Appendix G: Program for designing with the aid of linear programming a filter considered in the lecture notes as well as in the handbook chapter

```
****************************************************************************************
% This Matlab script-file optimizes the coefficients of
% of Example 4.10 in the handbook chapter T. Saramäki,
% "Finite Impulse Response Filter Design".
%
% Can be found in SUNs: ~ts/matlab/dsp/linear1.m
%
% Subfunctions needed: type1fre() (home-made)
%                      LP() (optimization toolbox)
%
% Copyright 1998 by Tapio Saramäki,
% Tampere University of Technology.

clear all
close all
NN=70 % Filter order
%------------------------------------------------------------
% Half the filter order+1 (number of unknown filter coefficients)
%------------------------------------------------------------
N=NN/2+1;
%-------------------------------
% Grid points
%--------------------------------
gridp=150;
grids1=200;
grids2=400;
Wfp=[0:0.15*pi/(gridp-1):0.15*pi];
Wap=[0.15*pi:0.15*pi/(gridp-1):0.3*pi];
Wfs=[0.4*pi:0.2*pi/(grids1-1):0.6*pi];
Was=[0.6*pi:0.4*pi/(grids2-1):pi];
%----------------------------------------------------
% Desired response in the passband and stopband
%----------------------------------------------------
Desfp=ones(1,gridp);
Desap=ones(1,gridp);
Desfs=zeros(1,grids1);
Desas=zeros(1,grids2);
%----------------------------------------------------
% Passband and stopband weights
%----------------------------------------------------
Weifp=ones(1,gridp);
Weiap=ones(1,gridp);
```

```
Weifs=20*ones(1,grids1);
Weias=10*ones(1,grids2);
%******************************************************
% Set up matrises for Linear Programming:
% min z=c'x subject to Ax<=b
% Note that a linear combination of unknowns is
% minimized. In the lecture notes this combination
% was maximized!!
%******************************************************
% For our problem, x(i)=h(NN-i+1) for i=1,2,...,N, where
% N=NN/2+1 with NN being the filter order. x(N+1) is delta_1,
% the maximum absolute value of the weighted error function
% E(omega)=W(omega)[H(omega)-D(omega)], where H(omega)=
% x(1)*Phi(1,Omega)+...+x(N)*Phi(N,omega) with
% Phi(1,Omega)=1 and Phi(i,Omega)= 2*cos((i-1)*omega)
% for i > 1.
%******************************************************
%
%***********************************************************************
%Frequency-domain conditions: Adjustable part
%************************************************************************
% H(omega(i))-delta_1/W(omega(i)) <= D((omega(i))
% and
% -H(omega(i))-delta_1/W(omega(i)) <= -D((omega(i))
%*************************************************************************
Wa=[Wap Was];
for i=1:N
        Ga(:,i)=type1fre(i,Wa)';
end
A1=-1./[Weiap Weias]';
A=[ Ga  A1
  -Ga  A1];
b=[Desap Desas -Desap -Desas]';
%*******************************************
% To be minimized is x(N+1)
%*******************************************
c=[zeros(1,N) 1];
%***********************************************************************
%Frequency-domain conditions: Fixed part
%************************************************************************
% H(omega(i)) <= D((omega(i))+delta_2/W(omega(i))
% and
% -H(omega(i)) <= -D((omega(i))+delta_2/W(omega(i))
%*************************************************************************
Wf=[Wfp Wfs];
for i=1:N
```

```
        Gf(:,i)=type1fre(i,Wf)';
end
A1=zeros(size(Wf))';
Af=[ Gf  A1
  -Gf  A1];
WWp=.002*ones(size(Desfp))./Weifp;
WWs=.002*ones(size(Desfs))./Weifs;
bf=[Desfp+WWp Desfs+WWs -Desfp+WWp -Desfs+WWs]';
b=[b;bf];
A=[A;Af];
%******************************************************************
% Zeros at 0.4pi 0.45pi 0.5pi 0.55pi 0.6pi 0.65pi
%****************************************************
kk=1:1:N-1;
for ll=40:5:65
 wz=ll*pi/100;
 clear A1
 clear b1
 clear A2
 clear b2
 b1=[0]';
 b2=[0]';
 A1=[ones(1,1) 2*cos(kk*wz) 0];
 A2=[-ones(1,1) -2*cos(kk*wz) 0];
 b=[b;b1;b2];
 A=[A;A1;A2];
end
%***********************************************************
% H(omega)=1 and the first four derivatives are zero at
% omega=0
% These are satisfied by
% 2*(x(N)+x(N-1)+...+x(2))+1=1
% (N-1)^2*x(N)+(N-2)^2*x(N-1)+...+1^2*x(2)=0
% (N-1)^2*x(N)+(N-2)^2*x(N-1)+...+1^2*x(2)=0
%************************************************************
clear A1
clear b1
clear A2
clear b2
b1=[1]';
b2=[-1]';
A1=[ones(1,1) 2*ones(1,N-1) 0];
A2=[-ones(1,1) -2*ones(1,N-1) 0];
b=[b;b1;b2];
A=[A;A1;A2];
kk2=kk.*kk;
```

```
kk4=kk2.*kk2;
clear A1
clear b1
clear A2
clear b2
b1=[0]';
b2=[0]';
A1=[0 kk2 0];
A2=[0 -kk2 0];
b=[b;b1;b2];
A=[A;A1;A2];
clear A1
clear b1
clear A2
clear b2
b1=[0]';
b2=[-0]';
A1=[0 kk4 0];
A2=[0 -kk4 0];
b=[b;b1;b2];
A=[A;A1;A2];
%*************************************************
% Call LP function for Linear Programming
% This function is included in Optimization Toolbox
%*************************************************
x=lp(c,A,b);
%------------------------------------------
% h(NN-i+1)=x(i) for i=1,2,...N (N=NN/2+1)
% x(N+1) is the stopband ripple.
%------------------------------------------
%*************************************************
% Form the impulse response of the filter
%*************************************************
B=x(1:N)';
h=[fliplr(x(2:N)') B];
ripple=x(N+1)
[H,z]=zeroam(h,.0,1.,5000);
M=2*(N-1);
figure(1)
plot(z/pi,20*log10(abs(H)));axis([0 1 -130 10]);grid
xlabel('Angular frequency omega/pi');
ylabel('Amplitude in dB');
[H,z]=zeroam(h,.0,.3,5000);
figure(2)
dmin=1-min(H);dmax=max(H)-1;
plot(z/pi,H);axis([0 .3 1-1.05*dmin 1+1.05*dmax]);grid
```

```
title('Passband details')
xlabel('Angular frequency omega/pi');
ylabel('Amplitude');
******************************************************************
function ares=type1fre(i,W)
% **********************************************************
% ares=type1fre(i,W)
% Calculates 2*cos((i-1)*W) for i>1 and
% ones(size(W)) for i=1.

%  Written by: Tapio Saramaki

% Copyright 1998 by Tapio Saramäki,
% Tampere University of Technology.
%************************************************************
if i>1
  ares=2*cos((i-1)*W);
else
  ares=ones(size(W));
end
```