ENGR 451 - Lab 6 FIR Filters Part II

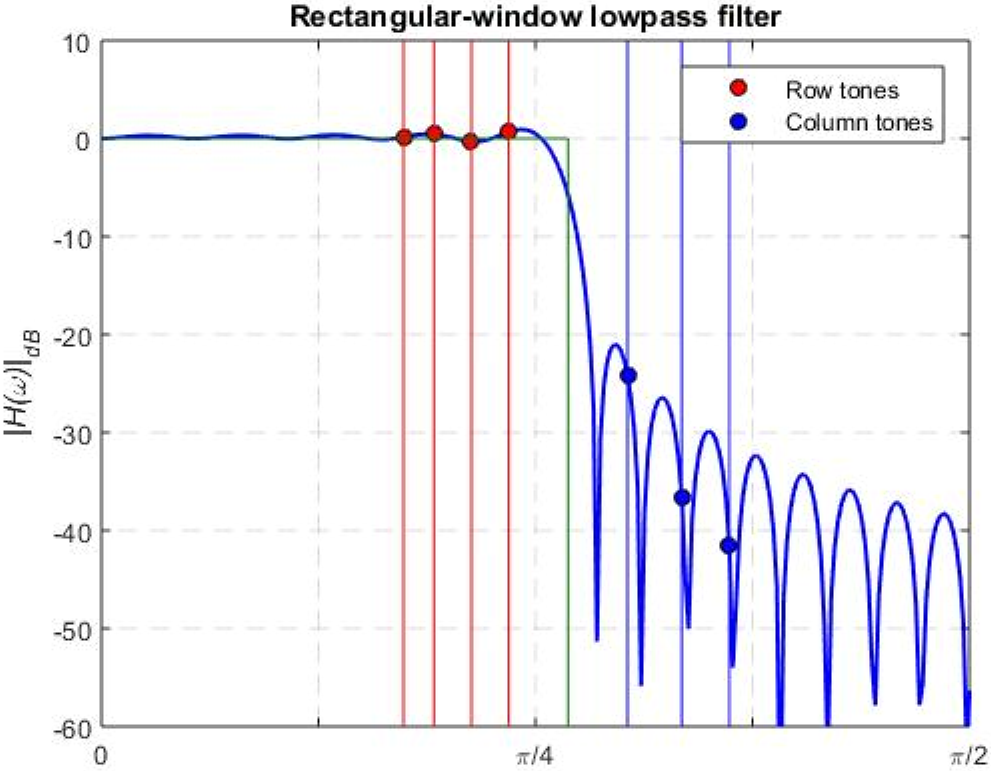
Table of Contents

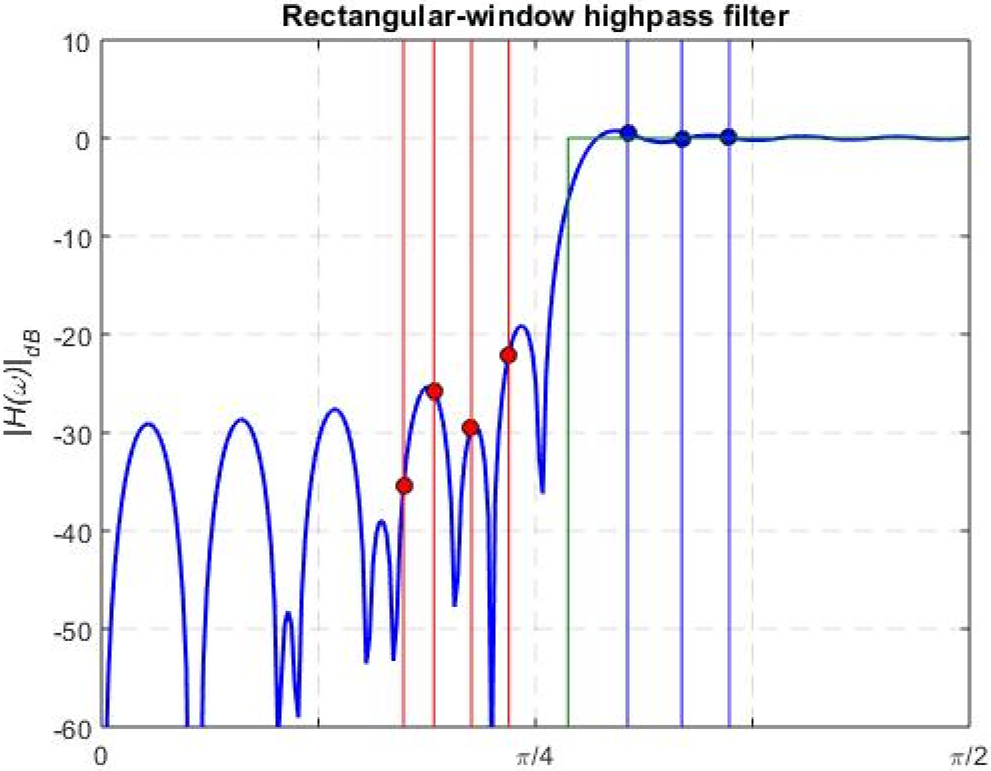
by Steven Bao & Zizhe Ren (Mar 2016)

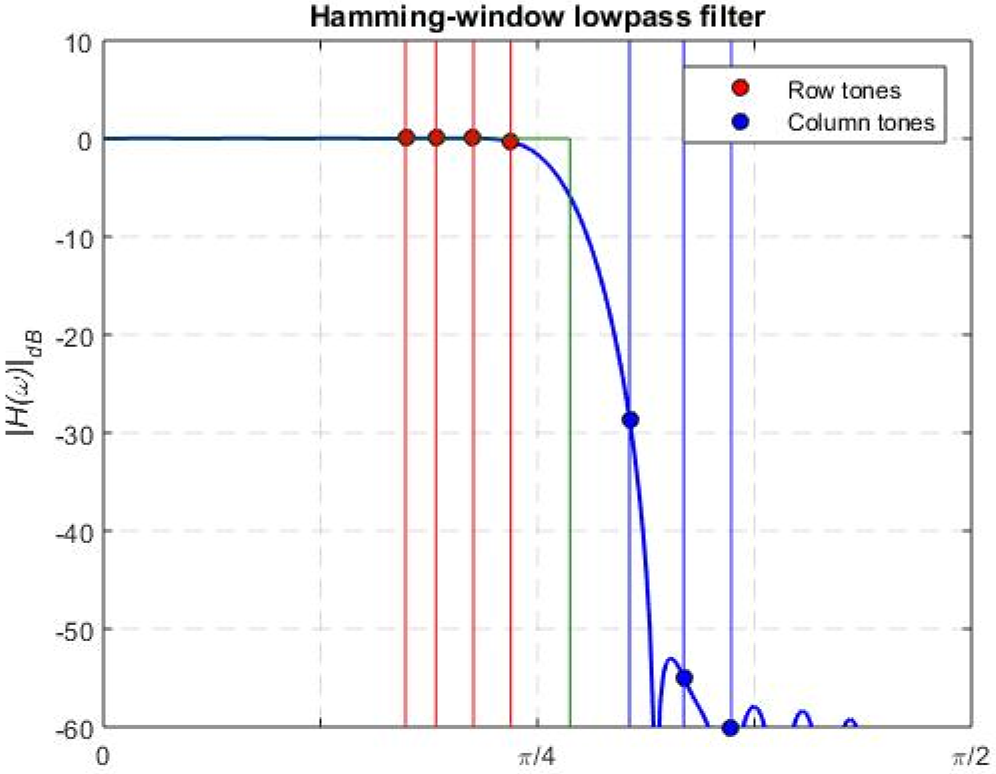
Part 1: Filter codes

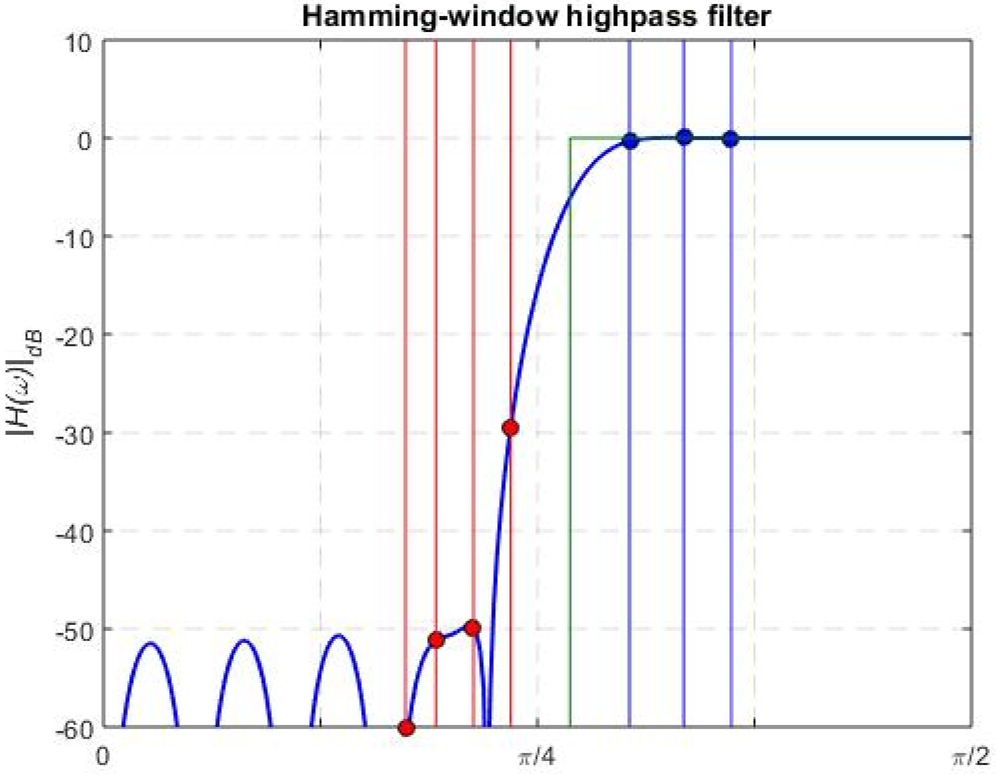
```
% ---------- Rectangular-window Filter ---------- %
[s,fs]=audioread('phonetones.wav'); % read the wave from file
N = 73; % set the length of filter
M = (N-1)/2;
wc = 2 * (1075 / fs); % set the cut off frequency
wtones = 2 *[697 770 852 941 1209 1339 1447] / 8000;
hlp = fir1(N-1, wc, rectwin(N));
figure;
p = magdb_lab6(hlp);
title('Rectangular-window lowpass filter');
hold on; dots = sevendots(p);
legend([dots(1) dots(2)], 'Row tones', 'Column tones');
plot([0 wc], [0 0], [wc wc], [0 -60], 'color', [0,0.5,0]);
hhp = - hlp;
hhp(M+1) = hhp(M+1) + 1;
figure;
p = magdb_lab6(hhp);
title('Rectangular-window highpass filter');
hold on; sevendots(p);
plot([wc 0.5], [0 0], [wc wc], [0 -60], 'color', [0,0.5,0]);

% ---------- Hamming-window Filter ---------- %
hlp = fir1(N-1, wc, hamming(N));
figure;
p = magdb_lab6(hlp);
title('Hamming-window lowpass filter');
hold on; dots = sevendots(p);
legend([dots(1) dots(2)], 'Row tones', 'Column tones');
plot([0 wc], [0 0], [wc wc], [0 -60], 'color', [0,0.5,0]);
hhp = - hlp;
hhp(M+1) = hhp(M+1) + 1;
figure;
p = magdb_lab6(hhp);
title('Hamming-window highpass filter');
hold on; sevendots(p);
plot([wc 0.5], [0 0], [wc wc], [0 -60], 'color', [0,0.5,0]);

% ---------- Kaiser-window Filter ---------- %
dw = 0.9 * (wtones(5) - wtones(4));
[~, beta] = kaiserparams(dw, 0.01);
```

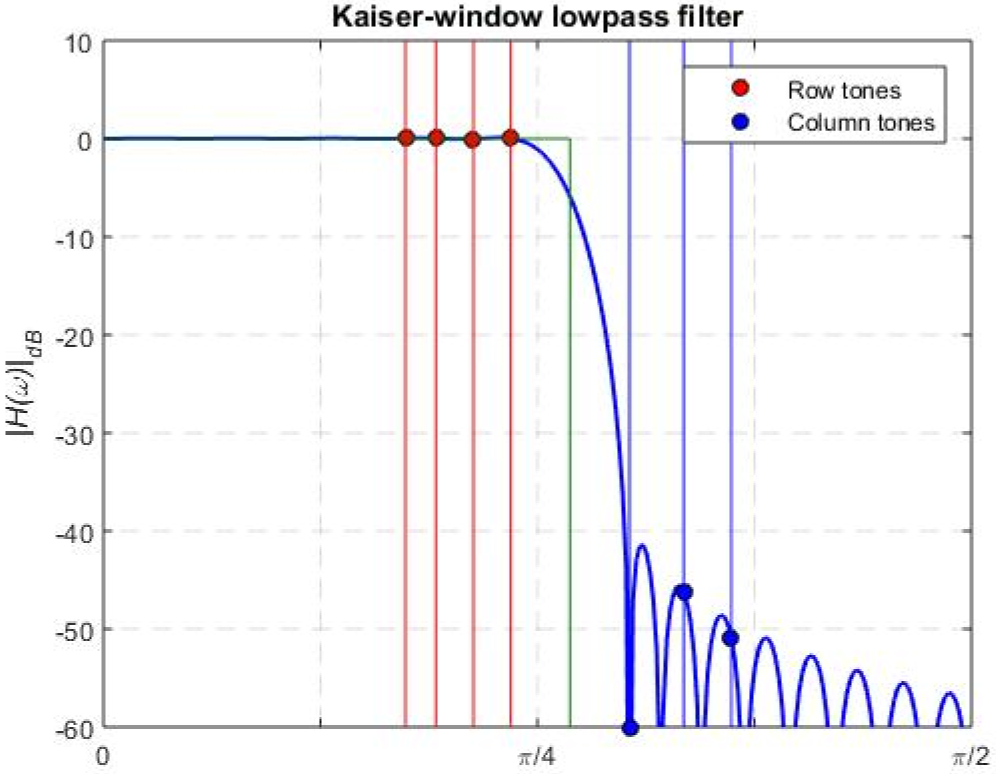
```
hlp = fir1 (N-1, wc, kaiser(N, beta));
figure;
p = magdb_lab6(hlp);
title('Kaiser-window lowpass filter');
hold on; dots = sevendots(p);
legend([dots(1) dots(2)], 'Row tones', 'Column tones');
plot([0 wc], [0 0], [wc wc], [0 -60], 'color', [0,0.5,0]);
hhp = - hlp;
hhp(M+1) = hhp(M+1) + 1;
figure;
p = magdb_lab6(hhp);
title('Kaiser-window highpass filter');
hold on; sevendots(p);
plot([wc 0.5], [0 0], [wc wc], [0 -60], 'color', [0,0.5,0]);

% ---------- Spline Filter ---------- %
dw = 0.77*(wtones(5)- wtones(4));
n = -M:M;
h0 = wc * sinc(wc * n);
r1 = sinc((dw/2) * n);
hlp = h0.* r1;
figure;
p = magdb_lab6(hlp);
title('Spline lowpass filter');
hold on; dots = sevendots(p);
legend([dots(1) dots(2)], 'Row tones', 'Column tones');
plot([0 wc], [0 0], [wc wc], [0 -60], 'color', [0,0.5,0]);
hhp = - hlp;
hhp(M+1) = hhp(M+1) + 1;
figure;
p = magdb_lab6(hhp);
title('Spline highpass filter');
hold on; sevendots(p);
plot([wc 0.5], [0 0], [wc wc], [0 -60], 'color', [0,0.5,0]);

% ----- Uniform spacing Frequency-sampled filter ----- %
k = 0:M;
w = 2 * k / N;
Ak = double(w < wc)';
C = cos((w * pi)'* k);
b = C \ Ak;
hlp = [0.5*b(end:-1:2); b(1); 0.5*b(2:end)];
figure; p = magdb_lab6(hlp);
title('Uniform spacing Frequency-sampled lowpass filter');
hold on;
dots = sevendots(p);
legend([dots(1) dots(2)], 'Row tones', 'Column tones');
x = get(p, 'xdata');
y = get(p, 'ydata');
index = zeros(1, 37);
xx = zeros(1, 37);
yx = zeros(1, 37);
for i=1:37
    [~, index(i)] = min(abs(w(i) - x));
    xx(i) = x(index(i));
    yx(i) = y(index(i));
    if yx(i) < -60
        yx(i) = -60;
    end
end
plot(xx,yx, 'X','color', 'k', 'LineWidth', 1)
plot([0 wc], [0 0], [wc wc], [0 -60], 'color', [0,0.5,0]);
hhp = - hlp;
hhp(M+1) = hhp(M+1) + 1;
figure;
p = magdb_lab6(hhp);
title('Uniform spacing Frequency-sampled highpass filter');
hold on; sevendots(p);
x = get(p, 'xdata');
y = get(p, 'ydata');
index = zeros(1, 37);
xx = zeros(1, 37);
yx = zeros(1, 37);
for i=1:37
    [~, index(i)] = min(abs(w(i) - x));
    xx(i) = x(index(i));
    yx(i) = y(index(i));
    if yx(i) < -60
        yx(i) = -60;
    end
end
plot(xx,yx, 'X','color', 'k', 'LineWidth', 1)
plot([wc 0.5], [0 0], [wc wc], [0 -60], 'color', [0,0.5,0]);

% ----- Non-uniform spacing Frequency-sampled filter ----- %
k = 0:M;
w = 2 * k / N;
index = zeros(1, 7);
for i=1:7
    [~, index(i)] = min(abs(wtones(i) - w));
    w(index(i)) = wtones(i);
end
Ak = double(w < wc)';
C = cos((w * pi)'* k);
b = C \ Ak;
hlp = [0.5*b(end:-1:2); b(1); 0.5*b(2:end)];
figure;
p = magdb_lab6(hlp);
title('Non-uniform spacing Frequency-sampled lowpass filter');
hold on; dots = sevendots(p);
legend([dots(1) dots(2)], 'Row tones', 'Column tones');
x = get(p, 'xdata');
y = get(p, 'ydata');
index = zeros(1, 37);
xx = zeros(1, 37);
yx = zeros(1, 37);
for i=1:37
    [~, index(i)] = min(abs(w(i) - x));
    xx(i) = x(index(i));
    yx(i) = y(index(i));
    if yx(i) < -60
        yx(i) = -60;
    end
end
plot(xx,yx, 'X','color', 'k', 'LineWidth', 1)
plot([0 wc], [0 0], [wc wc], [0 -60], 'color', [0,0.5,0]);
hhp = - hlp;
hhp(M+1) = hhp(M+1) + 1;
figure;
p = magdb_lab6(hhp);
title('Non-uniform spacing Frequency-sampled highpass filter');
hold on; sevendots(p);
x = get(p, 'xdata');
y = get(p, 'ydata');
index = zeros(1, 37);
xx = zeros(1, 37);
yx = zeros(1, 37);
for i=1:37
    [~, index(i)] = min(abs(w(i) - x));
    xx(i) = x(index(i));
    yx(i) = y(index(i));
    if yx(i) < -60
        yx(i) = -60;
    end
end
plot(xx,yx, 'X','color', 'k', 'LineWidth', 1)
plot([wc 0.5], [0 0], [wc wc], [0 -60], 'color', [0,0.5,0]);

% ---------- Optimum Filter ---------- %
hlp = firpm(N-1, [0 wtones([4 5]) 1], [1 1 0 0]);
figure;
p = magdb_lab6(hlp);
title('Optimum lowpass filter');
hold on; dots = sevendots(p);
legend([dots(1) dots(2)], 'Row tones', 'Column tones');
plot([0 wc], [0 0], [wc wc], [0 -60], 'color', [0,0.5,0]);
hhp = - hlp;
hhp(M+1) = hhp(M+1) + 1;
figure;
p = magdb_lab6(hhp);
title('Optimum highpass filter');
hold on;sevendots(p);
plot([wc 0.5], [0 0], [wc wc], [0 -60], 'color', [0,0.5,0]);
```

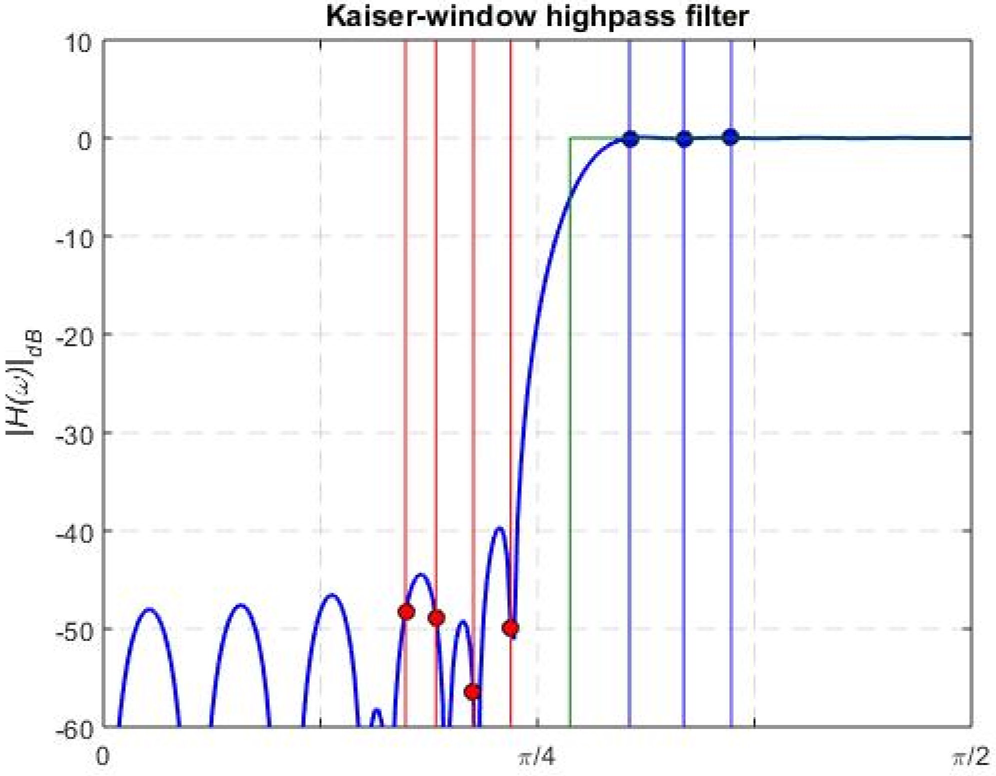

Rectangular-window lowpass filter
10
0
-10
-20
-30
-40
-50
-60
|H(ω)|_dB
0
π/4
π/2
● Row tones
● Column tones

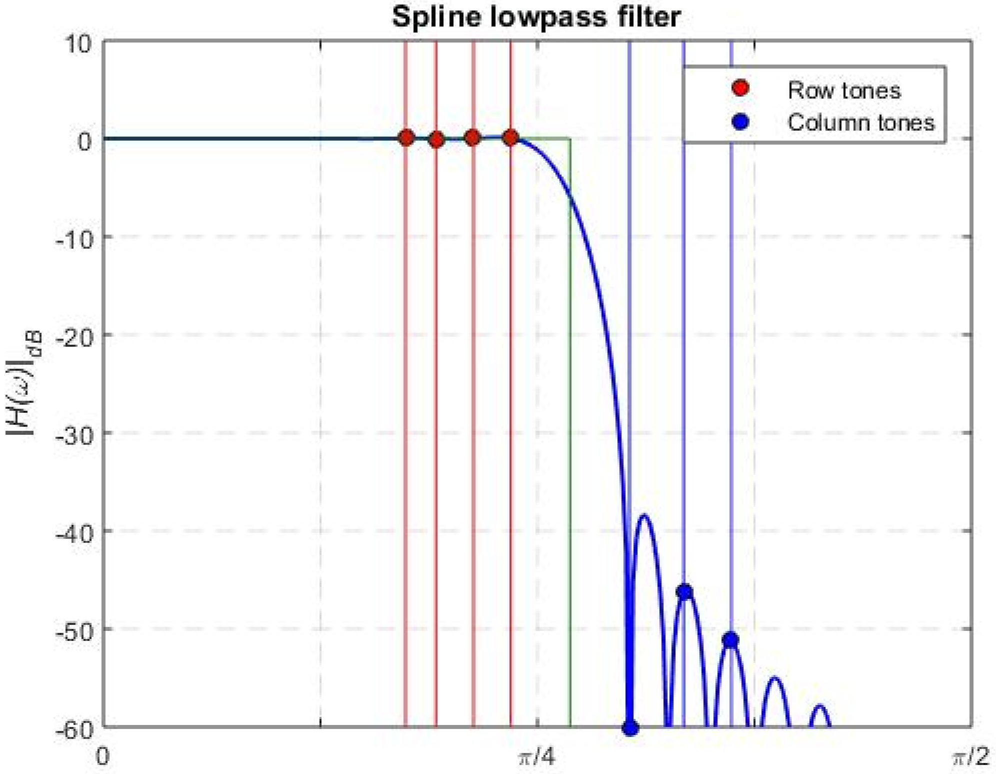

Rectangular-window highpass filter
10
0
-10
-20
-30
-40
-50
-60
|H(ω)|_dB
0
π/4
π/2

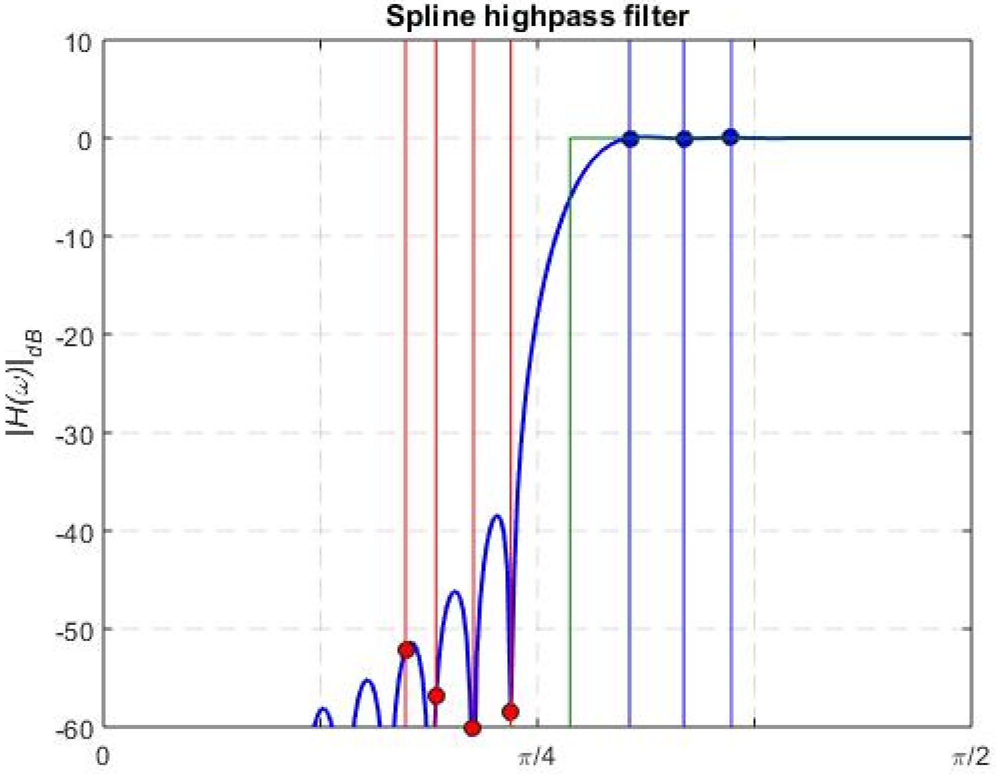

Hamming-window lowpass filter
Row tones
Column tones
10
0
-10
-20
-30
-40
-50
-60
$|H(\omega)|_{dB}$
0
$\pi/4$
$\pi/2$

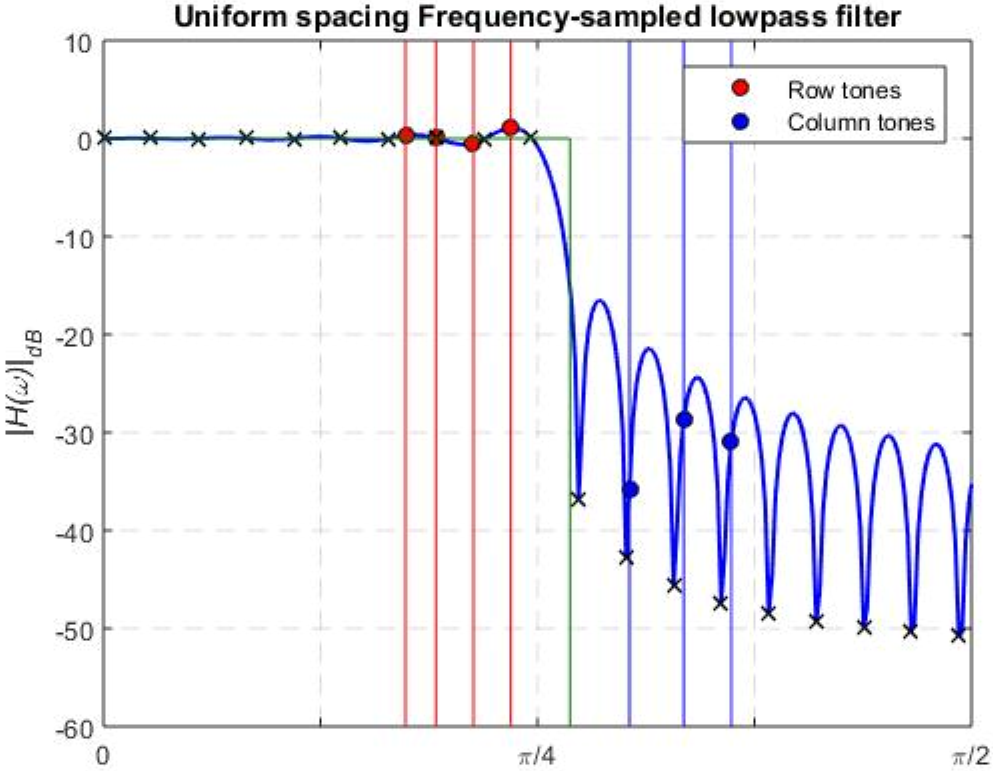

Hamming-window highpass filter
10
0
-10
-20
-30
-40
-50
-60
$|H(\omega)|_{dB}$
0
$\pi/4$
$\pi/2$

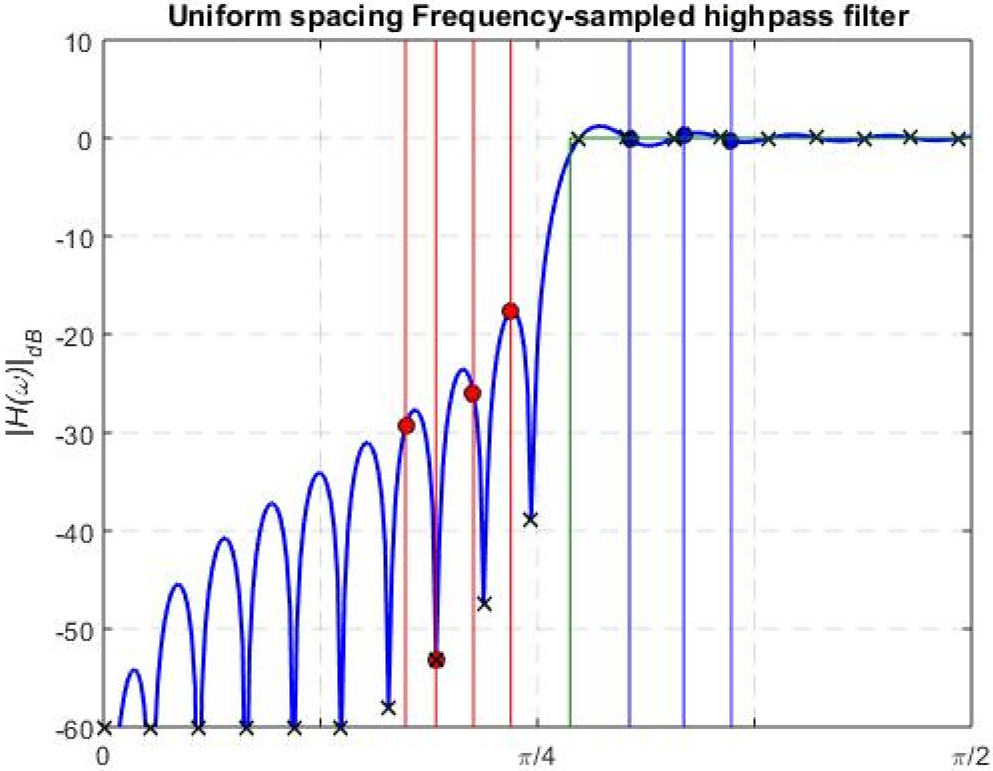

Kaiser-window lowpass filter
● Row tones
● Column tones
$|H(\omega)|_{dB}$
10
0
-10
-20
-30
-40
-50
-60
0
π/4
π/2

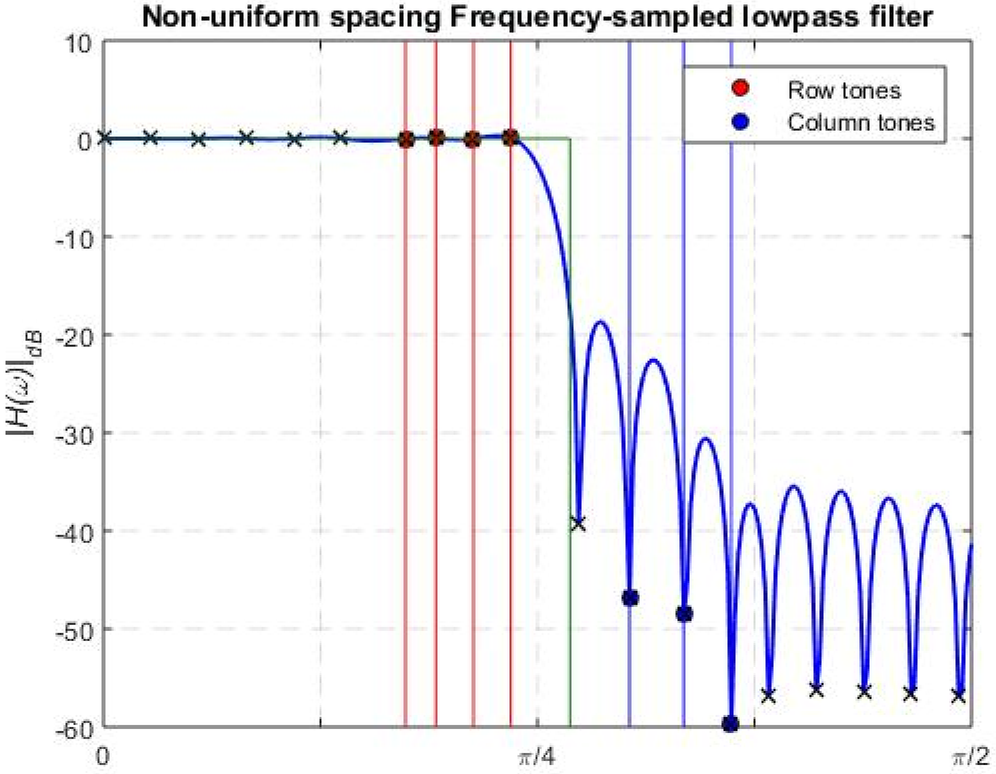

Kaiser-window highpass filter
$|H(\omega)|_{dB}$
10
0
-10
-20
-30
-40
-50
-60
0
π/4
π/2

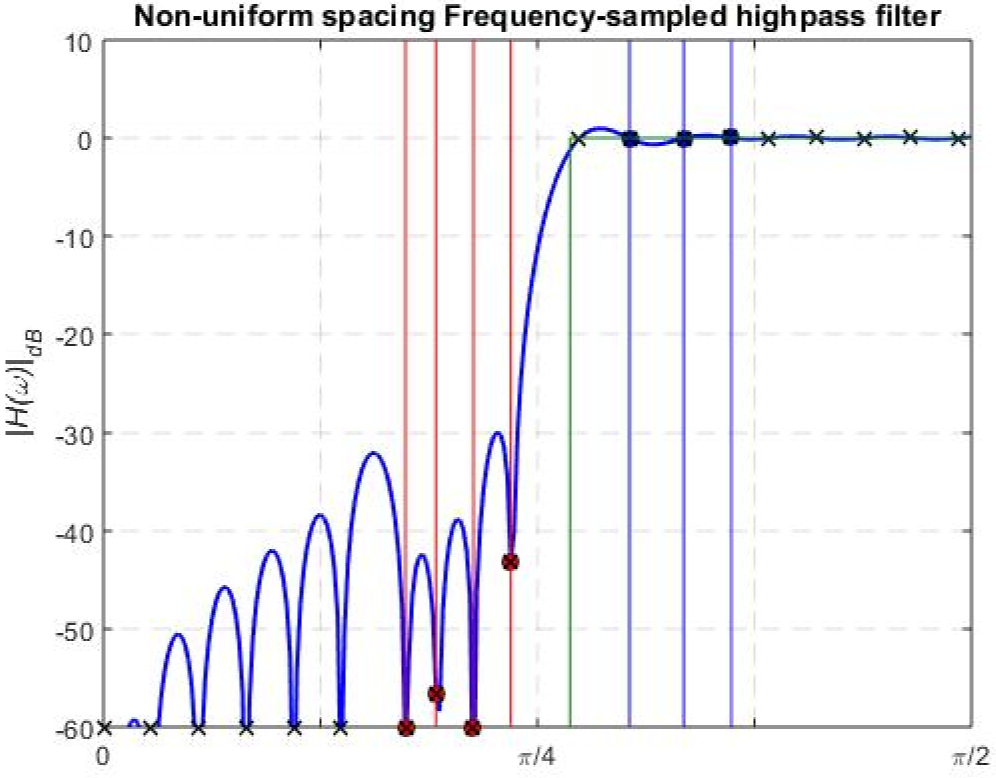

Spline lowpass filter
● Row tones
● Column tones
$|H(\omega)|_{dB}$
10
0
-10
-20
-30
-40
-50
-60
0
π/4
π/2

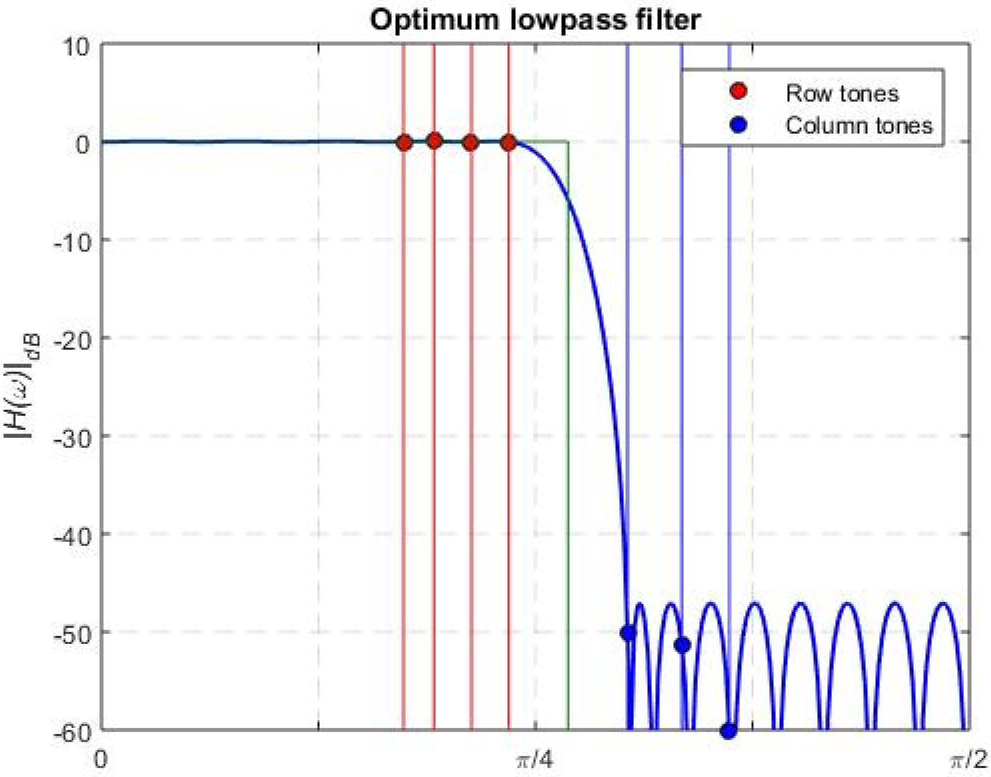

Spline highpass filter
$|H(\omega)|_{dB}$
10
0
-10
-20
-30
-40
-50
-60
0
π/4
π/2

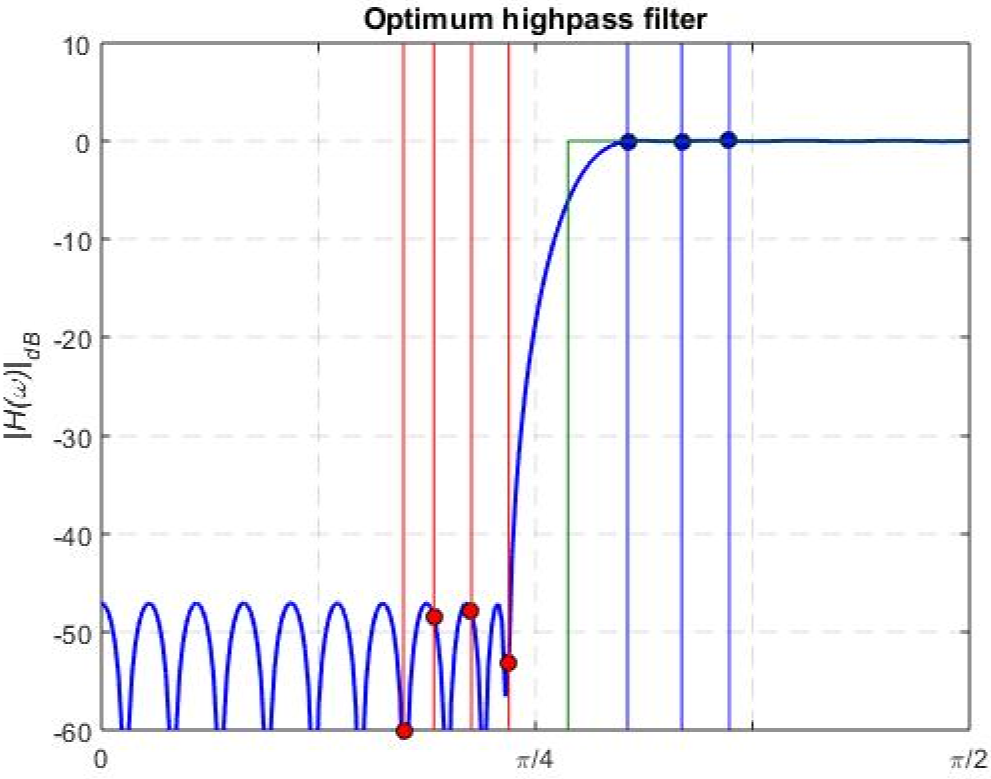

Uniform spacing Frequency-sampled lowpass filter
Row tones
Column tones
$|H(\omega)|_{dB}$
10
0
-10
-20
-30
-40
-50
-60
0
π/4
π/2


Uniform spacing Frequency-sampled highpass filter
$|H(\omega)|_{dB}$
10
0
-10
-20
-30
-40
-50
-60
0
π/4
π/2


Non-uniform spacing Frequency-sampled lowpass filter
● Row tones
● Column tones
$|H(\omega)|_{dB}$
10
0
-10
-20
-30
-40
-50
-60
0
π/4
π/2


Non-uniform spacing Frequency-sampled highpass filter
$|H(\omega)|_{dB}$
10
0
-10
-20
-30
-40
-50
-60
0
π/4
π/2


Optimum lowpass filter
Row tones
Column tones
$|H(\omega)|_{dB}$
10
0
-10
-20
-30
-40
-50
-60
0
$\pi/4$
$\pi/2$


Optimum highpass filter
$|H(\omega)|_{dB}$
10
0
-10
-20
-30
-40
-50
-60
0
$\pi/4$
$\pi/2$

Part 2: plotting functions

```
disp(' ')
disp('--- magdb_lab6.m --------------------------------')
type('magdb_lab6')
disp('--- sevendots.m -------------------------------')
type('sevendots')
```

```
--- magdb_lab6.m --------------------------------

function ph = magdb_lab6(h)
H = fft(h,1024);
w = linspace(0, 2*pi, 1024);
dB = mag2db(abs(H));
ph = plot(w/pi, dB);
set(ph,'LineWidth', 1.5, 'color', 'b');
ax = gca;
ax.XTick=[0 0.125 0.25 0.375 0.5];
ax.XTickLabel={'0','','\pi/4','','\pi/2'};
ax.GridLineStyle = '--';
xlim([0 0.5]);
ylim([-60 10]);
% xlabel('\omega(rad/\pi)');
ylabel('|\itH(\omega)|_d_B');
grid on;
return
--- sevendots.m ------------------------------

function y = sevendots(p)
% SEVENDOTS plots the seven dots corresponding to the row
% and column frequency of phone tones for lab6, on the
% frequency response dB scale plotting of the filters.
% by Steven Bao & Zizhe Ren (Mar 2016)
x = get(p, 'xdata');
y = get(p, 'ydata');
wtones = 2 *[697 770 852 941 1209 1339 1447] / 8000;
index = zeros(1, 7);
xdots = zeros(1, 7);
ydots = zeros(1, 7);
for i=1:7
    [~, index(i)] = min(abs(wtones(i) - x));
    xdots(i) = x(index(i));
    ydots(i) = y(index(i));
    if ydots(i) < -60
        ydots(i) = -60;
    end
end

dot1 = plot(xdots(1),ydots(1), 'ro', 'MarkerFaceColor',
 'r','MarkerEdgeColor','k');
plot(xdots(2),ydots(2), 'ro', 'MarkerFaceColor',
 'r','MarkerEdgeColor','k');
```

```
plot(xdots(3),ydots(3), 'ro', 'MarkerFaceColor',
 'r','MarkerEdgeColor','k');
plot(xdots(4),ydots(4), 'ro', 'MarkerFaceColor',
 'r','MarkerEdgeColor','k');
plot(xdots(5),ydots(5), 'ro', 'MarkerFaceColor',
 'b','MarkerEdgeColor','k');
plot(xdots(6),ydots(6), 'ro', 'MarkerFaceColor',
 'b','MarkerEdgeColor','k');
dot7 = plot(xdots(7),ydots(7), 'ro', 'MarkerFaceColor',
 'b','MarkerEdgeColor','k');
rlines = [1;1] * xdots(1:4);
plot(rlines, ylim, 'r', 'LineWidth',0.2);
blines = [1;1] * xdots(5:7);
plot(blines, ylim, 'b', 'LineWidth',0.2);
y = [dot1,dot7];
end
```

Published with MATLAB® R2015a