Appendix C: Design of differentiators using adjustable windows

```
*******************************************************************************************
%-----------------------------------------------------
% Matlab m-file (difwinad.m)
% for designing an FIR differntiator using the
% Kaiser, Saramaki, Dolph-Chebyshev, or transitional
% window.
%
% The filters are designed in such a way that the
% desired stopband or passband criterion just met
% and the other one is exceeded.
%
% For the differentiator, the use gives the passband
% edges wp and ws as fractions of pi (half the
% sampling rate). dp is the maximumdeviation from
% omega in the passband and ds is the maximum
% deviation from zero in the stopband.
%
% Note that in the program N is the filter length
% so that N-1 is the order
%
% Tapio Saramäki  27.10.97
%
% This file can be found in SUN's:
% ~ts/matlab/dsp/hilwinad.m
%-----------------------------------------------------
clear all;close all
disp('I am a program for designing an FIR differentiator')
disp('with the aid of the Kaiser, Saramaki,')
disp('Dolph-Chebyshev, or transitional window')
disp('As input data, I first need the following:')
%------------------------------------------------------
wp=input('Passband edge as a fraction of pi: ');
ws=input('Stopband edge as a fraction of pi: ');
dp=input('Maximum deviation from omega in the passband');
ds=input('Maximum deviation from zero in the stopband');
disp('Type 1 for Kaiser, 2 for Saramaki, 3 for')
itype=input('for Dolph-Chebyshev, or 4 for transitional window: ');
%----------------------------------------------------------------
% Estimate the length N; Order is N-1.
%----------------------------------------------------------------
if itype==1
  [N,alpha,wc]=kaiord(wp,ws,dp/(pi/(wp+ws)/2),ds/(pi/(wp+ws)/2));
```

```
end
if itype==2
  [N,beta,wc]=sarord(wp,ws,dp/(pi/(wp+wc)/2),ds/(pi/(wp+ws)/2));
end
if itype==3
  [N,beta,wc]=dchord(wp,ws,dp/(pi/(wp+ws)/2),ds/(pi/(wp+ws)/2));
end
  [N,beta,rho,wc]=traord(wp,ws,dp/(pi/(wp+ws)/2),ds/(pi/(wp+ws)/2));
end
%-------------------------------------------------------------
% Find the filter coefficients h and the window coefficients
% wind for the estimated length N such that either the passband
% or stopband criterion is just met. For a too low value of N,
% this is not true and N is increased. N is the length.
%-------------------------------------------------------------
[h,wind]=difsubad(itype,N,wp,ws,dp,ds);
%-------------------------------------------------------
% Test whether the criteria are met at the edges.
%-------------------------------------------------------
isu=0;
[a,z]=zeroam(h,ws,ws,1);
if abs(a) > ds isu=1;end
[a,z]=zeroam(h,wp,wp,1);
if abs(z-a) > dp isu=1;end
increase=1;
if isu==0 increase=0;end
if isu==0 hs=h;winds=wind;NS=N;end
%-------------------------------------------------------
% Increase the length by 2 until the criteria are met.
%-------------------------------------------------------
if increase==1
  ll=0;
  while ll < 1
    N=N+2;
    [h,wind]=difsubad(itype,N,wp,ws,dp,ds);
%-------------------------------------------------------
% Test whether the criteria are met at the edges.
%-------------------------------------------------------
    isu=0;
    [a,z]=zeroam(h,ws,ws,1);
    if abs(a) > ds isu=1;end
    [a,z]=zeroam(h,wp,wp,1);
    if abs(z-a) > dp isu=1;end
    if isu==0 ll=1;
    hs=h;winds=wind;NS=N;end
  end
```

```
end
%-----------------------------------------------------
% Decrease the length by 2 until the criteria are just met
%-----------------------------------------------------
if increase==0
  ll=0;
  while ll < 1
    N=N-2;
    [h,wind]=difsubad(itype,N,wp,ws,dp,ds);
%-----------------------------------------------------
% Test whether the criterion is met at the lower edge.
%-----------------------------------------------------
    isu=0;
    [a,z]=zeroam(h,ws,ws,1);
    if abs(a) > ds isu=1;end
    [a,z]=zeroam(h,wp,wp,1);
    if abs(z-a) > dp isu=1;end
    ll=1;
    if isu==0 ll=0;
    hs=h;winds=wind;NS=N;end
    end
end
%-----------------------------------------------------
% Test whether the criteria are met by N=NS-1. If yes,
% decrease the length by 2 until the criteria are just met.
%-----------------------------------------------------
N=NS+1;
  ll=0;
    while ll < 1
    N=N-2;
    [h,wind]=difsubad(itype,N,wp,ws,dp,ds);
%-----------------------------------------------------
% Test whether the criteria at are met at the edges.
%-----------------------------------------------------
    isu=0;
    [a,z]=zeroam(h,ws,ws,1);
    if abs(a) > ds isu=1;end
    [a,z]=zeroam(h,wp,wp,1);
    if abs(z-a) > dp isu=1;end
    ll=1;
    if isu==0 ll=0;
    hs=h;winds=wind;NS=N;end
    end
end
fprintf('Minimum length is %g.\',NS)
%-----------------------------------------------------
```

```
% Plot the responses.
%-----------------------------------------------------
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))));
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');
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;
subplot(211)
dpp=dp;
if dp >= 1.5*ds dpp=1.5*ds; end
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;
subplot(212)
dss=ds;
```

```
if ds >= 1.5*dp dss=1.5*dp; end
plot(W/pi,H);axis([x1 x2 -dss dss]);
grid;
title('Stopband details for the resulting filter');
ylabel('Zero-phase response'); xlabel('Angular frequency omega/pi');
save fircoe hs -ascii -double
disp(' ')
disp('For further use, you can find')
disp('the filter coefficients in fircoe')
%-----------------------------------------------------------------------
*****************************************************************************************f
unction [h,wind]=difsubad(itype,N,wp,ws,ddp,dds)
%----------------------------------------------------
% Finds the parameters for the Kaiser (itype=1),
% Saramäki (itype=2), the Dolph-Chebyshev (itype=3),
% and the transitional window (itype=4) such that the
% the passband or stopband ripple criterion of the
% resulting differentiator is just met with 0.1%
% accuracy. The other one is less than the given value.
% dpp is the maximum deviation from omega in the
% passband from zero to wp, whereas dss is the maximum
% deviation from zero in the stopband from ws to unity.
% wp and ws are given as fractions of pi (half the
% sampling rate).
% If N is too low, the above is not possible.
% difwinad.m takes care of this and increases N.
%-----------------------------------------------------
% firtyp=1,2,3,4 for lowpass, highpass, bandpass,
% and bandstop filters
%-----------------------------------------------------
% The program returns both the filter coefficient h and
% the window function w. N is the length; the
% corresponding order is N-1.
%-----------------------------------------------------

%-----------------------------------------------------
% Tapio Saramäki  16.10.97
%-----------------------------------------------------
% This can be found in SUN's: ~ts/matlab/dsp/difsubad.m
%-----------------------------------------------------
dpp=ddp/(pi/(wp+ws)/2);
dp=dpp;
dss=dds/(pi/(wp+ws)/2);
ds=dss;
accuracy=1;
while accuracy == 1
```

```
  if itype==1
   [N1,alpha,wc]=kaiord(wp,ws,dp,ds);
   wind=rot90(kaiser(N,alpha));
  end
  if itype==2
   [N1,beta,wc]=sarord(wp,ws,dp,ds);
   %wind=saramaki(N,beta);
   wind=transit(N,beta,1); %faster
  end
  if itype==3
   [N1,beta,wc]=dchord(wp,ws,dp,ds);
   %wind=dcheb(N,beta);
   wind=transit(N,beta,0); %faster
  end
  if itype==4
   [N1,beta,rho,wc]=traord(wp,ws,dp,ds);
   wind=transit(N,beta,rho);
  end
  wc=(wp+ws)/2;
  h=firwind(N-1,wc,wind,'differentiator');
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%
%h=2*h;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%
  [H,W]=freqz(h,1,8*1024);
  na=round(ws*8*1024)+500/N;
  nb=round(wp*8*1024)-500/N;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%
  amaxa=max(abs(H(na:8*1024)));
  amaxb=max( abs( abs(W(1:nb))-abs(H(1:nb)) ) );
%  figure(1)
%  plot(W/pi,abs(H)/pi)
%  figure(2)
%  plot(W/pi,abs(W)-abs(H));pause
  amaxa=amaxa/(pi/(wp+ws)/2);
  amaxb=amaxb/(pi/(wp+ws)/2);
  nda=-20*log10(dss)+20*log10(amaxa);
  nda=-20*log10(ds)+nda;
  ds=10^(-nda/20);
  ndb=-20*log10(dpp)+20*log10(amaxb);
  ndb=-20*log10(dp)+ndb;
```

```
  dp=10^(-ndb/20);
  if abs((amaxa-dss)/dss) < .0001 & amaxb < dpp
    accuracy=0;
  end
  if (abs(amaxb-dpp)/dpp) < .0001 & amaxa < dss
    accuracy=0;
  end
  if dp < dpp/10 & ds < dss/10
    accuracy=0; % N is too low
  end
end
```

****