Praktikum Digitale Signalverarbeitung

im WS 2006/07

Versuch: Digitale Filter

am 03.12.2006 und 17.12.2006

Gruppenmitglieder (NT8.4):

Breitschaft, Marco 742 480
Milewski, Mario 744 255
Uhl, Michael 712 560

Betreuer:

Dipl. Ing. Sörgel

11.01.2007
Inhaltsverzeichnis

4. VERSUCHSVORBEREITUNG ................................................................. 3
   4.1. MATLAB ...................................................................................... 3
      4.1.1. Allgemeines ........................................................................... 3
      4.1.2. Entwurf rekursiver Filter ....................................................... 3
      4.1.3. Entwurf nichtrekursiver Filter ............................................... 3
   4.2. PROGRAMMENTWURF ................................................................. 3
   4.3. FILTERSYNTHESE ..................................................................... 7
      4.3.1. Einfache Filter 1. Ordnung .................................................... 7
      4.3.2. Notchfilter 2. Ordnung ......................................................... 12
      4.3.3. Filteranalyse .......................................................................... 15

5. VERSUCHSDURCHFÜHRUNG ............................................................ 17
   5.1. SOFTWARE-ENTWICKLUNG ....................................................... 17
   5.2. EIGENSCHAFTEN VON FILTERN 1. ORDNUNG ......................... 17
   5.3. EIGENSCHAFTEN VON FILTERN 2. ORDNUNG ......................... 23
      5.3.1. Realisierung der Filtereinstellung nach 4.3.2 ......................... 23
      5.3.2. Online-Filterung ..................................................................... 26
   5.4. EIGENSCHAFTEN VON FILTERN HÖHERER ORDNUNG ............. 27
      5.4.1. Manueller Entwurf ................................................................. 27
      5.4.2. Entwurf rekursiver Filter mit Matlab ..................................... 29
      5.4.3. Entwurf nichtrekursiver Filter mit Matlab ......................... 53
4. Versuchsvorbereitung

4.1. Matlab

4.1.1. Allgemeines
Siehe Anhang

4.1.2. Entwurf rekursiver Filter
Siehe Anhang

4.1.3. Entwurf nichtrekursiver Filter
Siehe Anhang

4.2. Programmentwurf

function [z,p,k]=fh_showfilt(b,a,Titel,freqnorm,Nsamples,Nfft,Wu,Wo)

%SHOWFILT Zeichnet von bereits entworfenen Filtern
%  %  % Betragsfrequenzgang
%  %  % Dämpfung
%  %  % Phase
%  %  % Gruppenlaufzeit
%  %  % Pol/Nullstellendiagramm
%  %  % Impulsantwort
%  %  % Sprungantwort
%  %  % Cosinus-Antwort
%  % in insgesamt 4 plots
%  % FH_SHOWFILT([A[,Titel[, freqnorm[,Nsamples[,Nfft[,Wu[,Wo]]]]]]])
% B: Zählerpolynom summe b(i)z**(-i)
% A(optional): Nennerpolynom summe a(i)z**(-i) mit a(1)=1 !!!!
% default: [1] (nichtrekursives System)
% Titel(optional): beliebiger String, der im Titel der Plots
% erscheint
% default: Datum+ID
% freqnormoptional): Erregung mit cos(n*pi*freqnorm)
% default: 0.5
% Nsamples(optional): Berechnung der der Impuls-, Sprung-
% und Cos-antwort mit Nsamples Werten
% default: 61
% Nf(optional): Berechnung der Frequenzbereichsgrößen
% mit einer Auflösung von 2*pi/Nf (Nf Werte von 0 bis fa-)
% default: 1024
% Wu,Wo(optional): untere und obere Grenzfrequenz für
Praktikum Digitale Signalverarbeitung
Gruppe 4 im WS 2006/07

Versuch: Digitale Filter

% Zoom-Darstellungen im Frequenzbereich [\pi]
% default: Wu = 0, Wo = 1
% copyright (c) 2001 by G.Wackersreuther

if nargin<1
    error('SHOWFILT requires 1..8 arguments');
end
if ~exist('A')
    A=[1];
end
if ~exist('Nsamples')
    Nsamples = 61;
end
if ~exist('Nf')
    Nf = 1024;
end
if ~exist('freqnorm')
    freqnorm = 0.5;
end
if ~exist('Wu')
    Wu = 0;
end
if ~exist('Wo')
    Wo = 1;
end
if ~exist('Titel')
    Titel = sprintf('%s: ID %f ',date,rand);
end

%-----------------------------------------------------------------------
% Berechnungen Frequenzbereich
% Berechnen der Nullstellen mit roots():
z = roots(b)

% Berechnen der Pole:
p = roots(a)
k=b(1);

% Zur Kontrolle:
% a=real(poly(p));
% b=real(poly(z)*k);
w = [Wu:2/Nf:Wo];

% Komplexer Frequenzgang:
h = freqz(b,a,w*pi); % Berechnen der Z-Transformierten Frequenzantwort

% Amplitudengang:
hbetrag = abs(h); % Absolutwert von h

% Daempfungsgang:
att=-20*log10(max(hbetrag,max(hbetrag)*1e-5)); % Dämpfungspole begrenzen;

% Phasengang [\pi]
phase = unwrap(angle(h));

% Gruppenlaufzeit direkt berechnet:
Praktikum Digitale Signalverarbeitung
Gruppe 4 im WS 2006/07
Versuch: Digitale Filter

```
% alternativ mit Ableiten der Phase:
gda = [-diff(phase)/(2/Nf) 0];

%Textausgaben:
Zahlerpolynom=b
Nennerpolynom=a
Nullstellen=z
Pole=p
Konstante=k

%Berechnungen Zeitbereich
n=[0:Nsamples];

%Einheitsimpuls:
impuls=[1,zeros(1,Nsamples)];

%Einheitssprung:
sprung = ones(1,(Nsamples + 1)); %zweidimensionales Array!!!
%ones(1,x) "1" ist die Dimensionsangabe!!!

%Cosinus:
cosinus = cos(n*pi*freqnorm);

%Impulsantwort:
yimpuls = filter(b,a,impuls);

%Sprungantwort:
ysprung = filter(b,a,sprung);

%Cosinusantwort:
ycosinus = filter(b,a,cosinus);

```

```matlab
figure
subplot(3,1,1);
stem(n,impuls,:);
hold on;
stem(n,yimpuls);
set(gca,'FontSize',10);
title(sprintf('%s: Impulsantwort',Titel));
xlabel('n');
ylabel('g(n)');
hold off;
subplot(3,1,2);
stem(n,sprung,:);
hold on;
stem(n,ysprung);
set(gca,'FontSize',10);
title('Sprungantwort');
xlabel('n');
ylabel('g_s(n)');
hold off;
subplot(3,1,3);
stem(n,cosinus,:);
```

Breitschaft, Marco
Milewski, Mario
Uhl, Michael
11.01.2007
Seite 5 von 70
Praktikum Digitale Signalverarbeitung
Gruppe 4 im WS 2006/07

Versuch: Digitale Filter

```matlab
%hold on;
stem(n,ycosinus);
set(gca,'FontSize',10);
title(sprintf('Cos-Antwort, freqnorm= %f.3 * pi',freqnorm));
xlabel('n');
ylabel('gc(n)');
%hold off;

%Ausgabe Amplituden- und Daempfungsgang
figure
subplot(2,1,1);
plot(w,hbetrag);
set(gca,'FontSize',10);
title(sprintf('%s: Betragsfrequenzgang',Titel));
xlabel('omega[pi]');
ylabel('|H(omega)|');
subplot(2,1,2);
plot(w,att);
set(gca,'FontSize',10);
title(sprintf('%s: Daempfungsgang',Titel));
xlabel('omega[pi]');
ylabel('att(omega)');

%Ausgabe Phase und Gruppenlaufzeit
figure
subplot(2,1,1);
plot(w,phase);
set(gca,'FontSize',10);
title(sprintf('%s: Phasengang [pi]',Titel));
xlabel('omega[pi]');
ylabel('b(omega)');
subplot(2,1,2);
plot(w,gd);
set(gca,'FontSize',10);
title(sprintf('%s: Gruppenlaufzeit direkt',Titel));
xlabel('omega[pi]');
ylabel('d(omega)');

%Ausgabe Phase und Gruppenlaufzeit mit Ableiten der Phase
figure
subplot(2,1,1);
plot(w,phase);
set(gca,'FontSize',10);
title(sprintf('%s: Phasengang [pi]',Titel));
xlabel('omega[pi]');
ylabel('b(omega)');
subplot(2,1,2);
plot(w,gda);
set(gca,'FontSize',10);
title(sprintf('%s: Gruppenlaufzeit -dphase/domega',Titel));
xlabel('omega[pi]');
ylabel('d(omega)');

%Ausgabe PN-Diagramm
figure
zplane(z,p);
title(sprintf('%s: PN-Diagramm',Titel));
%end;
```
4.3. Filtersynthese

4.3.1. Einfache Filter 1. Ordnung

Gleichung (7) \[ G(z) = z^{M-N} \sum_{k=0}^{N} b_k \cdot z^{N-k} \]
\[ + \sum_{k=0}^{M} a_k \cdot z^{M-k} \]

Filter 1. Ordnung: \[ G(z) = 1 \cdot \frac{\sum_{k=0}^{1} b_k \cdot z^{1-k}}{\sum_{k=0}^{1} a_k \cdot z^{1-k}} = \frac{b_0 \cdot z + b_1}{a_0 \cdot z + a_1} \]

mit \( b_0 = 1 = a_0 \): \[ G(z) = \frac{z + b_1}{z + a_1} = \frac{z - \beta}{z - \alpha} \]
Berechnung der Koeffizienten des linken Filters:

a) $\beta = 0; \quad \alpha = 0.9; \quad \rightarrow G(z) = \frac{1}{1 - 0.9 \cdot z^{-1}}$

\[B = \begin{bmatrix} 1 & 0 \end{bmatrix};\]
\[A = \begin{bmatrix} 1 & -0.9 \end{bmatrix};\]
\[\text{showfilter}(B, A, 'Vorbereitung 4.3.1 -1a', 0.25, 40)\]

b) $\beta = 0; \quad \alpha = 1.0; \quad \rightarrow G(z) = \frac{1}{1 - 1.0 \cdot z^{-1}}$

\[B = \begin{bmatrix} 1 & 0 \end{bmatrix};\]
\[A = \begin{bmatrix} 1 & -1 \end{bmatrix};\]
\[\text{showfilter}(B, A, 'Vorbereitung 4.3.1 -1b', 0.25, 40)\]
Praktikum Digitale Signalverarbeitung
Gruppe 4 im WS 2006/07
Versuch: Digitale Filter

c) $\beta = 0; \quad \alpha = 1,1; \quad \rightarrow \quad G(z) = \frac{1}{1-1,1 \cdot z^{-1}}$

\begin{align*}
B &= [ 1 \ 0 ]; \\
A &= [ 1 \ -1.1 ]; \\
showfilter(B,A, 'Vorbereitung 4.3.1 -1c',0.25,40)
\end{align*}
Berechnung der Koeffizienten des rechten Filters:

a) $\beta = 0.9; \quad \alpha = 0; \quad \rightarrow \quad G(z) = \frac{1 - 0.9 \cdot z^{-1}}{1}$

$$B = \begin{bmatrix} 1 & -0.9 \end{bmatrix};$$

$$A = \begin{bmatrix} 1 & 0 \end{bmatrix};$$

showfilter($B,A$, 'Vorbereitung 4.3.1 -2a', 0.25, 40)


b) $\beta = 1.0; \quad \alpha = 0; \quad \rightarrow \quad G(z) = \frac{1 - 1.0 \cdot z^{-1}}{1}$

$$B = \begin{bmatrix} 1 & -1 \end{bmatrix};$$

$$A = \begin{bmatrix} 1 & 0 \end{bmatrix};$$

showfilter($B,A$, 'Vorbereitung 4.3.1 -2b', 0.25, 40)
Praktikum Digitale Signalverarbeitung
Gruppe 4 im WS 2006/07
Versuch: Digitale Filter

c) $\beta = 1.1; \quad \alpha = 0; \quad \rightarrow \quad G(z) = \frac{1 - 1.1 \cdot z^{-1}}{1}$

$B = [1 \quad -1.1];$
$A = [1 \quad 0];$
showfilter(B, A, 'Vorbereitung 4.3.1 -2c', 0.25, 40)
4.3.2. Notchfilter 2. Ordnung

Ein Notchfilter ist eine Bandsperre mit sehr schmalem Sperrbereich.

System 2.Ordnung:  

\[ G(z) = \frac{b_0 \cdot z^{-2} + b_1 \cdot z + b_2}{a_0 \cdot z^{-2} + a_1 \cdot z + a_2} \]

Nullstellen liegen bei \( \frac{\omega_n}{8} \) direkt auf dem Einheitskreis:  

\[ \omega_n = \frac{360^\circ}{8} = 45^\circ \]

\[ \beta_{1,2} = 1 \cdot e^{\pm \frac{2\pi}{8}} = 1 \cdot e^{\pm \frac{\pi}{4}} = \sqrt{\frac{1}{2}} \pm j \sqrt{\frac{1}{2}} = 0,707 \pm j 0,707 \]

\[ \alpha_{1,2} = r \cdot e^{\pm \frac{2\pi}{8}} = r \cdot e^{\pm \frac{\pi}{4}} = r \cdot \sqrt{\frac{1}{2}} \pm j \cdot r \cdot \sqrt{\frac{1}{2}} = r \cdot (0,707 \pm j 0,707) \]

Daraus folgt nach Gleichung (15):

\[ a_0 = 1 \]

\[ a_1 = -(\alpha_1 + \alpha_2) = -r \cdot (e^{\frac{\pi}{4}} + e^{-\frac{\pi}{4}}) = -r \cdot \left( e^{\frac{\pi}{4}} + e^{-\frac{\pi}{4}} \right) \]

\[ a_2 = \alpha_1 \cdot \alpha_2 = r \cdot e^{\frac{\pi}{4}} \cdot r \cdot e^{-\frac{\pi}{4}} = r^2 \]

Aus Gleichung (16) folgt:

\[ b_0 = 1 \]

\[ \frac{b_1}{b_0} = -(\beta_1 + \beta_2) = -\left( e^{\frac{\pi}{4}} + e^{-\frac{\pi}{4}} \right) = -\sqrt{2} \]

\[ b_1 = -\sqrt{2} \cdot b_0 \]

\[ b_0 = -\frac{b_1}{\sqrt{2}} \]
Praktikum Digitale Signalverarbeitung
Gruppe 4 im WS 2006/07

Versuch: Digitale Filter

\[
\frac{b_2}{b_0} = \beta_1 \cdot \beta_2 = e^{\frac{\pi}{4}} \cdot e^{-\frac{\pi}{4}} = e^0 = 1
\]

\[b_2 = b_0\]

a) \( r = 0.5 \quad \Rightarrow \quad G(z) = \frac{b_0 \cdot z^2 + b_1 \cdot z + b_2}{a_0 \cdot z^2 + a_1 \cdot z + a_2} = \frac{z^2 - 1.414 \cdot z + 1}{z^2 - 0.707z + 0.25}\]

\[
B = \begin{bmatrix} 1 & -1.414 & 1 \end{bmatrix};
A = \begin{bmatrix} 1 & -0.707 & 0.25 \end{bmatrix};
\]

\text{showfilter}(B,A, 'Vorbereitung 4.3.2 -a',0.25,40)
b) $r = 0.9 \quad \Rightarrow \quad G(z) = \frac{z^2 - 1.414 \cdot z + 1}{z^2 - 1.273z + 0.81}$

$B = \begin{bmatrix} 1 & -1.414 & 1 \end{bmatrix}$;

$A = \begin{bmatrix} 1 & -1.273 & 0.81 \end{bmatrix}$;

`showfilter(B,A, 'Vorbereitung 4.3.2 -b',0.25,40)`

---

c) $r = 0.99 \quad \Rightarrow \quad G(z) = \frac{z^2 - 1.414 \cdot z + 1}{z^2 - 1.4z + 0.98}$

$B = \begin{bmatrix} 1 & -1.414 & 1 \end{bmatrix}$;

$A = \begin{bmatrix} 1 & -1.4 & 0.98 \end{bmatrix}$;

`showfilter(B,A, 'Vorbereitung 4.3.2 -c',0.25,40)`
4.3.3. Filteranalyse

Gegeben ist ein rekursives Filter 2. Ordnung \( G(z) = \frac{b_0 \cdot z^2 + b_1 \cdot z + b_2}{a_0 \cdot z^2 + a_1 \cdot z + a_2} \) mit folgenden Koeffizienten:

\[ b_0 = 1; b_1 = 0; b_2 = 0; a_0 = 1; a_1 = 1,3435; a_2 = 0,855 \]

\[ G(z) = \frac{z^2}{z^2 + 1,3435 \cdot z + 0,855} \]

Nullstellen:

\[ \text{solve ('z^2=0')} \]
\[ \text{ans} = 0 
\]

Polstellen:

\[ \text{solve ('z^2+1.3435*z+0.855=0')} \]
\[ \text{ans} = -0.67175000000000000000000000000000+.63541477595347119847102443729470*i 
\]
\[ -0.67175000000000000000000000000000-.63541477595347119847102443729470*i \]
B = \[ 1 \ 0 \ 0 \];
A = \[ 1 \ 1.3435 \ 0.855 \];
showfilter(B,A, 'Vorbereitung 4.3.3',0.25,40)
5. Versuchsdurchführung

5.1. Software-Entwicklung

5.2. Eigenschaften von Filtern 1. Ordnung

Linker Filter:

a) $\beta = 0; \quad \alpha = 0.9; \quad \rightarrow \quad G(z) = \frac{1}{1 - 0.9 \cdot z^{-1}}$

Eigenschaften:

- Stabil
- Tiefpass mit flacher Flanke im Betragsfrequenzgang und einen Phasengang ohne Sprungstellen
- Rekursiv
b) \( \beta = 0; \quad \alpha = 1,0; \quad \Rightarrow \quad G(z) = \frac{1}{1 - 1,0 \cdot z^{-1}} \)

Eigenschaften:
- Grenzstabil
- Tiefpass mit steiler Flanke im Betragsfrequenzgang und einen linear ansteigenden Phasengang
- Rekursiv
c) \( \beta = 0; \quad \alpha = 1,1; \quad \rightarrow \quad G(z) = \frac{1}{1 - 1,1 \cdot z^{-1}} \)

Eigenschaften:

- Instabil
- Tiefpass mit flacher Flanke im Betragsfrequenzgang
- Rekursiv
- Die Cosinus-Antwort weist kein Cosinus-Verhalten mehr auf, sondern ein langsam ansteigendes Verhalten
Praktikum Digitale Signalverarbeitung
Gruppe 4 im WS 2006/07
Versuch: Digitale Filter

Rechter Filter:

\[ G(z) = \frac{1 - 0.9 \cdot z^{-1}}{1} \]

- Stabil
- Hochpass mit flacher Flanke
- Nichtrekursiv
- Minimalphasig da die Nullstelle innerhalb des Einheitskreises.
b) \( \beta = 1.0; \quad \alpha = 0; \quad \rightarrow \quad G(z) = \frac{1 - 1.0 \cdot z^{-1}}{1} \)

**Eigenschaften:**

- Stabil
- Hochpass mit flacher Flanke
- Nichtrekursiv
- Minimalphasig da die Nullstelle auf dem Einheitskreis liegt
c) $\beta = 1.1; \quad \alpha = 0; \quad \rightarrow \quad G(z) = \frac{1 - 1.1 \cdot z^{-1}}{1}$

Eigenschaften:

- Stabil
- Hochpass mit flacher Flanke
- Nichtrekursiv
5.3. Eigenschaften von Filtern 2. Ordnung

5.3.1. Realisierung der Filtereinstellung nach 4.3.2.

a) \( r = 0.5 \quad \Rightarrow \quad G(z) = \frac{b_0 \cdot z^2 + b_1 \cdot z + b_2}{a_0 \cdot z^2 + a_1 \cdot z + a_2} = \frac{z^2 - 1.414 \cdot z + 1}{z^2 - 0.707z + 0.25} \)

Eigenschaften:

- Stabil mit einer kurzen Einschwingzeit
- Schmalbandiger Filter: sperrt die Frequenz \( = \frac{f_a}{4} \)
- Minimalphasig da die Nullstellen auf dem Einheitskreis liegen.
Praktikum Digitale Signalverarbeitung
Gruppe 4 im WS 2006/07
Versuch: Digitale Filter

b) \( r = 0.9 \Rightarrow G(z) = \frac{z^2 - 1.414 \cdot z + 1}{z^2 - 1.273z + 0.81} \)

Eigenschaften:
- Stabil mit einer langen Einschwingzeit
- Schmalbandiger Filter: sperrt die Frequenz = \( f_0/4 \)
- Minimalphasig da die Nullstellen auf dem Einheitskreis liegen.
c) \( r = 0.99 \implies G(z) = \frac{z^2 - 1.414 \cdot z + 1}{z^2 - 1.4z + 0.98} \)

Eigenschaften:

- Stabil
- Schmalbandiger Filter: sperrt die Frequenz \( f_0/4 \)
- Minimalphasig da die Nullstellen auf dem Einheitskreis liegen.
5.3.2. Online-Filterung

$U = 1,7 \, V$

Aus Matlab-Plot: $f_0 = 0,25 \rightarrow$ Messung 2,76kHz

Aus Matlab-Plot $f_1 = 0,2 \times \pi$ (entspr. 0,848) $\rightarrow$ 2,208kHz (enspr. $\hat{U} = 1,423V$: $1,423/1,7 = 0,837 \rightarrow$ OK)

Aus Matlab-Plot $f_2 = 0,3 \times \pi$ (entspr. 0,846) $\rightarrow$ 3,312 kHz (enspr. $\hat{U} = 1,383V$: $1,383/1,7 = 0,814 \rightarrow$ OK)
5.4. Eigenschaften von Filtern höherer Ordnung

5.4.1. Manueller Entwurf

Toleranzschema für Tiefpaß:

\[
0 \leq f \leq 0.1 \cdot f_a : \quad 0.9 \leq |G(f)| \leq 1.0
\]

\[
0.15 \cdot f_a \leq f \leq 0.5 \cdot f_a : \quad |G(f)| \leq 0.1
\]

\[
A = \begin{bmatrix}
1 & -3.0827 & 3.7502 & -2.1022 & 0.45682 \\
\end{bmatrix};
\]

\[
B = \begin{bmatrix}
0.2128*0.1007 & -0.24952*0.1007 & 0.29392*0.1007 & -0.25522*0.1007 & 0.21778*0.1007 \\
\end{bmatrix};
\]

\[
\text{showfilter}(B,A,'\text{Durchführung 5.4.1 - TP'},0.25,40);
\]
Toleranzschema für Hochpaß:

\[
0 \leq f \leq 0.25 \cdot f_u : \quad |G(f)| \leq 0.1
\]

\[
0.3 \cdot f_u \leq f \leq 0.5 \cdot f_u : \quad 0.9 \leq |G(f)| \leq 1.0
\]

\[
A = [1 \ 0.29084 \ 0.95917 \ 0.22258 \ 0.13296];
B = [0.1965 -0.38577 0.44829 -0.34461 0.10694];
showfilter(B,A, 'Durchführung 5.4.1 - HP',0.25,40);
5.4.2. Entwurf rekursiver Filter mit Matlab

5.4.2.1. Butterworth

Tiefpaß:

Matlab-Coefficient File (butterworthTP.fcf):

```matlab
% % Generated by MATLAB(R) 7.0 and the Signal Processing Toolbox 6.2.
% % Generated on: 04-Dec-2006 19:14:05
% Discrete-Time IIR Filter (real)

% Filter Structure : Direct-Form II, Second-Order Sections
Filter Order : 8
Stable : Yes
Linear Phase : No

SOS matrix:
1 2 1 1 -1.781987663565043 0.881939514863889
1 2 1 1 -1.606735168718103 0.696857091525947
1 2 1 1 -1.494260315541172 0.578073513530400
```
Praktikum Digitale Signalverarbeitung

Gruppe 4 im WS 2006/07

Versuch: Digitale Filter

1 2 1 1 -1.439716831516752 0.520470680490273
Scale Factors:
0.024987962824712
0.022530480701961
0.020953299497307
0.020188462243380
1.000000000000000

Matlab-M-File (butterworthTP.m):

function Hd = butterworthTP
% BUTTERWORTHTP Returns a discrete-time filter object
% M-File generated by MATLAB(R) 7.0 and the Signal Processing Toolbox 6.2.
% Generated on: 04-Dec-2006 19:04:18
% Butterworth Lowpass filter designed using FDESIGN.LOWPASS.
% All frequency values are normalized to 1.
Fpass = 0.1;  % Passband Frequency
Fstop = 0.15; % Stopband Frequency
Apass = 1.93; % Passband Ripple (dB)
Astop = 26;  % Stopband Attenuation (dB)
match = 'stopband'; % Band to match exactly
% Construct an FDESIGN object and call its BUTTER method.
h = fdesign.lowpass(Fpass, Fstop, Apass, Astop);
Hd = butter(h, 'MatchExactly', match);

Filter dargestellt mit showfilter.m:

Wpass = 0.1;Wstop = 0.15;Apass = 1.93;Astop = 26;
[N, Wn] = BUTTORD(Wpass, Wstop, Apass, Astop);
[B, A] = BUTTER(N,Wn,'low');
showfilter(B,A, 'Durchfuehrung 5.4.2 - Butter-TP',0.25,40);
Besser mit N=80 Abtastwerten:
Hochpaß:

Matlab-Coefficient File (butterworthHP.fcf):

```matlab
% Generated by MATLAB(R) 7.0 and the Signal Processing Toolbox 6.2.
% Generated on: 04-Dec-2006 20:51:56
% Discrete-Time IIR Filter (real)
% Filter Structure : Direct-Form II, Second-Order Sections
% Filter Order      : 16
% Stable            : Yes
% Linear Phase      : No

SOS matrix:
1 -2 1 1 -1.111846477064217  0.854516983361015
1 -2 1 1 -0.973010012386824  0.622943122252171
1 -2 1 1 -0.870604984989959  0.452135491516595
1 -2 1 1 -0.795277463144658  0.326492094286546
1 -2 1 1 -0.740770167528025  0.235576030312467
1 -2 1 1 -0.702917682654999  0.172439547436883
1 -2 1 1 -0.679017959846558  0.132575732818425
1 -2 1 1 -0.667448784410324  0.113278795001406
Scale Factors:
0.741590865106308
0.648988283659748
```
Praktikum Digitale Signalverarbeitung
Gruppe 4 im WS 2006/07

Versuch: Digitale Filter

11.01.2007  Seite 33 von 70

1.000000000000000
0.445181894852933
0.452898423166246
0.445181894852933
0.494086549460123
0.530442389357801
0.580685119126638

Matlab-M-File (butterworthHP.m):

function Hd = butterworthHP
% BUTTERWORTHHP Returns a discrete-time filter object
%
% M-File generated by MATLAB(R) 7.0 and the Signal Processing Toolbox 6.2.
% Generated on: 04-Dec-2006 20:05:55
%
% Butterworth Highpass filter designed using FDESIGN.HIGHPASS.
% All frequency values are normalized to 1.
Fstop = 0.25;        % Stopband Frequency
Fpass = 0.3;         % Passband Frequency
Astop = 26;          % Stopband Attenuation (dB)
Apass = 1.93;        % Passband Ripple (dB)
match = 'passband';  % Band to match exactly
%
% Construct an FDESIGN object and call its BUTTER method.
h  = fdesign.highpass(Fstop, Fpass, Astop, Apass);
Hd = butter(h, 'MatchExactly', match);

% [EOF]

Filter dargestellt mit showfilter.m:

Wpass = 0.3;Wstop = 0.25;Apass = 1.93;Astop = 26;
[N, Wn] = BUTTORD(Wpass, Wstop, Apass, Astop);
[B, A] = BUTTER(N,Wn,'high');
showfilter(B, A, 'Durchfuhrung 5.4.2 - Butter-HP',0.25,40);
Besser mit N=80 Abtastwerten:

Das Butterworth-Filter hat im Vergleich zu den anderen Filtern einen flacheren Frequenzgang. Die Pole liegen auf einer elliptischen Bahn innerhalb des Einheitskreises (→ stabil)
5.4.2.2. Chebyshev 1

Tiefpaß:

Matlab-Coefficient File (chebyshev1TP.fcf):

```
% % Generated by MATLAB(R) 7.0 and the Signal Processing Toolbox 6.2.
% % Generated on: 04-Dec-2006 21:15:49
% % Discrete-Time IIR Filter (real)

Filter Structure : Direct-Form II, Second-Order Sections
Filter Order : 5
Stable : Yes
Linear Phase : No

SOS matrix:
1 2 1 1 -1.866924945670918 0.958449538974630
1 2 1 1 -1.856243682771697 0.893373751260179
1 1 0 1 -0.932088013807489 0.000000000000000
Scale Factors:
0.022881148325928
0.009282517122121
```

Breitschaft, Marco Milewski, Mario Uhl, Michael

11.01.2007 Seite 35 von 70
Matlab-M-File (chebyshev1TP.m):

```
function Hd = chebyshev1TP
%CHEBYSHEV1TP Returns a discrete-time filter object

% M-File generated by MATLAB(R) 7.0 and the Signal Processing Toolbox 6.2.
% Generated on: 04-Dec-2006 21:15:00

% Chebyshev Type I Lowpass filter designed using FDESIGN.LOWPASS.
% All frequency values are normalized to 1.

Fpass = 0.1;         % Passband Frequency
Fstop = 0.15;        % Stopband Frequency
Apass = 1.93;        % Passband Ripple (dB)
Astop = 26;          % Stopband Attenuation (dB)
match = 'passband';  % Band to match exactly

% Construct an FDESIGN object and call its CHEBY1 method.
Hd = cheby1(h, 'MatchExactly', match);

% [EOF]
```

Filter dargestellt mit showfilter.m:

```
Wpass = 0.1;Wstop = 0.15;Apas = 1.93;Astop = 26;
[N, Wn] = cheb1ord(Wpass, Wstop, Apas, Astop);
[B, A] = cheby1(N, Apas, Wn, 'low');
showfilter(B, A, 'Durchfuehrung 5.4.2 - Cheby-TP',0.25,40);
```
Versuch: Digitale Filter

Besser mit N=180 Abtastwerten:
Hochpaß:

Matlab-Coefficient File (chebyshev1HP.fcf):

```matlab

% Generated by MATLAB(R) 7.0 and the Signal Processing Toolbox 6.2.
% Generated on: 04-Dec-2006 21:37:48
% Discrete-Time IIR Filter (real)
% -------------------------------
% Filter Structure : Direct-Form II, Second-Order Sections
% Filter Order : 6
% Stable : Yes
% Linear Phase : No

SOS matrix:
1  -2  1  1  -1.109416273135593  0.923693830018257
1  -2  1  1  -0.592321150719529  0.713065631111627
1  -2  1  1   0.584973144178701  0.330202698711277
Scale Factors:
0.758277525788463
0.576346695457789
0.18630738633144
0.800755628506701
```

Breitschaft, Marco  Milewski, Mario  Uhl, Michael

11.01.2007  Seite 38 von 70
Matlab-M-File (chebyshev1HP.m):

```matlab
function Hd = chebyshev1HP
%CHEBYSHEV1HP Returns a discrete-time filter object
% M-File generated by MATLAB(R) 7.0 and the Signal Processing Toolbox 6.2.
% Generated on: 04-Dec-2006 21:37:03
% Chebyshev Type I Highpass filter designed using FDESIGN.HIGHPASS.
% All frequency values are normalized to 1.

% Stopband Frequency
Fstop = 0.25;
% Pasband Frequency
Fpass = 0.3;
% Stopband Attenuation (dB)
Astop = 26;
% Passband Ripple (dB)
Apass = 1.93;
% Band to match exactly
match = 'passband';

% Construct an FDESIGN object and call its CHEBY1 method.
h  = fdesign.highpass(Fstop, Fpass, Astop, Apass);
Hd = cheby1(h, 'MatchExactly', match);

% [EOF]
```

Filter dargestellt mit showfilter.m:

```matlab
% Stopband Frequency
Wstop = 0.25; Apass = 1.93; Astop = 26;
[N, Wn] = cheb1ord(Wpass, Wstop, Apass, Astop);
[B, A] = cheby1(N, Apass, Wn, 'high');
showfilter(B, A, 'Durchfuehrung 5.4.2 - Cheby-HP',0.25,40);
```
Das Chebychev-Filter 1 benötigt für das Toleranzschema nur noch eine Filterordnung von 3. Die Pole liegen innerhalb des Einheitskreises (-> stabil)
5.4.2.3. Chebyshev 2

Tiefpaß:

Matlab-Coefficient File (chebyshev2TP.fcf):

```matlab
% Generated by MATLAB(R) 7.0 and the Signal Processing Toolbox 6.2.
% Generated on: 04-Dec-2006 22:28:04
% Discrete-Time IIR Filter (real)

Filter Structure : Direct-Form II, Second-Order Sections
Filter Order : 5
Stable : Yes
Linear Phase : No

SOS matrix:
1  -1.760378187403535  1  1  -1.728669630430773  0.861860179676496
1  -1.428096060997887  1  1  -1.372669129559438  0.541518958378701
1   1.000000000000000  0  1  -0.540896701818420  0.000000000000000
Scale Factors: 0.555836498365956 0.295241590945991
```

Breitschaft, Marco

Milewski, Mario

Uhl, Michael

11.01.2007
Matlab-M-File (chebyshev2TP.m):

```matlab
function Hd = chebyshev2TP
    %CHEBYSHEV2TP Returns a discrete-time filter object
    %
    % M-File generated by MATLAB(R) 7.0 and the Signal Processing Toolbox 6.2.
    % Generated on: 04-Dec-2006 22:28:16
    %
    % Chebyshev Type II Lowpass filter designed using FDESIGN.LOWPASS.
    % All frequency values are normalized to 1.
    Fpass = 0.1;         % Passband Frequency
    Fstop = 0.15;        % Stopband Frequency
    Apass = 1.93;        % Passband Ripple (dB)
    Astop = 26;          % Stopband Attenuation (dB)
    match = 'stopband';  % Band to match exactly
    % Construct an FDESIGN object and call its CHEBY2 method.
    h  = fdesign.lowpass(Fpass, Fstop, Apass, Astop);
    Hd = cheby2(h, 'MatchExactly', match);
    % [EOF]
```

Filter dargestellt mit showfilter.m:

```matlab
Wpass = 0.1;Wstop = 0.15;Apass = 1.93;Astop = 26;
[N, Wn] = cheb2ord(Wpass, Wstop, Apass, Astop);
[B, A] = cheby2(N, Astop, Wn, 'low');
showfilter(B,A, 'Durchfuehrung 5.4.2 – Cheby2-TP',0.25,40);
```
Besser mit N=80 Abtastwerten:
Hochpaß:

Matlab-Coefficient File (chebyshev2HP.fcf):

```matlab
% % Generated by MATLAB(R) 7.0 and the Signal Processing Toolbox 6.2. % % Generated on: 04-Dec-2006 22:44:27 % Discrete-Time IIR Filter (real) ----------------------------- Filter Structure : Direct-Form II, Second-Order Sections Filter Order : 6 Stable : Yes Linear Phase : No

SOS matrix:
1 -1.448039041497417 1 1 -1.113403334268761 0.794895596798576 0.843464617443635 0.702600247212075 0.627423700267539 1.000000000000000
1 -1.683965705762913 1 1 -1.087065803697065 0.501289411892763 0.702600247212075 0.627423700267539 0.346957524300378 1.000000000000000
1 -1.954549557806759 1 1 -1.134220592150099 0.346957524300378 0.843464617443635 0.702600247212075 0.627423700267539 1.000000000000000
```

Breitschaft, Marco  Milewski, Mario  Uhl, Michael

11.01.2007  Seite 44 von 70
Matlab-M-File (chebyshev2HP.m):

function Hd = chebyshev2HP
%CHEBYSHEV2HP Returns a discrete-time filter object
%
% M-File generated by MATLAB(R) 7.0 and the Signal Processing Toolbox 6.2.
% Generated on: 04-Dec-2006 22:43:57
%
% Chebyshev Type II Highpass filter designed using FDESIGN.HIGHPASS.
% All frequency values are normalized to 1.
Fstop = 0.25; % Stopband Frequency
Fpass = 0.3; % Passband Frequency
Astop = 26; % Stopband Attenuation (dB)
Apass = 1.93; % Passband Ripple (dB)
match = 'passband'; % Band to match exactly
% Construct an FDESIGN object and call its CHEBY2 method.
h = fdesign.highpass(Fstop, Fpass, Astop, Apass);
Hd = cheby2(h, 'MatchExactly', match);

% [EOF]

Filter dargestellt mit showfilter.m:

Wpass = 0.3; Wstop = 0.25; Apass = 1.93; Astop = 26;
[N, Wn] = cheb2ord(Wpass, Wstop, Apass, Astop);
[B, A] = cheby2(N, Astop, Wn, 'high');
showfilter(B, A, 'Durchfuhrung 5.4.2 – Cheby2-HP', 0.25, 40);
Das Chebychev2-Filter hat die gleiche Ordnung wie das Chebychev1. Die Polstellen liegen weiterhin innerhalb des Einheitskreises, die Nullstellen befinden sich nun auf dem Einheitskreis verteilt, was bedeutet, dass das Filter minimalphasig ist.
5.4.2.4. Elliptische oder Cauer-Filter

Tiefpaß:

Matlab-Coefficient File (ellipTP.fcf):

```
% Generated by MATLAB(R) 7.0 and the Signal Processing Toolbox 6.2.
% Generated on: 04-Dec-2006 23:00:09

Discrete-Time IIR Filter (real)

Filter Structure : Direct-Form II, Second-Order Sections
Filter Order : 3
Stable : Yes
Linear Phase : No

SOS matrix:
1  -1.731332750128489  1  1  -1.799201182309701  0.892602576277942
1  1.000000000000000  0  1  -0.835369129223804  0.000000000000000

Scale Factors:
0.159916528984199
0.178947878051409
1.000000000000000
```

Breitschaft, Marco
Milewski, Mario
Uhl, Michael

11.01.2007
Praktikum Digitale Signalverarbeitung
Gruppe 4 im WS 2006/07
Versuch: Digitale Filter

Matlab-M-File (ellipTP.m):

function Hd = ellipTP
%ELLIPTP Returns a discrete-time filter object

% M-File generated by MATLAB(R) 7.0 and the Signal Processing Toolbox 6.2.
% Generated on: 04-Dec-2006 22:59:44

% Elliptic Lowpass filter designed using FDESIGN.LOWPASS.
% All frequency values are normalized to 1.
Fpass = 0.1;         % Passband Frequency
Fstop = 0.15;        % Stopband Frequency
Apass = 1.93;        % Passband Ripple (dB)
Astop = 26;          % Stopband Attenuation (dB)
m {h}  = 'stopband'; % Band to match exactly

% Construct an FDESIGN object and call its ELLIP method.
{h} = fdesign.lowpass(Fpass, Fstop, Apass, Astop);
Hd = ellip(h, 'MatchExactly', match);

% [EOF]

Filter dargestellt mit showfilter.m:

Wpass = 0.1; Wstop = 0.15; Apass = 1.93; Astop = 26;
{[N, Wn] = ellipord(Wpass, Wstop, Apass, Astop);
[B, A] = ellip(N, Apass, Astop, Wn, 'low');
showfilter(B, A, 'Durchfuhrung 5.4.2 - Ellip-TP', 0.25, 40);
Praktikum Digitale Signalverarbeitung
Gruppe 4 im WS 2006/07
Versuch: Digitale Filter

Besser mit N=100 Abtastwerten:
Hochpaß:

Matlab-Coefficient File (ellipHP.fcf):

```matlab
% % Generated by MATLAB(R) 7.0 and the Signal Processing Toolbox 6.2.
% % Generated on: 04-Dec-2006 23:25:04
% Discrete-Time IIR Filter (real)
-----------------------------------------------
Filter Structure : Direct-Form II, Second-Order Sections
Filter Order : 4
Stable : Yes
Linear Phase : No

SOS matrix:
1  -1.852534608301975  1  1  -0.473865374930582  0.263593092023464
1  -1.459169297052111  1  1  -1.125073547581566  0.863414979587882

Scale Factors:
0.424366254468867
0.863932427278487
1.000000000000000
```

Matlab-M-File (ellipHP.m):

```````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````````
function Hd = ellipHP
% ELLIPHP Returns a discrete-time filter object
% M-File generated by MATLAB(R) 7.0 and the Signal Processing Toolbox 6.2.
% Generated on: 04-Dec-2006 23:24:51
% Elliptic Highpass filter designed using FDESIGN.HIGHPASS.
% All frequency values are normalized to 1.
Fstop = 0.25; % Stopband Frequency
Fpass = 0.3; % Passband Frequency
Astop = 26; % Stopband Attenuation (dB)
Apass = 1.93; % Passband Ripple (dB)
match = 'stopband'; % Band to match exactly
% Construct an FDESIGN object and call its ELLIP method.
h = fdesign.highpass(Fstop, Fpass, Astop, Apass);
Hd = ellip(h, 'MatchExactly', match);

% EOF

Filter dargestellt mit showfilter.m:

Wp = 0.3; Ws = 0.25; Ap = 1.93; As = 26;
[N, Wn] = ellipord(Wp, Ws, Ap, As);
[B, A] = ellip(N, Ap, As, Wn, 'high');
showfilter(B, A, 'Durchführung 5.4.2 – Ellip-HP', 0.25, 40);
5.4.3. Entwurf nichtrekursiver Filter mit Matlab

5.4.3.1. Fourierapproximation ohne Fenster (bzw. mit Rechteckfenster)

\[ \begin{align*}
fg &= 1.25; \\
fa &= 10; \\
N &= [-100:1:100]; \\
X &= N(1:100); \\
Gn &= \text{sinc}(X*2*\pi*fg/fa); \\
Gn(101) &= 1; \\
X &= N(102:201); \\
Gn(102:201) &= \text{sinc}(X*2*\pi*fg/fa); \\
A &= [1 \ zeros(1, \text{length}(Gn)-1)]; \\
\text{showfilter}(Gn,A,'...',0.5,200);
\end{align*} \]
5.4.3.2. Fourierapproximation mit Kaiserfenster

\[
N = [-100:1:100];
G_n(1:100) = \text{sinc}(N(1:100) \times 2\pi / 8);
G_n(101) = 1;
G_n(102:201) = \text{sinc}(N(102:201) \times 2\pi / 8);
\%
\text{plot}(G_n);
\%
\text{stem}(N, G_n), \text{title('Manuelle Berechnung der Impulsantwort')}, x\text{label('N')}, y\text{label('G(N)')};
\%
f_k = \text{kaiser(length(N), 3)};
hf = G_n \times f_k'; \text{ % Impulsantwort multipliziert mit Zeilenvektor von } f_k
\%
\text{showfilter(hf, 1, 'Durchführung 5.4.3.1 - Fourierapprox mit Kaiserfenster', 0.5, 200);
}
\%
\]

Zusätzlich wurde noch der Hoch- und Tiefpass nach dem vorliegenden Toleranzschema mit dem Matlab Toolsets „wintool“ und „fdatool“ entwickelt:
Praktikum Digitale Signalverarbeitung
Gruppe 4 im WS 2006/07
Versuch: Digitale Filter

a) TP

\[ W_{\text{pass}} = 0.1; W_{\text{stop}} = 0.15; A_{\text{pass}} = 0.8; A_{\text{stop}} = 0.05; \]
\[ [N, W_n, \text{BETA, TYPE}] = \text{kaiserord}([W_{\text{pass}} W_{\text{stop}}], [1 0], [A_{\text{stop}} A_{\text{pass}}]); \]
\[ [B, A] = \text{fir1}(N, W_n, \text{TYPE}, \text{kaiser}(N+1, \text{BETA})); \]
\[ \text{showfilter}(B, A, \ '\text{Durchfuehrung 5.4.3 - Kaiser-TP}', 0.25, 40); \]
Praktikum Digitale Signalverarbeitung
Gruppe 4 im WS 2006/07
Versuch: Digitale Filter

Breitschaft, Marco
Milewski, Mario
Uhl, Michael
11.01.2007
Seite 56 von 70
Entwicklung mit dem Filter Design Tool:

Matlab-Coefficient File (kaiserTP.fcf):

```
% % Generated by MATLAB(R) 7.0 and the Signal Processing Toolbox 6.2.
% % Generated on: 05-Dec-2006 21:28:06
% % Discrete-Time FIR Filter (real)
% 
% Filter Structure : Direct-Form FIR
% Filter Order : 51
% Stable : Yes
% Linear Phase : Yes (Type 2)

Numerator:
-0.004186661757874
-0.001597962388617
0.001735276332056
0.005362908074962
0.008707427406639
0.011125392575765
0.012291309020288
0.01130254164266
0.00803303622809
0.003089507567640
0.00036907493151
-0.010487953329575
-0.017211229734232
```
Matlab-M-File (kaiserTP.m):

function Hd = kaiser-TP
%KAISER-TP Returns a discrete-time filter object
%
% M-File generated by MATLAB(R) 7.0 and the Signal Processing Toolbox 6.2.
% Generated on: 05-Dec-2006 21:27:15
%
% FIR Window Lowpass filter designed using the FIR1 function.
% All frequency values are normalized to 1.
Fpass = 0.1;        % Passband Frequency
Fstop = 0.15;       % Stopband Frequency
Dpass = 0.8;        % Passband Ripple
Dstop = 0.05;       % Stopband Attenuation
flag  = 'noscale';  % Sampling Flag

% Calculate the order from the parameters using KAISERORD.
[N,Wn,BETA,TYPE] = kaiserord([Fpass Fstop], [1 0], [Dstop Dpass]);

% Calculate the coefficients using the FIR1 function.
b  = fir1(N, Wn, TYPE, kaiser(N+1, BETA), flag);
Hd = dfilt.dffir(b);

% [EOF]
b) HP

\[ W_{pass} = 0.3; W_{stop} = 0.25; A_{pass} = 0.8; A_{stop} = 0.05; \]
\[
[N, Wn, BETA, TYPE] = kaiserord([W_{stop} W_{pass}], [0 1], [A_{stop} A_{pass}]);
\]
\[
[B, A] = fir1(N, Wn, TYPE, kaiser(N+1, BETA));
\]
\[
showfilter(B, A, 'Durchführung 5.4.3 - Kaiser-HP', 0.25, 60);
\]
Entwicklung mit dem Filter Design Tool:

Matlab-Coefficient File (kaiserHP.fcf):

```
% Generated by MATLAB(R) 7.0 and the Signal Processing Toolbox 6.2.
% Generated on: 05-Dec-2006 21:44:54

Discrete-Time FIR Filter (real)
-----------------------------
Filter Structure : Direct-Form FIR
Filter Order      : 52
Stable            : Yes
Linear Phase      : Yes (Type 1)

Numerator:
0.003355386124275
-0.003069342165513
-0.008270560437595
-0.008034326829243
-0.001596699917130
0.007179889586729
0.011977341611241
0.008432080989070
-0.002203668319411
-0.013048599428049
-0.015839410657942
-0.006951449302219
0.009020069512884
0.021205693613065
0.008432080989070
0.011977341611241
0.008432080989070
-0.002203668319411
0.009020069512884
Praktikum Digitale Signalverarbeitung
Gruppe 4 im WS 2006/07

Versuch: Digitale Filter

Breitschaft, Marco Milewski, Mario Uhl, Michael

11.01.2007 Seite 61 von 70
5.4.3.3. Chebyshev-Approximation

\[ N = [-100:1:100]; \]
\[ G_n(1:100) = \text{sinc}(N(1:100) \times 2\pi/8); \]
\[ G_n(101) = 1; \]
\[ G_n(102:201) = \text{sinc}(N(102:201) \times 2\pi/8); \]
\[ \text{stem}(N, G_n), \text{title}('Manuelle Berechnung der Impulsantwort'), \text{xlabel('N'), ylabel('G(N)');} \]
\[ cf = \text{chebwin}(\text{length}(N), 26); \]
\[ hf = G_n .* cf'; \] % Impulsantwort multipliziert mit Zeilenvektor von kf
\[ \text{showfilter}(hf, 1, 'Durchfuehrung 5.4.3.3 - Chebyshevapprox', 0.5, 200); \]

Zusätzlich wurde noch der Hoch- und Tiefpass nach dem vorliegenden Toleranzschema mit dem Matlab Toolsets „wintool“ und „fdatool“ entwickelt:
a) TP

Einhaltung des Bandabstands von 26dB → Dämpfung der Nebenkeule (SidelobeAtten) 26dB

Die geforderten Frequenzen $W_{\text{pass}}=0.1f_a$ und $W_{\text{Stop}}=0.15f_a$ ($W_C=0.125$) wurden mit einem Filter nach der Chebyshevapproximation mit der Ordnung 68 erreicht.
Matlab-Coefficient File (chebyshev-approxTP.fcf):

% Generated by MATLAB(R) 7.0 and the Signal Processing Toolbox 6.2.
% Generated on: 05-Dec-2006 23:18:08
% Discrete-Time FIR Filter (real)
-----------------------------
Filter Structure : Direct-Form FIR
Filter Order      : 68
Stable            : Yes
Linear Phase      : Yes (Type 1)

Numerator:
0.006619972912920
0.000736036004925
-0.000000000000000
-0.000935093016232
-0.001936072184234
-0.002824981952464
-0.003404615941420
-0.003493038964647
-0.002961988729794
-0.001772433669532
0.000000000000000
0.002156274981025
0.004386341542846
0.006303960354063
0.007501374221564
0.00761349068889
0.006408867347345
0.003814831045377
-0.000000000000000
-0.004630738354916
-0.009453926316121
-0.013685924440475
-0.016473320429304
-0.017004992169197
-0.014631106465558
-0.008971990870574
0.000000000000000
0.011919659931707
0.026037985843769
0.041283097786455
0.056367511164761
0.069929252464715
0.080688928391184
0.087601977863294
0.089985467547245
0.087601977863294
0.080688928391184
0.069929252464715
0.056367511164761
0.041283097786455
0.026037985843769
-0.011919659931707
-0.000000000000000
-0.008971990870574
-0.014631106465558
-0.017004992169197
-0.016473320429304
-0.013685924440475
Praktikum Digitale Signalverarbeitung
Gruppe 4 im WS 2006/07

Versuch: Digitale Filter

Matlab-M-File (chebyshev-approxTP.m):

function Hd = chebyshev-approxTP

%CHEBYSHEV-APPROXTP Returns a discrete-time filter object

% M-File generated by MATLAB(R) 7.0 and the Signal Processing Toolbox 6.2.
% Generated on: 05-Dec-2006 23:17:26

% FIR Window Lowpass filter designed using the FIR1 function.
% All frequency values are normalized to 1.
N             = 68;         % Order
Fc            = 0.125;      % Cutoff Frequency
flag          = 'noscale';  % Sampling Flag
SidelobeAtten = 26;         % Window Parameter

% Create the window vector for the design algorithm.
win = chebwin(N+1, SidelobeAtten);

% Calculate the coefficients using the FIR1 function.
b  = fir1(N, Fc, 'low', win, flag);
Hd = dfilt.dffir(b);

% [EOF]

Filter dargestellt mit showfilter.m:

N = 68; Wc = 0.125; SidelobeAtten = 26;
win = chebwin(N+1, SidelobeAtten);
[B, A] = fir1(N, Wc, 'low', win);
showfilter(B,A, 'Durchfuehrung 5.4.3 – Chebyshev-window-TP',0.25,80);
b) HP

Einhaltung des Bandabstands von 26dB $\rightarrow$ Dämpfung der Nebenkeule (SidelobeAtten) 26dB

Die geforderten Frequenzen $W_{\text{Stop}}=0.25f_a$ und $W_{\text{pass}}=0.3f_a$ ($\rightarrow W_C=0.275$) wurden mit einem Filter nach der Chebyshevapproximation mit der Ordnung 78 erreicht.

Matlab-Coefficient File (chebyshev-approxHP.fcf):

```matlab
% % Generated by MATLAB(R) 7.0 and the Signal Processing Toolbox 6.2.
% % Generated on: 05-Dec-2006 23:48:12
% Discrete-Time FIR Filter (real)
% Filter Structure : Direct-Form FIR
% Filter Order : 78
% Stable : Yes
% Linear Phase : Yes (Type 1)

Numerator:
-0.006206275291375
-0.001438890912657
-0.000847128812565
0.000555727615623
0.001837421724795
0.001954393567086
```
### Praktikum Digitale Signalverarbeitung

**Gruppe 4 im WS 2006/07**

**Versuch: Digitale Filter**

| 0.000563366780278 |
| -0.001557187413150 |
| -0.002893582360831 |
| -0.002244635520406 |
| 0.000271960075995 |
| 0.003057890807036 |
| 0.004002974694559 |
| 0.002033538721394 |
| -0.001863591996035 |
| -0.00503226256240 |
| -0.004899920865904 |
| -0.00976217774323 |
| 0.004401217857274 |
| 0.0073616717575986 |
| 0.005196665331352 |
| -0.001361787253182 |
| -0.00805127954917 |
| -0.009640071912186 |
| -0.0063295015494672 |
| 0.005631663695705 |
| 0.013270623327193 |
| 0.012199400127986 |
| 0.001341746565603 |
| -0.013243969394186 |
| -0.021056315979341 |
| -0.014152227248184 |
| 0.006498733271732 |
| 0.029230297994466 |
| 0.036680255543770 |
| 0.015442510864775 |
| -0.035002266209961 |
| -0.099629893582489 |
| -0.153760584496000 |
| 0.460914667644684 |
| -0.153760584496000 |
| -0.099629893582489 |
| -0.035002266209961 |
| 0.015442510864775 |
| 0.036680255543770 |
| 0.029230297994466 |
| 0.006498733271732 |
| -0.014152227248184 |
| -0.021056315979341 |
| -0.013243969394186 |
| 0.001341746565603 |
| 0.012199400127986 |
| 0.013270623327193 |
| 0.005631663695705 |
| -0.0043295015494672 |
| -0.009840071912186 |
| -0.00805127954917 |
| -0.001361787253182 |
| 0.005196665331352 |
| 0.0073616717575986 |
| 0.004401217857274 |
| -0.00976217774323 |
| -0.004899920865904 |
| -0.00503226256240 |
| -0.00195439567086 |
| 0.001837421724795 |
| 0.00555727615623 |
| -0.00847128812565 |

---

Breitschaft, Marco  Milewski, Mario  Uhl, Michael

11.01.2007  Seite 68 von 70
Matlab-M-File (chebyshev-approxHP.m):

```matlab
function Hd = chebyshev-approxHP
%CHEBYSHEV-APPROXHP Returns a discrete-time filter object

% M-File generated by MATLAB(R) 7.0 and the Signal Processing Toolbox 6.2.
% Generated on: 05-Dec-2006 23:47:23

% FIR Window Highpass filter designed using the FIR1 function.
% All frequency values are normalized to 1.

N = 78;         % Order
Fc = 0.275;      % Cutoff Frequency
flag = 'noscale'; % Sampling Flag
SidelobeAtten = 26; % Window Parameter

% Create the window vector for the design algorithm.
win = chebwin(N+1, SidelobeAtten);

% Calculate the coefficients using the FIR1 function.
b = fir1(N, Fc, 'high', win, flag);
Hd = dfilt.dffir(b);

% [EOF]
```

Filter dargestellt mit showfilter.m:

```matlab
N = 78; Wc = 0.275; SidelobeAtten = 26;
win = chebwin(N+1, SidelobeAtten);
[B, A] = fir1(N, Wc, 'high', win);
showfilter(B, A, 'Durchfuehrung 5.4.3 – Chebyshev-window-HP',0.25,60);
```