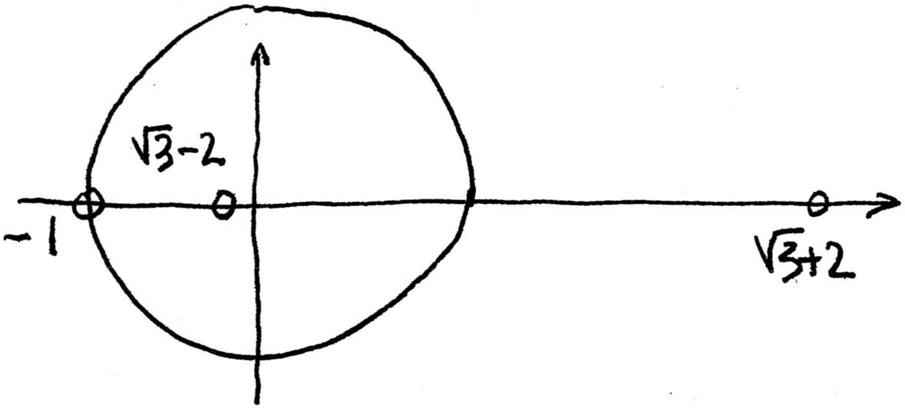
Homework 6 solution.

Theory

1)

$$H_0(z) = -\frac{1}{8} z^{-2} (1+z)^2 (z+\sqrt{3}-2)(z-\sqrt{3}-2).$$



$H_0(-1) = 0$, in fact -1 is a zero of order 2

$$H_0(1) = -\frac{1}{8} + \frac{1}{4} + \frac{3}{4} + \frac{1}{4} - \frac{1}{8} = 1.$$

So H_0 is <u>a low pass filter.</u>

2) $$H_0(e^{jw}) = \frac{3}{4} + \frac{1}{2}\cos w - \frac{1}{4}\cos 2w$$

$$= 1 + \frac{1}{2}\cos w - \frac{1}{2}\cos^2 w$$

$$\boxed{H_0(e^{jw}) = 1 + \frac{1}{2}\cos(w)\,(1 - \cos(w))}$$

The phase is constant and equal to 0.

3) See plot attached.

41 $H_0(z)F_0(z) =$

$$= -\frac{1}{8} a z^{-3} + \frac{1}{4} a z^{-2} + \frac{3}{4} a z^{-1} + \frac{1}{4} a - \frac{1}{8} a z$$

$$- \frac{1}{8} b z^{-2} + \frac{1}{4} b z^{-1} + \frac{3}{4} b + \frac{1}{4} b z - \frac{1}{8} b z^{2}$$

$$- \frac{1}{8} a z^{-1} + \frac{1}{4} a + \frac{3}{4} a z + \frac{1}{4} a z^{2}$$

$$- \frac{1}{8} b z^{3}$$

Similarly,

$H_0(-z)F_0(-z) =$

$$\frac{1}{8} a z^{-3} + \frac{1}{4} a z^{-2} - \frac{3}{4} a z^{-1} + \frac{1}{4} a + \frac{1}{8} a z$$

$$- \frac{1}{8} b z^{-2} - \frac{1}{4} b z^{-1} + \frac{3}{4} b - \frac{b}{4} z - \frac{1}{8} b z^{2}$$

$$+ \frac{1}{8} a z^{-1} + \frac{1}{4} a - \frac{3}{4} a z^{+1} + \frac{1}{4} a z^{2}$$

$$+ \frac{1}{8} b z^{3}$$

So $H_0(z)F_0(z) + H_0(-z)F_0(-z) =$

$$\left(\frac{a}{2} - \frac{b}{4}\right) z^{-2} + \frac{1}{2}(2a + 3b) + \left(\frac{a}{2} - \frac{b}{4}\right) z^{2}$$

If we want this expression to be equal to 2 we need to have $b = 2a$

and $\frac{1}{2}(2a + 6a) = 4a = 2$ or $\boxed{a = \frac{1}{2}, b = 1}$

5) $F_0(z) = \frac{1}{2}(z^{-1} + 2 + z) = \frac{z^{-1}}{2}(z^2 + 2z + 1)$

$F_0(z) = \frac{z^{-1}}{2}(z+1)^2 = \frac{z}{2}(1 + z^{-1})^2$

$F_0(-1) = 0$ $\quad -1$ is a zero of order 2

$F_0(1) = 2$

F_0 is a <u>lowpass filter.</u>

6) $F_0(e^{j\omega}) = \frac{1}{2}\left\{ e^{-j\omega} + e^{j\omega} + 2 \right\} = \frac{1}{2}\left\{ 2\cos(\omega) + 2 \right\}.$

$F_0(e^{j\omega}) = \cos(\omega) + 1 = \boxed{2\cos^2(\omega/2) = F_0(e^{j\omega})}$

F_0 has linear phase (constant $= 0$).

7) $F_1(z) = z^{-1} \sum h_0[n](-z)^{-n}$

$= \sum_n (-1)^n h_0[n] \, z^{-1} z^{-n}$

$= \sum_m (-1)^{m-1} h_0[m-1] \, z^{-m}$ $\qquad m = n+1$

so $\boxed{f_1[n] = (-1)^{n-1} h_0[n-1]}$

similarly $\boxed{h_1[n] = (-1)^{n+1} f_0[n+1]}$

8) We have

$$Y(z) = \frac{1}{2}\{F_0(z)H_0(z) + F_1(z)H_1(z)\}X(z)$$

$$+ \frac{1}{2}\{F_0(z)H_0(-z) + F_1(z)H_1(-z)\}X(-z)$$

$$= \frac{1}{2}\{F_0(z)H_0(z) + H_0(-z)F_0(-z)\}X(z)$$

$$+ \frac{1}{2}\{F_0(z)H_0(-z) - H_0(-z)F_0(z)\}X(-z)$$

(from (4))

$$= X(z). \quad \text{(from (3))}.$$

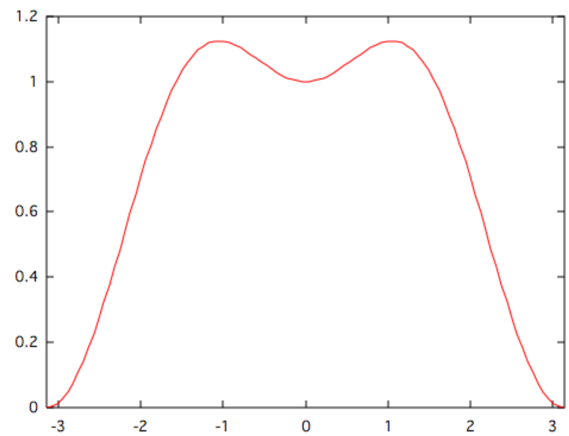


Figure 1: Frequency response of F_0.

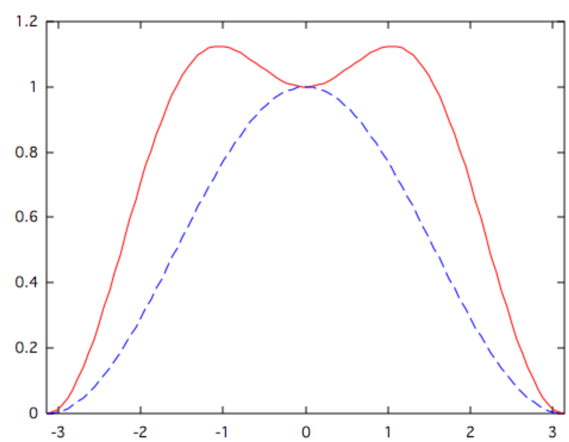


Figure 2: Frequency response of H_0 (dashed), and F_0 (solid).

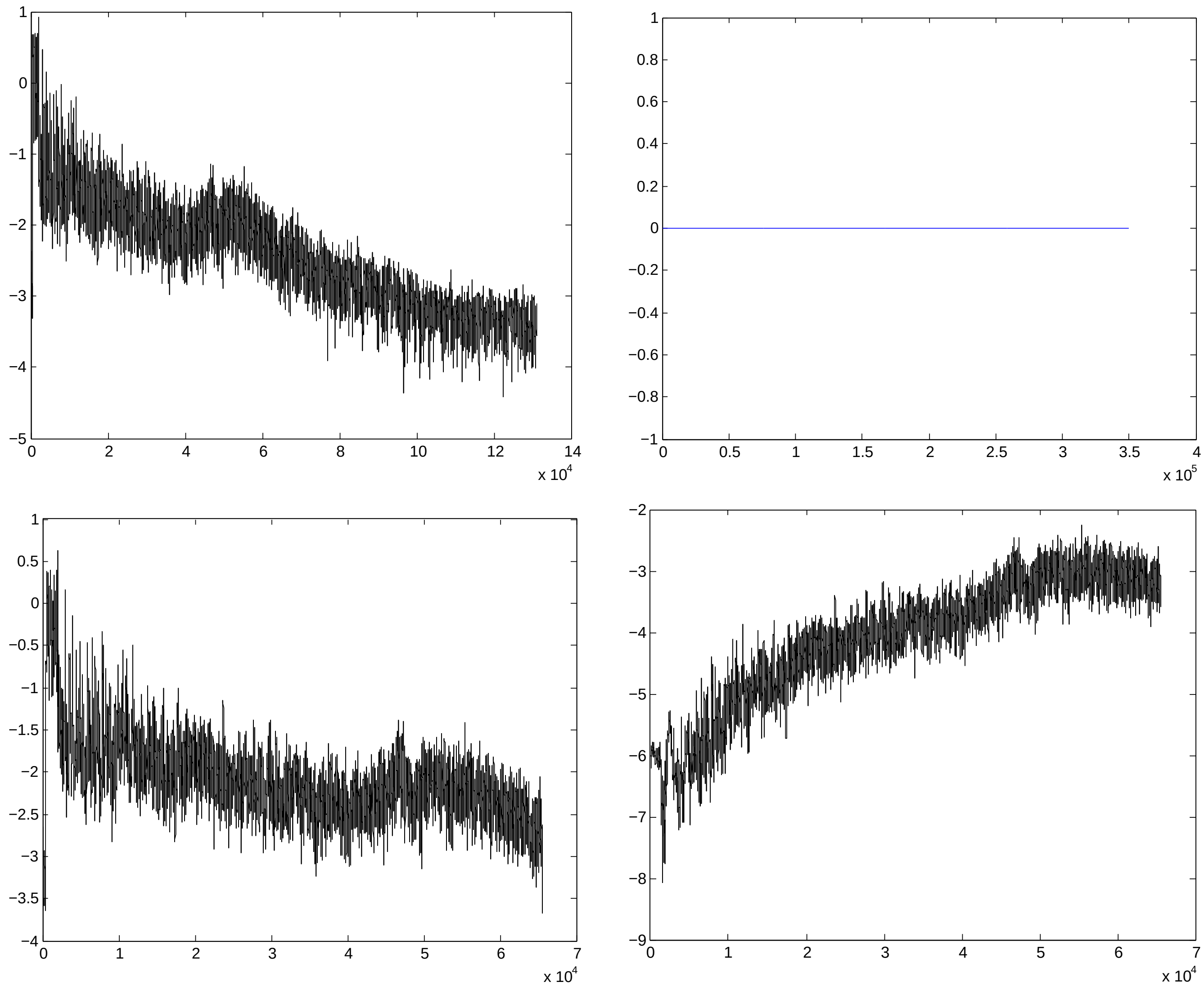


Figure 3: Top: left: power spectrum ($|X(e^{j\omega})|^2$) of `beat.wav`; right: reconstruction error of `beat.wav`. Bottom: left: power spectrum of the lowpass filtered (with H_0) and downsampled signal (notice the reduce frequency range); right: power spectrum of the highpass filtered (with H_1) and downsampled signal.

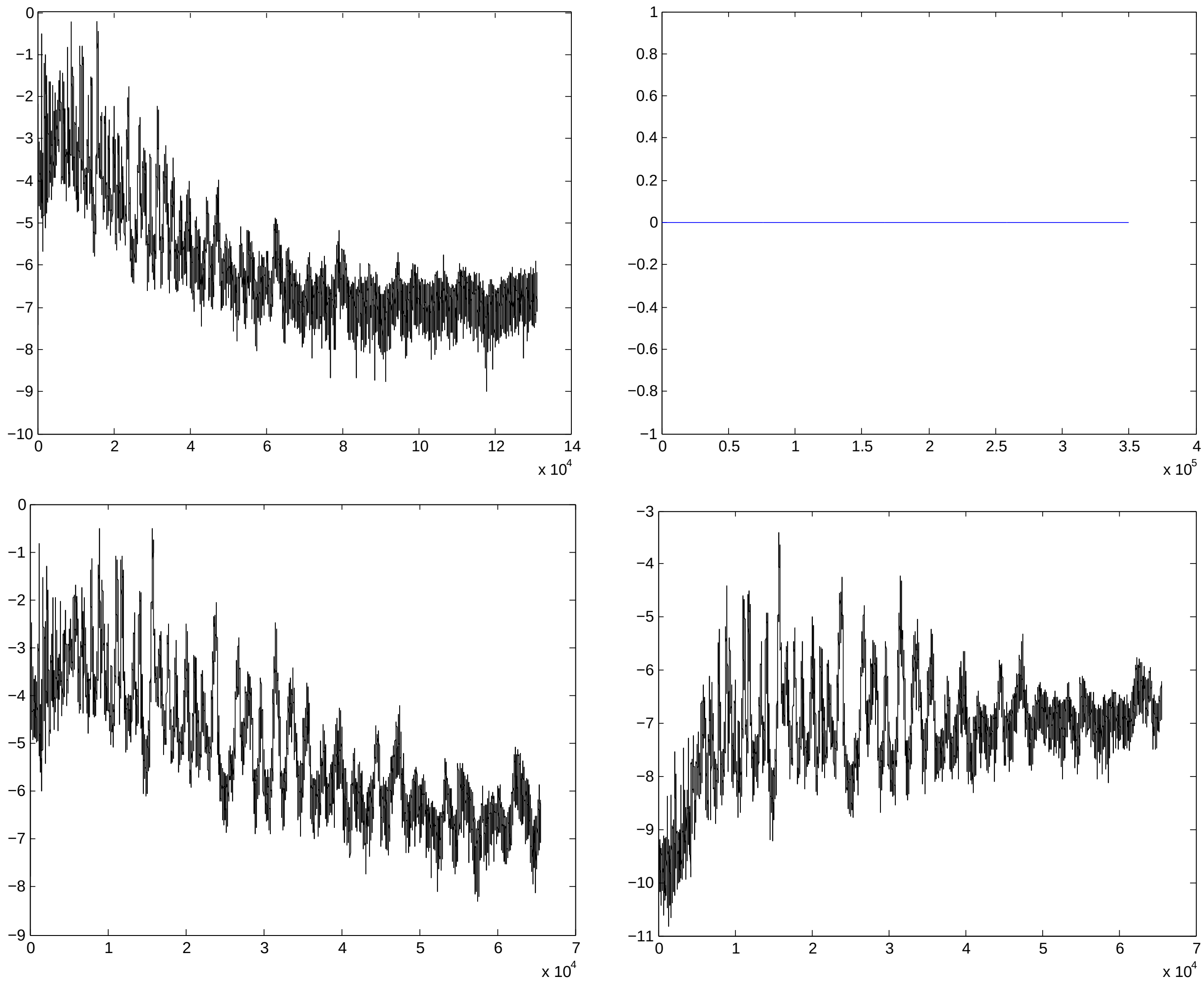


Figure 4: Top: left: power spectrum ($|X(e^{j\omega})|^2$) of `flute.wav`; right: reconstruction error of `flute.wav`. Bottom: left: power spectrum of the lowpass filtered (with H_0) and downsampled signal (notice the reduce frequency range); right: power spectrum of the highpass filtered (with H_1) and downsampled signal. Note also the monochromatic nature of the spectrum: only the main frequency and a few harmonics are present (compare to the corresponding plots for `beat.wav`

```
%
% solution for homework 11.
%
% Francois Meyer, April 2011

[w,fs,nb]=wavread('beat.wav');
w=w';


f0=[-1 2 6 2 -1]./8;
f1=[-1 2 -1];

h0=[1 2 1]./2;
h1=[-1 -2 6 -2 -1]./16;

lFilter = max(length(f0), max(length(f1), max(length(h0), length(h1))));

%
% symmetric extension
%

lW    = length (w);
Left  = fliplr (w(2:lFilter+1));
Right = fliplr (w(lW -lFilter +1:lW));

x  = [Left w Right];
lX = length (x);

clear w;
clear Left;
clear Right;

%
% convolution decimation
%

 v0 = filtdec (x,f0);
 v1 = filtdec (x,f1);

 %
%  try me if you want to do convolution WITHOUT decimation
%
```

```
%
% v0 = conv (x,h0);
% v1 = conv (x,h1);
%

% sound (x,  fs);

 slowrate=fs/2;

% sound (v0, slowrate);
% sound (v1, slowrate);



%
% Fourier transform
%
% X= abs(fft(x)).^2;

X= pmtm (x,4);

figure;plot (log10(X(1:floor(length(X)/2))),'k-');


%
% Fourier transform
%
% V0= abs(fft(v0)).^2;
% V1= abs(fft(v1)).^2;

V0= pmtm (v0,4);
V1= pmtm (v1,4);

figure; plot (log10(V0(1:floor((length(V0) + 1)/2))),'k-');
figure; plot (log10(V1(1:floor((length(V1) + 1)/2))),'k-');

%
% upsampling and convolution
%

y0 = upfilt (v0,h0);
y1 = upfilt (v1,h1);
```

```
ly = max(length(y0), length(y1));

%
% reconstructed signal
%

y = [y0 zeros(1:ly - length(y0))] + [y1 zeros(1:ly - length(y1))];

%
% compute the difference between the original signal and the reconstructed
% one. Because we did some extension, we need to extract the original part of
% of the signal.

error = x(lFilter+1:lW+lFilter)-y(lFilter+4:lW+lFilter+3);
max(error)                                                                max
figure; plot (error);

%
% plot the reconstructed signal on top of the original signal
%

figure ;plot (x(lFilter+1:lW+lFilter),'b-');
figure ;plot (y(lFilter+4:lW+lFilter+3),'r-');

keyboard

return;
```

```
function y = filtdec(x,h);

% Convolution and decimation
% polyphase implementation


lh = length(h);
lp = floor((lh-1)/2) + 1;

p  = reshape([reshape(h,1,lh),zeros(1,lp*2-lh)],2,lp);
lx = length(x);
ly = floor((lx+lh-2)/2) + 1;
lu = floor((lx)/2) + 1; % length of decimated sequences

u = [zeros(1,2-1),reshape(x,1,lx),zeros(1,2*lu -lx -1)];
u = flipud(reshape(u,2,lu)); % the decimated sequences

y = zeros(1,lu+lp-1);

for m = 1:2
  y = y + conv(u(m,:),p(m,:));
end

y = y(1:ly);
```

```
function y = upfilt (x,h);

% expansion and convolution
% polyphase implementation

lh = length(h);
lq = floor((lh-1)/2) + 1;
q  = flipud(reshape([reshape(h,1,lh),zeros(1,lq*2-lh)],2,lq));
lx = length(x);
ly = lx*2 + lh -1;
lv = lx + lq;
v  = zeros(2,lv);

for l = 1:2
   v(l,1:lv-1) = conv(x,q(l,:));
end

y = reshape(flipud(v),1,2*lv);
y = y(1:ly);
```

```
function y = ppint(x,h,L);
% USAGE y = ppint(x,h,M).
% L-fold expansion and convolution, by polyphase decomposition.
% Input parameters:
% x: the input sequence
% h: the FIR filter coefficients
% L: the expansion factor.
% Output parameters:
% y: the output sequence.

lh = length(h); lq = floor((lh-1)/L) + 1;
q = flipud(reshape([reshape(h,1,lh),zeros(1,lq*L-lh)],L,lq));
lx = length(x); ly = lx*L+lh-1;
lv =lx + lq; % length of decimated sequences
v = zeros(L,lv);
for l = 1:L, v(l,1:lv-1) = conv(x,q(l,:)); end
y = reshape(flipud(v),1,L*lv);
y = y(1:ly);
```