Appendix B: Design of Hilbert transformers using adjustable windows

```
%-------------------------------------------------------
% Matlab m-file (hilwinad.m)
% for designing an FIR Hilbert tranformer using the
% Kaiser, Saramaki, Dolph-Chebyshev, or transitional
% window.
%
% The filters are designed in such a way that the
% desired passband criteria are just met.
%
% 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 Hilbert')
disp('transformer')
disp('with the aid of the Kaiser, Saramaki,')
disp('Dolph-Chebyshev, or transitional window')
disp('As input data, I first need the filter type:')
firtyp=input('1 for even and 2 for odd order');
%-------------------------------------------------------
wp=input('Lower edge as a fraction of pi: ');
dp=input('Maximum deviation from unity');
disp('Type 1 for Kaiser, 2 for Saramaki, 3 for')
itype=input('for Dolph-Chebyshev, or 4 for transitional window: ');
%-------------------------------------------------------------
% Estimate the length N; even [odd] for odd [even] order.
%-------------------------------------------------------------
if itype==1
  [N,alpha,wc]=kaiord(wp,3*wp,dp/2,dp/2);
end
if itype==2
  [N,beta,wc]=sarord(wp,3*wp,dp/2,dp/2);
end
if itype==3
  [N,beta,wc]=dchord(wp,3*wp,dp/2,dp/2);
end
```

```
  [N,beta,rho,wc]=traord(wp,3*wp,dp/2,dp/2);
end
if firtyp==1 & rem(N-1,2)==1
  N=N+1;
end
if firtyp==2 & rem(N-1,2)==0
  N=N+1;
end
%------------------------------------------------------------
% Find the filter coefficients h and the window coefficients
% wind for the estimated length N such that either the passband
% criterion is just met. For a too low value of N,
% this is not true and N is increassed. N is the length.
%------------------------------------------------------------
[h,wind]=hilsubad(itype,N,wp,dp);
%-----------------------------------------------------
% Test whether the criterion is met at the lower edge.
%-----------------------------------------------------
isu=0;
[a,z]=zeroam(h,wp,wp,1);
if abs(1-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]=hilsubad(itype,N,wp,dp);
%-----------------------------------------------------
% Test whether the criterion is met at the lower edge.
%-----------------------------------------------------
    isu=0;
    [a,z]=zeroam(h,wp,wp,1);
    if abs(1-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]=hilsubad(itype,N,wp,dp);
%-------------------------------------------------------
% Test whether the criterion is met at the lower edge.
%-------------------------------------------------------
    isu=0;
    [a,z]=zeroam(h,wp,wp,1);
    if abs(1-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))));
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');
figure(2)
subplot(211)
[H,W]=zeroam(hs,.0,1.,8*1024);
plot(W/pi,(abs(H)));axis([0 1 0 1+dp]);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 firtyp==1
  x1=wp;x2=1-wp;
end
if firtyp==2
  x1=wp;x2=1;
end
plot(W/pi,H);axis([x1 x2 1-dp 1+dp]);
grid;
title('Passband 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]=hilsubad(itype,N,wp,ddp)
%-----------------------------------------------------
% 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 ripple criterion of the resulting Hilbert
% transformer is just met with 0.1 % accuracy.
% dpp is the maximum deviation from unity in the
% passband.
% If N is too low, the above is not possible.
% hilwinad.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/hilsubad.m
%-----------------------------------------------------
dpp=ddp/2;
dp=dpp;
```

```
accuracy=1;
while accuracy == 1
  if itype==1
    [N1,alpha,wc]=kaiord(wp,3*wp,dp,dp);
    wind=rot90(kaiser(N,alpha));
  end
  if itype==2
    [N1,beta,wc]=sarord(wp,3*wp,dp,dp);
    %wind=saramaki(N,beta);
    wind=transit(N,beta,1); %faster
  end
  if itype==3
    [N1,beta,wc]=dchord(wp,3*wp,dp,dp);
    %wind=dcheb(N,beta);
    wind=transit(N,beta,0); %faster
  end
  if itype==4
    [N1,beta,rho,wc]=traord(wp,3*wp,dp,dp);
    wind=transit(N,beta,rho);
  end
  h=firwind(N-1,1.,wind,'Hilbert');
  [H,W]=freqz(h,1,8*1024);
  na=round(wp*8*1024)+round(500/N);
  nb=length(H);
  if rem(N-1,2)==0
    nb=round((1-wp)*8*1024)-round(500/N);
  end
  amax=max(abs(1-abs(H(na:nb))));
  amax=amax/2;
  nd=-20*log10(dpp)+20*log10(amax);
  nd=-20*log10(dp)+nd;
  dp=10^(-nd/20);
  if abs((amax-dpp)/dpp) < .0001
    accuracy=0;
  end
  if dp < dpp/10
    accuracy=0; % N is too low
  end
end
*********************************************************************************************
```