

Appendix D: Design of lowpass, highpass, bandpass, and bandstop filters as well as Hilbert transformers and differentiators using fixed windows

```
%-----  
% Matlab m-file (firwifix.m)  
% for designing a lowpass, highpass, bandpass, or  
% bandstop linear-phase FIR filter as well as  
% Hilbert transformers and differentiators using  
% fixed window functions:  
%  
% - rectangular  
% - Bartlett  
% - Hann  
% - Hamming  
% - Blackman  
%  
%  
% For lowpass and highpass cases, the passband and  
% stopband edges  $w_p$  and  $w_s$  are given as fractions  
% of  $\pi$  (half the sampling rate).  
% For bandpass and bandstop cases, two passband  
% edges  $w_{p1}$  and  $w_{p2}$  as well as two stopband edges  
%  $w_{s1}$  and  $w_{s2}$  are given.  
% For Hilbert transformers, only the lower cutoff  $w_p$   
% is given. For the even order, another cutoff as  
% a fraction of  $\pi$  is at  $1-w_c$ .  
% For differentiators,  $w_p$  and  $w_s$  are given: on  
%  $[0, \pi \cdot w_p]$  it is desired to approximate  $\omega$   
% and on  $[\pi \cdot w_p, \pi]$  zero.  
%  
% Tapio Saramäki 15.11.97  
%  
% This file can be found in SUN's:  
% ~ts/matlab/dsp/firwifix.m  
%-----  
clear all;close all  
disp('I am a program for designing a lowpass, highpass,')  
disp('bandpass, and bandstop linear-phase FIR filter')  
disp('as well as Hilbert transformers and differentiators')  
disp('with the aid of the rectangular, Bartlett, Hann')  
disp('Hamming, or Blackman window')  
disp('As input data, I first need the filter type')  
disp('1 for lowpass filter, 2 for highpass filter');
```

```

disp('3 for bandpass filter, 4 for bandstop filter');
disp('5 for Hilbert transformer, and')
firtyp=input('6 for differentiator:');
%-----
if firtyp < 3
    wp=input('Passband edge as a fraction of pi: ');
    ws=input('Stopband edge as a fraction of pi: ');
    wc=(wp+ws)/2;
    Deltaw=abs(ws-wp);
end
%-----
if firtyp > 2 & firtyp < 5
    wp(1)=input('First passband edge as a fraction of pi: ');
    wp(2)=input('Second passband edge as a fraction of pi: ');
    ws(1)=input('First stopband edge as a fraction of pi: ');
    ws(2)=input('Second stopband edge as a fraction of pi: ');
    wc=(wp+ws)/2;
    Deltaw=min(abs(ws-wp));
end
%-----
if firtyp==5
    wp(1)=input('First passband edge as a fraction of pi: ');
    Deltaw=2*wp(1);
end
%-----
if firtyp==6
    disp('For the differentiator the passband edge is the')
    disp('the point up to which you like to approximate omega')
    disp('The stopband edge is the point after which')
    disp('you like to approximate zero')
    wp=input('Passband edge as a fraction of pi: ');
    ws=input('Stopband edge as a fraction of pi: ');
    wc=(wp+ws)/2;
    Deltaw=abs(ws-wp);
end
%-----
disp('Now it is time to select one among the following windows:');
disp('rectangular, Bartlett, Hann, Hamming, or Blackman window');
As1=20.9;d1=10^(-As1/20);
As3=43.9;d3=10^(-As3/20);
As4=54.5;d4=10^(-As4/20);
As5=75.3;d5=10^(-As5/20);
if firtyp < 5
    disp('The estimated stopband attenuation is:')
    disp('for rectangular window: 21 dB');
    disp('no estimate for the Bartlett window');

```

```

disp('for the Hann window: 44 dB');
disp('for the Hamming window: 54 dB ');
disp('for the Blackman window: 75 dB ');
end
if firtyp==5
disp('The estimated deviation from unity in the passband is:')
disp('for the rectangular window: 0.18 ');
disp('no estimate for the Bartlett window');
disp('for the Hann window: 0.013 ');
disp('for the Hamming window: 0.0038 ');
disp('for the Blackman window: 0.00035 ');
end
if firtyp==6
disp('The estimated maximum deviation from omega in the')
disp('passband and from zero in the stopband is:')
fprintf('for the rectangular window: %g\,d1*wc*pi);
disp(' ');
disp('no estimate for the Bartlett window');
fprintf('for the Hann window: %g\,d3*wc*pi);
disp(' ');
fprintf('for the Hamming window: %g\,d4*wc*pi);
disp(' ');
fprintf('for the Blackman window: %g\,d5*wc*pi);
disp(' ');
end
%-----
disp('Select the window type:')
disp('Type 1 for rectangular, 2 for Bartlett, 3 for Hann')
itype=input('4 for Hamming, 5 for Blackman: ');
%-----
% Estimate the order N; in highpass and bandstop cases N must
% be even. For Bartlett window, there exists no formula!!
%-----
if itype==1 N=rectord(0,Deltaw)-1; end
if itype==3 N=hannord(0,Deltaw)-1; end
if itype==4 N=hammord(0,Deltaw)-1; end
if itype==5 N=blackord(0,Deltaw)-1; end
%-----
if firtyp==2 | firtyp==4
if rem(N,2)==1; N=N+1;end
end
%-----
if itype < 2 | itype > 2
fprintf('The estimated order is %g\,N);
disp(' ');
disp('For highpass and bandstop cases, N must be even!!');

```

```

N=input('Your selection for the order: ')
end
if itype==2
    disp('For the Bartlett window, there exists no estimate!!');
    disp('For highpass and bandstop cases, order must be even!!');
    N=input('Your selection for the order: ');
end
ll=1;
while ll==1
%-----
    if itype==1 wind=rectan(N+1); end
    if itype==2 wind=bartlett(N+1); end
    if itype==3 wind=hann(N+1); end
    if itype==4 wind=hamming(N+1); end
    if itype==5 wind=blackman(N+1); end
%-----
    if firtyp==1 h=firwind(N,wc,wind);end
    if firtyp==2 h=firwind(N,wc,wind,'high');end
    if firtyp==3 h=firwind(N,wc,wind);end
    if firtyp==4 h=firwind(N,wc,wind,'stop');end
    if firtyp==5 h=firwind(N,wc,wind,'Hilbert');end
    if firtyp==6 h=firwind(N,wc,wind,'differentiator');end
%-----
    % Plot the responses.
%-----
    winds=wind;hs=h;
    figure(1)
    subplot(211)
    [H,W]=zeroam(winds,.0,1.,8*1024);
    plot(W/pi,20*log10(abs(H)));
    amin=2.5*max(20*log10(abs(H(4*1024:8*1024+1))));
    if amin > 0
        amin=-50;
    end
    amax=1.2*max(20*log10(abs(H(1:1024))));
    grid;axis([0 1 amin amax])
    title('Window function');
    ylabel('Amplitude in dB');
    xlabel('Angular frequency omega/pi')
    subplot(212)
    impz(winds); grid;
    title('Window function');xlabel('n in samples');
    ylabel('Impulse response');title('Window function');
%-----
    % Lowpass, highpass, bandstop, and bandpass filters
%-----

```

```

if firtyp < 5
figure(2)
subplot(211)
[H,W]=zeroam(hs,.0,1.,8*1024);
amm=-120;
plot(W/pi,20*log10(abs(H)));axis([0 1 amm 20]);grid;
title('Resulting filter');
ylabel('Amplitude in dB');
xlabel('Angular frequency omega/pi');
subplot(212)
impz(hs);
xlabel('n in samples');
ylabel('Impulse response');
title('Resulting filter'); grid
figure(3)
if firtyp==1
x1=0;x2=wp;
end
if firtyp==2
x1=wp;x2=1;
end
if firtyp==3
x1=wp(1);x2=wp(2);
end
if firtyp==4
x1=0;x2=1;
end
subplot(211)
dpp=1.2*(max(H)-1);
if itype==2 dpp=.05; end
plot(W/pi,H);axis([x1 x2 1-dpp 1+dpp]);
grid;
title('Passband details for the resulting filter');
ylabel('Zero-phase response');
xlabel('Angular frequency omega/pi');
if firtyp==1
x1=ws;x2=1;
end
if firtyp==2
x1=0;x2=ws;
end
if firtyp==3
x1=0;x2=1;
end
if firtyp==4
x1=ws(1);x2=ws(2);
end

```

```

end
subplot(212)
plot(W/pi,H);axis([x1 x2 -dpp dpp]);
grid;
title('Stopband details for the resulting filter');
ylabel('Zero-phase response');
xlabel('Angular frequency omega/pi');
end
%-----
% Hilbert transformer
%-----
if firtyp==5
figure(2)
subplot(211)
[H,W]=zeroam(hs,.0,1.,8*1024);
dpp=1.02*max(H);
plot(W/pi,(abs(H)));axis([0 1 0 dpp]);grid;
title('Resulting filter');
ylabel('Amplitude');
xlabel('Angular frequency omega/pi');
subplot(212)
impz(hs);
title('Resulting filter');
xlabel('n in samples');
ylabel('Impulse response');
title('Resulting filter');grid;
figure(3)
if rem(N,2)==0
x1=wp;x2=1-wp;
end
if rem(N,2)==1
x1=wp;x2=1;
end
end
[HH,WW]=zeroam(hs,x1,x2,4*1024);
dpp1=max(HH)-1;dpp2=1-min(HH);
dpp=max(dpp1,dpp2);
plot(W/pi,H);axis([x1 x2 1-dpp 1+dpp]);
grid;
title('Passband details for the resulting filter');
ylabel('Zero-phase response');
xlabel('Angular frequency omega/pi');
end
%-----
% Differentiator
%-----
if firtyp==6

```

```

figure(2)
subplot(211)
[H,W]=zeroam(hs,.0,1.,8*1024);
plot(W/pi,(abs(H))/pi);
axis([0 1 0 1.1*max(abs(H)/pi)]);grid;
title('Resulting filter');
ylabel('Amplitude as a fraction of pi');
xlabel('Angular frequency omega/pi');
subplot(212)
impz(hs);
title('Resulting filter');
xlabel('n in samples');
ylabel('Impulse response');
title('Resulting filter');
grid;
figure(3)
x1=0;x2=wp;
[HH,WW]=zeroam(hs,x1,x2,4*1024);
dpp1=max(HH-WW);dpp2=abs(min(HH-WW));
dpp=max(dpp1,dpp2);
subplot(211)
plot(W/pi,H-W);axis([x1 x2 -dpp dpp]);
grid;
title('H(w)-w for the resulting filter');
ylabel('Zero-phase response');
xlabel('Angular frequency omega/pi');
x1=ws;x2=1;
[HH,WW]=zeroam(hs,x1,x2,4*1024);
dpp1=max(HH);dpp2=abs(min(HH));
dpp=max(dpp1,dpp2);
subplot(212)
plot(W/pi,H);axis([x1 x2 -dpp dpp]);grid;
title('Stopband details for the resulting filter');
ylabel('Zero-phase response');
xlabel('Angular frequency omega/pi');
end
%-----
disp('Are you satisfied with the result or do you like');
disp('to change the filter order');
ll=input('1 for changing the order, 0 for not');
if ll==1
    disp('For highpass and bandstop cases, order is even')
    N=input('New filter order');
end
end
%-----

```

```

save fircoe hs -ascii -double
disp('For further use, you can find')
disp('the filter coefficients in fircoe')
*****f

```

```

unction [N,wc]=rectkord(wp,ws)
%-----
% Given the passband and stopband edges,
% wp and ws as a fraction of pi (half the
% sampling rate) for a lowpass filter,
% [N,wc]=rectord(wp,ws) evaluates for
% the rectangular window estimated length
% N=ceil(1.84/(ws-wp)+1) as well as
% wc=(wp+ws)/2.
%-----

```

```

%-----
% See T. Saramäki, "Finite impulse response Filter
% Design" in Handbook for Digital Signal Processing,
% S. K. Mitra and J. F. Kaiser, Eds, John Wiley &
% Sons, 1993
%-----

```

```

% Tapio Saramäki 2.11.1997
% Can be found in SUN's: ~ts/matlab/dsp/rectord.m
%-----

```

```

% Cutoff frequency of the ideal filter
%-----

```

```

wc=(wp+ws)/2;
%-----

```

```

% Determine N
%-----

```

```

N=ceil(1.84/(ws-wp)+1);
%-----

```

```

*****f

```

```

unction wind = rectan(n)
%rectan(n) returns the n-point rectangular window.

```

```

%-----
% Tapio Saramäki November 2, 1997
% Can be found in SUN's: ~ts/matlab/dsp
%-----

```

```

nn = round(n);
wind=ones(size(1:nn));

```

```

*****

```

```

function wind = bartlett(n)
%bartlett(n) returns the n-point Bartlett window.

```



```

% See Table 4-3 in T. Saramäki, "Finite impulse
% response Filter Design" in Handbook for Digital
% Signal Processing, S. K. Mitra and J. F. Kaiser,
% Eds, John Wiley & Sons, 1993. Also even values
% for n are made possible.
%-----
% Tapio Saramäki November 2, 1997
% Can be found in SUN's: ~ts/matlab/dsp
%-----
nn = round(n);
nodd=rem(nn-1,2);
if nodd==0
    m=-(nn-1)/2:1:(nn-1)/2;
    wind=-abs(m)/((nn-1)/2+1)+1;
end
if nodd==1
    nnn=2*nn-1;
    m=-(nnn-1)/2:2:(nnn-1)/2;
    wind=-abs(m)/((nnn-1)/2+1)+1;
end
*****f
unction [N,wc]=hannord(wp,ws)
%-----
% Given the passband and stopband edges,
% wp and ws as a fraction of pi (half the
% sampling rate) for a lowpass filter,
% [N,wc]=hannord(wp,ws) evaluates for
% the Hann window estimated length
% N=ceil(6.22/(ws-wp)+1) as well as
% wc=(wp+ws)/2.
%-----

%-----
% See T. Saramäki, "Finite impulse response Filter
% Design" in Handbook for Digital Signal Processing,
% S. K. Mitra and J. F. Kaiser, Eds, John Wiley &
% Sons, 1993
%-----
% Tapio Saramäki 2.11.1997
% Can be found in SUN's: ~ts/matlab/dsp/hannord.m
%-----
% Cutoff frequency of the ideal filter
%-----
wc=(wp+ws)/2;
%-----
% Determine N

```

```

%-----
N=ceil(6.22/(ws-wp)+1);
%-----
*****f

unction wind = hann(n)
%hann(n) returns the n-point Hann window.

% See Table 4-3 in T. Saramäki, "Finite impulse
% response Filter Design" in Handbook for Digital
% Signal Processing, S. K. Mitra and J. F. Kaiser,
% Eds, John Wiley & Sons, 1993. Also even values
% for n are made possible.
%-----
% Tapio Saramäki November 2, 1997
% Can be found in SUN's: ~ts/matlab/dsp
%-----
nn = round(n);
m=-(nn-1)/2:1:(nn-1)/2;
wind=.5*cos(2*pi*m/nn)+0.5;
*****f

unction [N,wc]=harmord(wp,ws)
%-----
% Given the passband and stopband edges,
% wp and ws as a fraction of pi (half the
% sampling rate) for a lowpass filter,
% [N,wc]=harmord(wp,ws) evaluates for
% the Hamming window estimated length
% N=ceil(6.64/(ws-wp)+1) as well as
% wc=(wp+ws)/2.
%-----

%-----
% See T. Saramäki, "Finite impulse response Filter
% Design" in Handbook for Digital Signal Processing,
% S. K. Mitra and J. F. Kaiser, Eds, John Wiley &
% Sons, 1993
%-----
% Tapio Saramäki 2.11.1997
% Can be found in SUN's: ~ts/matlab/dsp/harmord.m
%-----
% Cutoff frequency of the ideal filter
%-----
wc=(wp+ws)/2;
%-----
% Determine N
%-----

```

```

N=ceil(6.64/(ws-wp)+1);
%-----
*****f

unction wind = hamming(n)
%hamming(n) returns the n-point Hamming window.

% See Table 4-3 in T. Saramäki, "Finite impulse
% response Filter Design" in Handbook for Digital
% Signal Processing, S. K. Mitra and J. F. Kaiser,
% Eds, John Wiley & Sons, 1993. Also even values
% for n are made possible.
%-----
% Tapio Saramäki November 2, 1997
% Can be found in SUN's: ~ts/matlab/dsp
%-----
nn = round(n);
m=-(nn-1)/2:1:(nn-1)/2;
wind=.46*cos(2*pi*m/nn)+0.54;
*****f

unction [N,wc]=blackord(wp,ws)
%-----
% Given the passband and stopband edges,
% wp and ws as a fraction of pi (half the
% sampling rate) for a lowpass filter,
% [N,wc]=blackord(wp,ws) evaluates for
% the Blackman window estimated length
% N=ceil(11.13/(ws-wp)+1) as well as
% wc=(wp+ws)/2.
%-----

%-----
% See T. Saramäki, "Finite impulse response Filter
% Design" in Handbook for Digital Signal Processing,
% S. K. Mitra and J. F. Kaiser, Eds, John Wiley &
% Sons, 1993
%-----
% Tapio Saramäki 2.11.1997
% Can be found in SUN's: ~ts/matlab/dsp/blackord.m
%-----
% Cutoff frequency of the ideal filter
%-----
wc=(wp+ws)/2;
%-----
% Determine N
%-----
N=ceil(11.13/(ws-wp)+1);

```

```

%-----
*****f
unction wind = blackman(n)
%blackman(n) returns the n-point Blackman window.

% See Table 4-3 in T. Saramäki, "Finite impulse
% response Filter Design" in Handbook for Digital
% Signal Processing, S. K. Mitra and J. F. Kaiser,
% Eds, John Wiley & Sons, 1993. Also even values
% for n are made possible.
%-----
% Tapio Saramäki November 2, 1997
% Can be found in SUN's: ~ts/matlab/dsp
%-----
nn = round(n);
m=-(nn-1)/2:1:(nn-1)/2;
wind=.5*cos(2*pi*m/nn)+0.42;
wind=wind+.08*cos(4*pi*m/nn);
*****

```