, -0.0414618, 0.0574904,
 0, -0.131773, 0.27259, 0.666667, 0.27259, -0.131773, 0, 0.0574904,
 -0.0414618, 0, 0.0223156, -0.0162775, 0, 0.00826446, -0.00568619,
 0, 0.00242564, -0.00147191, 0, 0.000413824, -0.000165765, 0, 0}
```

Ich überlasse es interessierten Lesern, mit Literatur und den bereits vorhandenen *Mathematica*-Filtern Unterschiede verschiedener Filter-Designs zu testen und ihr Auge auf die Besonderheiten der verschiedenen Entwurfsmethoden zu richten.

Eine kleine Literatur-Auswahl zum Schluss:

Literatur: Neben [1] empfehle ich zur Vertiefung

[2] A.V. Oppenheim, R.W.Schafer — Zeitdiskrete Signalverarbeitung, Oldenbourg, 1999

[3] H.W. Schüßler — Digitale Signalverarbeitung 1, Springer, 2008

[4] H. Wupper — Einführung in die digitale Signalverarbeitung, Hüthig, 1998