EE323 · Digital Signal Processing

Lab 7&8: Digital Filter Design

任振裕 / 11812214

December 18, 2020

Contents

Introduction 1

7.3 Design of a Simple FIR Filter 2

7.4 Design of a Simple IIR Filter 4

7.6 Filter Design Using Truncation 7

8.2 Filter Design Using Standard Windows 8

8.3 Filter Design Using the Kaiser Window 10

Summary 14

Appendix 14

Introduction

This lab merges the content of Lab 7 and lab 8, which is mainly composed of two parts:

- In the **first part**, we will cover some basic examples of FIR and IIR filters, and then introduces the concepts of FIR filter design.
- In the **second part**, we will mainly discuss some systematic methods of designing FIR filters.
- To understand the designs of FIR and IIR digital designs, we should keep different situations of below equations in our minds all the time:

Info:

- Difference equation of LTI causal digital filter (with input $x[n]$ and output $y[n]$):

$$y[n] = \sum_{i=0}^{N-1} b_i x[n-i] - \sum_{k=1}^{M-1} a_k y[n-k]$$

- Impulse response, z-transform, and z-transform:

$$h[n] = \sum_{i=0}^{N-1} b_i \delta[n-i] - \sum_{k=1}^{M-1} a_k h[n-k]$$

$$H(z) \triangleq \frac{Y(z)}{X(z)} = \frac{\sum_{i=0}^{N-1} b_i z^{-i}}{1 + \sum_{k=1}^{M-1} a_k z^{-k}}$$

$$H\left(e^{j\omega}\right) = H(z)\big|_{z=e^{j\omega}} = \frac{\sum_{i=0}^{N-1} b_i e^{-j\omega i}}{1 + \sum_{k=1}^{M-1} a_k e^{-j\omega k}}$$

Info:

- For FIR filters:
  * No poles, only zeros. $\rightarrow a_k = 0, \forall k \in \{1, 2, \ldots, M-1\}$:

$$h[n] = \sum_{i=0}^{N-1} b_i \delta[n-i]$$

  * Frequency response:

$$H\left(e^{j\omega}\right) = \sum_{i=0}^{N-1} b_i e^{-j\omega i}$$

- For IIR filters: Both poles and zeros exist, therefore impulse and frequency responses follow the above general forms.

7.3 Design of a Simple FIR Filter

Question 1:

- Analytical transfer function:

$$\begin{aligned} H_f(z) &= \left(1 - z_1 z^{-1}\right)\left(1 - z_2 z^{-1}\right) \\ &= \left(1 - e^{j\theta} z^{-1}\right)\left(1 - e^{-j\theta} z^{-1}\right) \\ &= 1 - 2\cos\theta z^{-1} + z^{-2} \end{aligned}$$

- Difference equation:

$$y[n] = x[n] - 2\cos\theta x[n-1] + x[n-2]$$

- The system diagram:

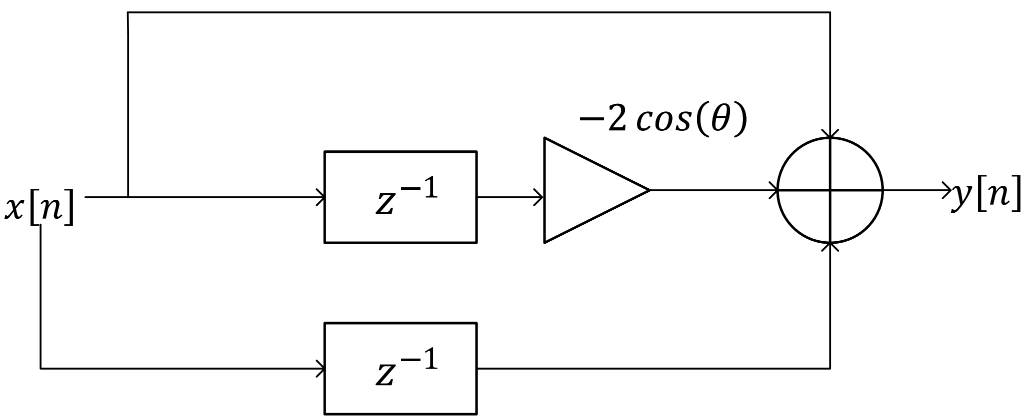

Figure 1: System diagram

- MATLAB codes for this section:

q7_3a.m

```
clear;
w = -pi:0.01:pi; z = exp(1j*w);
```

```
H1 = 1-2*cos(pi/6)*z.^(-1)+z.^(-2);
H2 = 1-2*cos(pi/3)*z.^(-1)+z.^(-2);
H3 = 1-2*cos(pi/2)*z.^(-1)+z.^(-2);
subplot(1,3,1);plot(w,abs(H1));title('\theta=\pi/6');
subplot(1,3,2);plot(w,abs(H2));title('\theta=\pi/3');
subplot(1,3,3);plot(w,abs(H3));title('\theta=\pi/2');
```

- Running result:

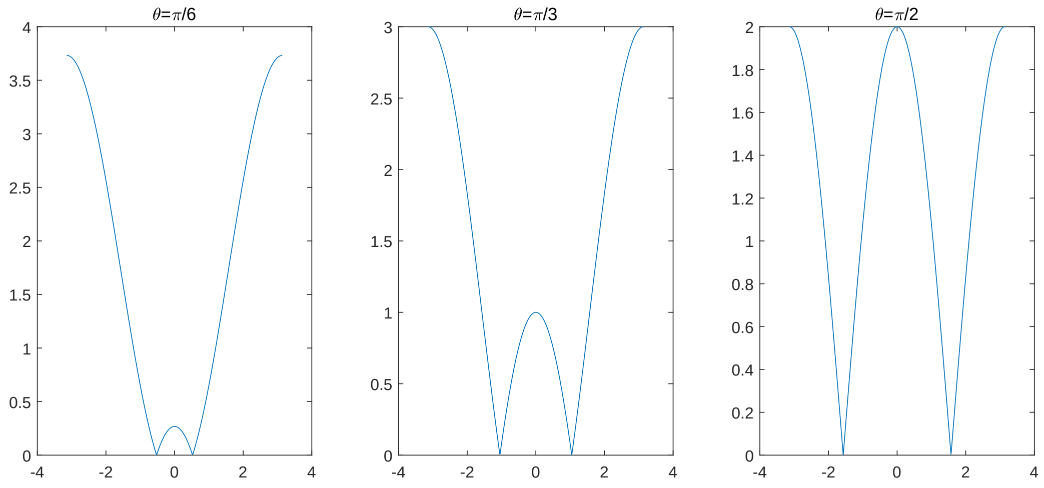

Figure 2: Hf with different θ

- Analysis: $\omega = \theta \longrightarrow |H_f(e^{j\omega})| == 0$.

Question 2:

- FIRfilter.m:

FIRfilter.m

```
function y = FIRfilter(x)
[X,w]=DTFT(x,0);
[Xmax,Imax]=max(abs(X));
h = [1 -2*cos(w(Imax)) 1];
y = conv(x,h);
end
```

- q7_3b.m:

q7_3b.m

```
clear;
load nspeech1;
y0 = FIRfilter(nspeech1);
y1 = nspeech1(100:200);
y2 = nspeech1(100:1100);
y3 = y0(100:200);
y4 = y0(100:1100);
[X2,w2] = DTFT(y2,0);
```

```
[Xmax,Imax] = max(abs(X2));
[X4,w4] = DTFT(y4,0);
subplot(221),plot([0:length(y1)-1],y1);title('Plot of the time domain
    of 101 samples');
subplot(222),plot(w2,abs(X2));title('Plot of magnitude of DTFT for
   1001 samples');
subplot(223),plot([0:length(y1)-1],y3);title('Plot of filtered output
   ');
subplot(224),plot(w4,abs(X4));title('Plot of Magnitude response of
   DTFT');
```

- Running result:

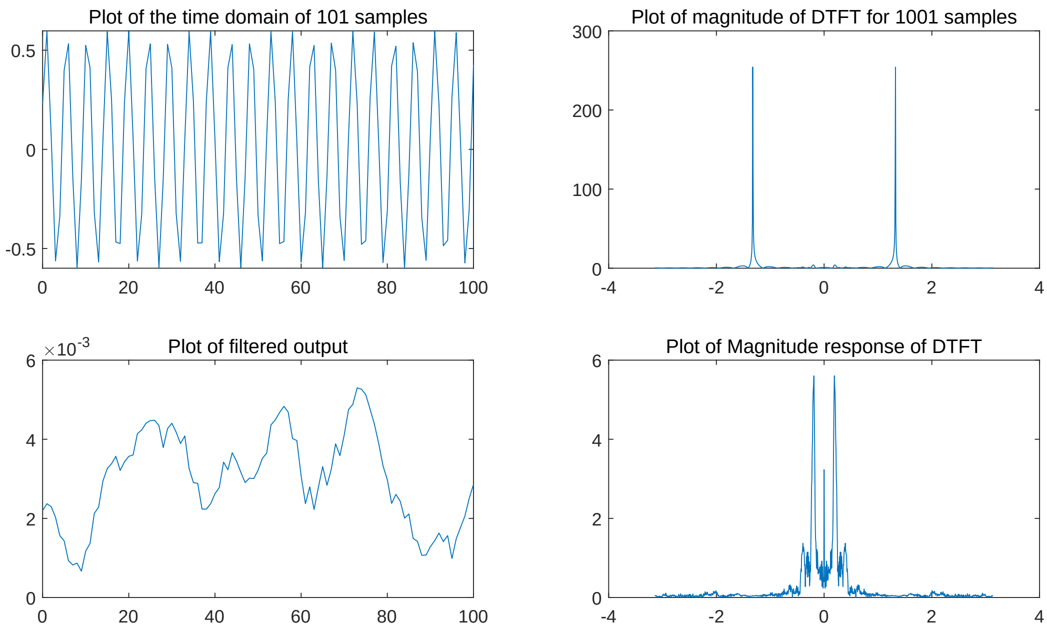

Figure 3: Effects of filtering.

- Analysis:
  - This is a bandpass filter.
  - After filtering, the distance of impulses becomes lower in the frequency domain and the the waveform in the time domain changes a lot.

7.4 Design of a Simple IIR Filter

Question 1:

- Analytical transfer function:

$$H_i(z) = \frac{1-r}{1-2r\cos\theta z^{-1} + r^2 z^{-2}}$$

- Difference equation:

$$y[n] - 2r\cos\theta y[n-1] + r^2 y[n-2] = (1-r)x[n]$$

- System diagram:

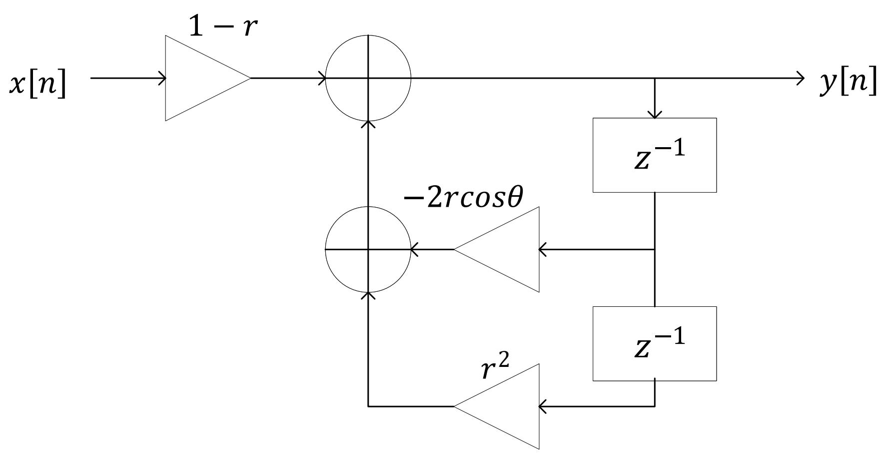

Figure 4: System diagram

- q7_4a.m:

q7_4a.m
```
clear;
w=-pi:0.01:pi;
z = exp(j*w);
r1 = 0.99;r2 = 0.9; r3 = 0.7;
Hi1 = (1-r1)./(1-2*r1*cos(pi/3).*z.^(-1)+(r1.^2)*(z.^(-2)));
Hi2 = (1-r2)./(1-2*r2*cos(pi/3).*z.^(-1)+(r2.^2)*(z.^(-2)));
Hi3 = (1-r3)./(1-2*r3*cos(pi/3).*z.^(-1)+(r3.^2)*(z.^(-2)));
subplot(1,3,1);plot(w,abs(Hi1));title('r=0.99');
subplot(1,3,2);plot(w,abs(Hi2));title('r=0.9');
subplot(1,3,3);plot(w,abs(Hi3));title('r=0.7');
```

- Running result:

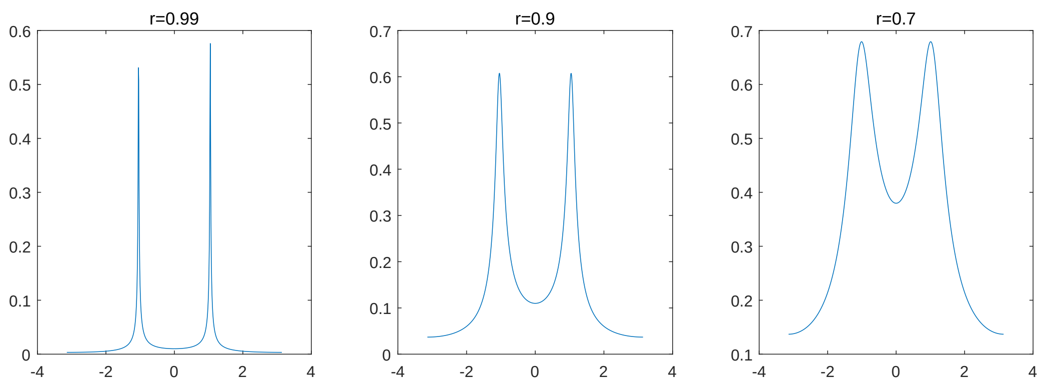

Figure 5: H_i with different r.

- Analysis: As r decreases, the magnitude response would become smoother. (The width of two impulses would become larger. And when $r \to 1$, the impulse could be considered as ideal impulse.)

Question 2:

- IIRfilter.m:

IIRfilter.m

```
function y = IIRfilter(x)
theta = (3146/8000)*2*pi;
r = 0.995;
N = length(x);
y = zeros(1,N);
y(1) = x(1);
y(2) = x(2)+2*r*cos(theta);
for i = 3:N
y(i) = x(i)+2*r*cos(theta).*y(i-1)-(r^2).*y(i-2);
end
end
```

- q7_4b.m:

q7_4b.m

```
clear;
load pcm;
sig1 = pcm(100:200);
sig2 = pcm(100:1100);
n = 0:length(sig1)-1;
[X,w] = DTFT(sig2,0);
f = IIRfilter(pcm);
f1 = f(100:200);
f2 = f(100:1100);
theta=(3146/8000)*2*pi;
[F,w] = DTFT(f2,0);
w1 = w(w>(theta-0.02));
w1 = w1(w1<(theta+0.02));
X1 = X(w>(theta-0.02) & w<(theta+0.02));
f_X1 = F(w>(theta-0.02) & w<(theta+0.02));
subplot(611),plot(n,sig1);title('101 samples of the orignal signal');
subplot(612),plot(w,abs(X));title('magnitude of DTFT');xlim([-pi pi])
    ;
subplot(613),plot(n,f1);title('101 samples of the filtered output');
subplot(614),plot(w,abs(F));title('magnitude of DTFT filtered output'
    );xlim([-pi pi]);
subplot(615),stem(w1,abs(X1));title('magnitude of DTFT in the range
    of the orignal signal');
subplot(616),stem(w1,abs(f_X1));title('magnitude of DTFT in the range
     of the filtered output');
```

- Running Result:

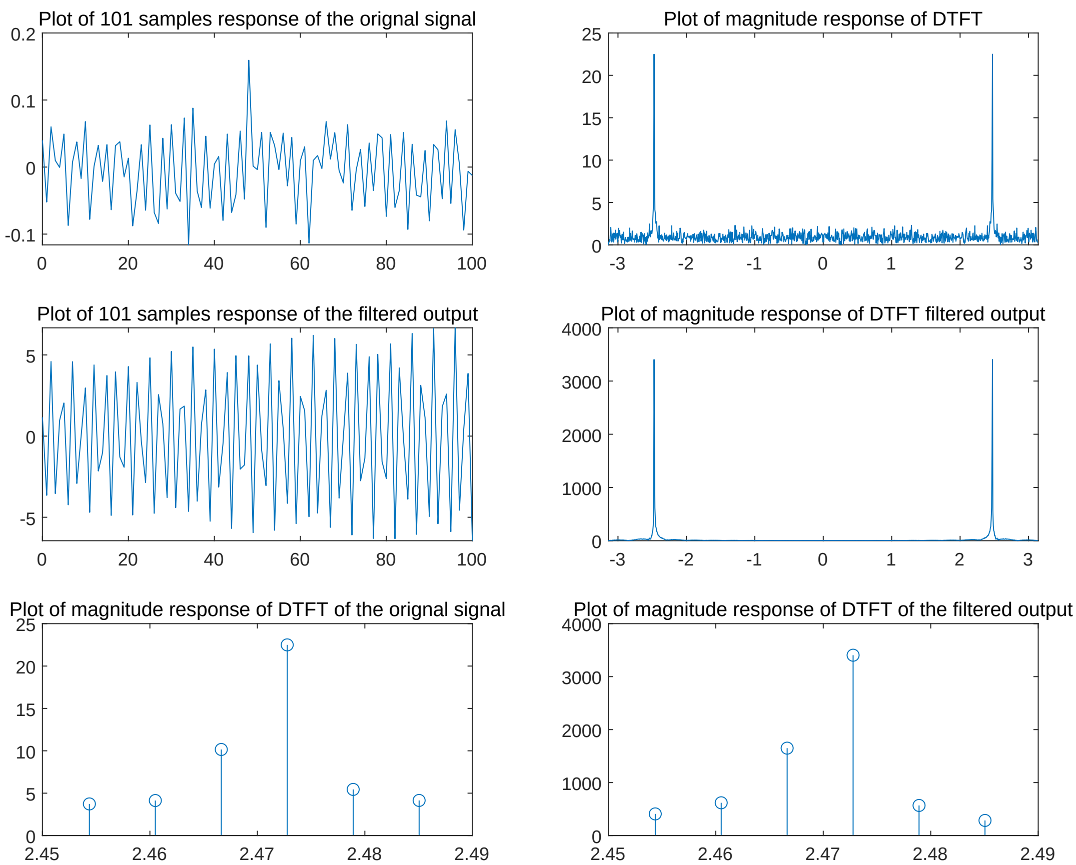

Figure 6: Six plots of Questions 2

- Analysis: After filtering, the speech noise in the frequency domain decrease much more. Moreover, the sound after filtering is much more clearer.

7.6 Filter Design Using Truncation

- LPFtrunc.m and test codes:

LPFtrunc.m

```
function h=LPFtrunc(N)
for i=1:N
h(i)=2/pi*sinc(2/pi*(i-1-(N-1)/2));
end
end
```

q7_6.m

```
clear;
load nspeech2.mat;
N1=21;
N2=101;
h1 = LPFtrunc(N1);
[X1,w1] = DTFT(h1,512);
```

```
h2 = LPFtrunc(N2);
[X2,w2] = DTFT(h2,512);
figure(1);
subplot(121),plot(w1,abs(X1));title("magnitude not in dB, N=21");
subplot(122),plot(w2,abs(X2));title("magnitude not in dB, N=101");
figure(2);
subplot(121),plot(w1,20*log10(abs(X1)));title("magnitude in dB, N
    =21");
subplot(122),plot(w2,20*log10(abs(X2)));title("magnitude in dB, N
    =101");
```

- Running result:

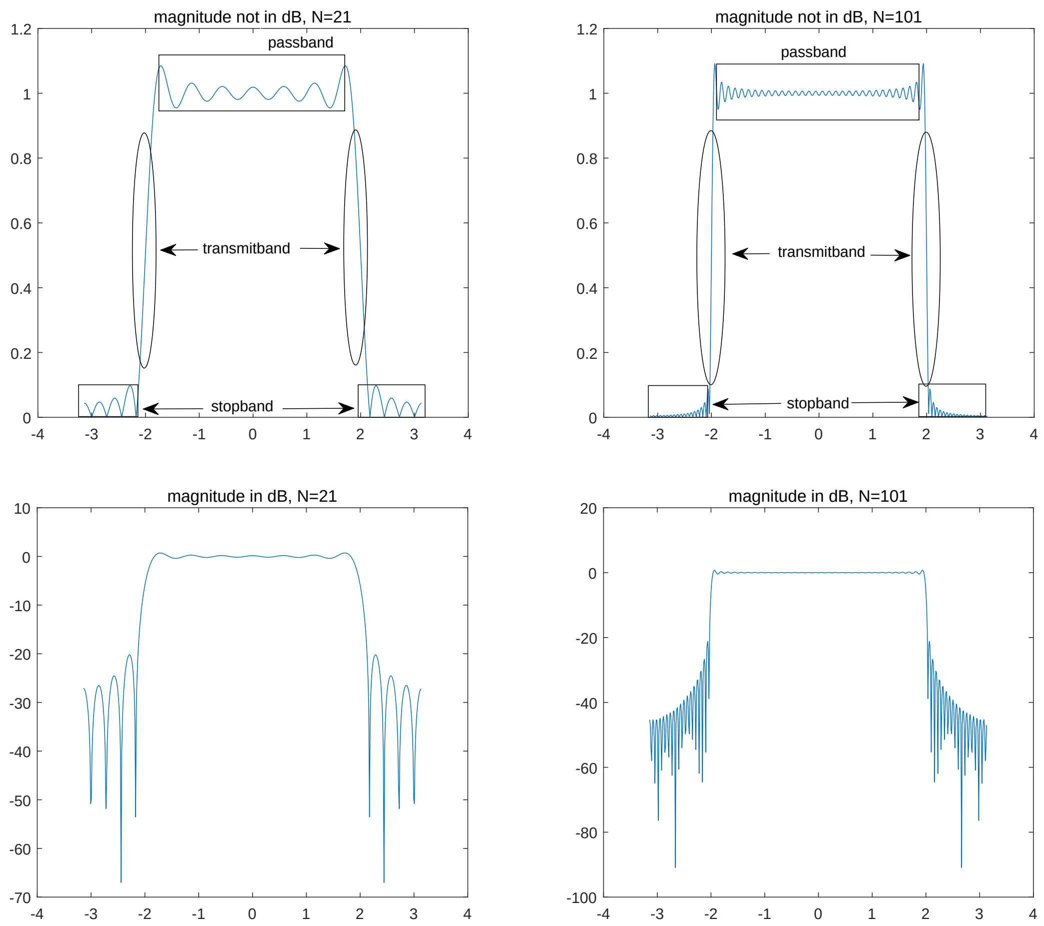

Figure 7: Magnitude Response.

- Analysis:
  1. When N increases, the filtering becomes much more better because of the more sampling points.(Compare the Gibbs Phenomenon in the passband.)
  2. After filtering, the filtered result sounds much smoother, with fewer sharp tones.

8.2 Filter Design Using Standard Windows

Running Result:

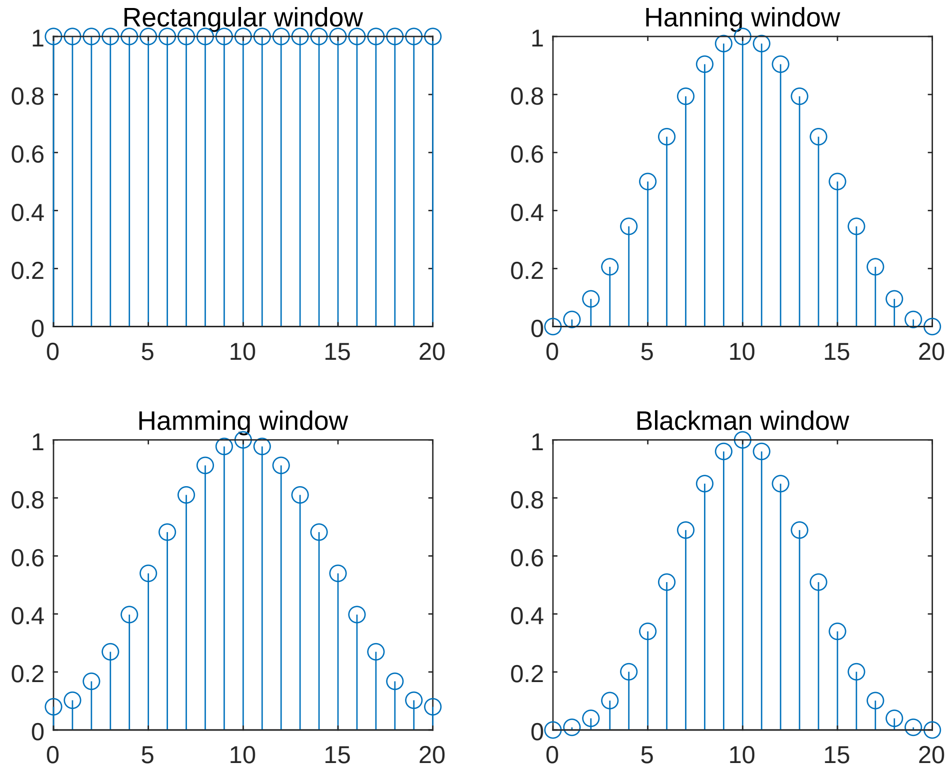

Figure 8: Time domain plots of the four windows.

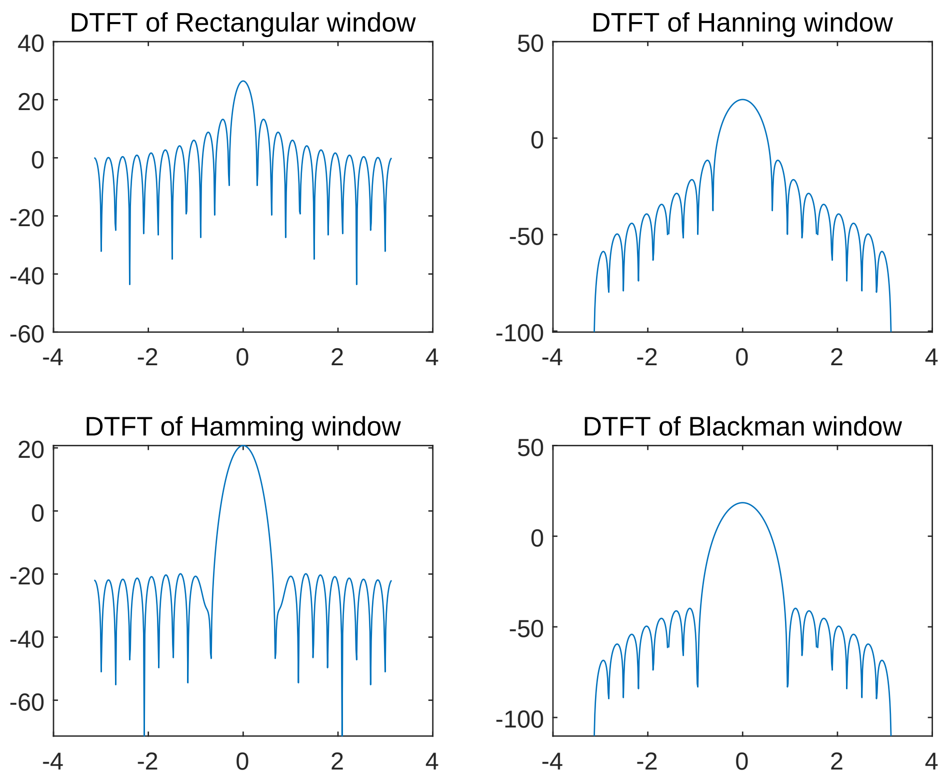

Figure 9: Frequency domain plots of the four windows.

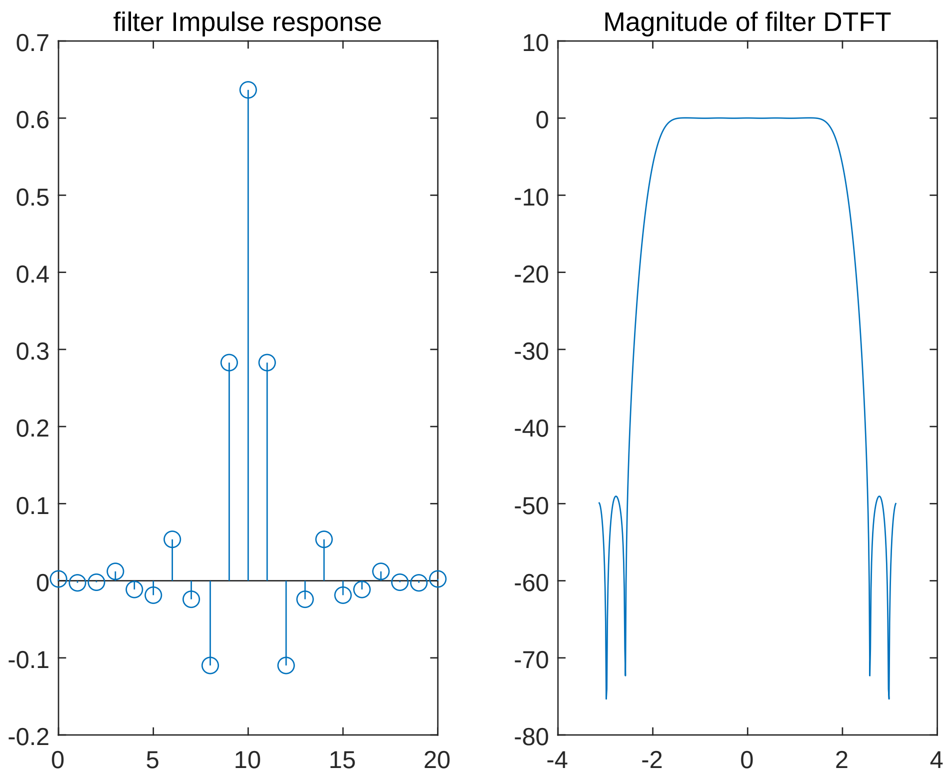

Figure 10: impulse response and magnitude of filter's DTFT

Type	Theoretical mainlobe (dB)	Measured mainlobe width (dB)	Theoretical peak to sidelobe amp (dB)	Measurered peak to sidelobe amp (dB)
rectangular window	0.598	0.6	-13	-13.2
hanning window	1.2	1.2	-32	-31.5
hamming window	1.2	1.2	-43	-41.4
blackman window	1.8	1.9	-58	-58.5

Table of the measured and theoretical window spectrum parameters.

Analysis:

1. The theoretical result and measured result are very close.
2. Performance comparison: blcak window > hamming window > hanning window > rectangular window. (Larger mainlobe widh and lower sidelobe.)

8.3 Filter Design Using the Kaiser Window

Codes:

q8_3.m

```
% For both Questions 1 and 2.
clear;
N=0:20;
x1=kaiser(21,0);x2=kaiser(21,1);x3=kaiser(21,5);
[X1,w1]=DTFT(x1,512);
[X2,w2]=DTFT(x2,512);
[X3,w3]=DTFT(x3,512);



figure(1);
sgtitle('three plots of Kaiser windows.');
subplot(3,2,1);stem(N,x1);xlabel('n(s)');title('kaiser window for
    beta=0');
subplot(3,2,2);plot(w1,20*log10(X1));xlabel('w(rad)');title('
    magnitude of DTFT for beta=0');
subplot(3,2,3);stem(N,x2);xlabel('n(s)');title('kaiser window for
    beta=1');
subplot(3,2,4);plot(w2,20*log10(X2));xlabel('w(rad)');title('
    magnitude of DTFT for beta=1');
subplot(3,2,5);stem(N,x3);xlabel('n(s)');title('kaiser window for
    beta=5');
subplot(3,2,6);plot(w3,20*log10(X3));xlabel('w(rad)');title('
    magnitude of DTFT for beta=5');



figure(2);
load nspeech2;
beta=4.0909;N=51;
w = kaiser(N, beta);
h1 = LPFtrunc(N);
h2 = h1.*w';
[H,w] = DTFT(h2,512);
sgtitle('three plots of magnitude response.')
subplot(3,1,1);
plot(w,20*log10(H));title('DTFT of filter in dB');
subplot(3,1,2);
plot(w(abs(w)<=1.8),20*log10(H(abs(w)<=1.8)));title('|w| <= 1.8 in dB
    ');
subplot(3,1,3);
plot(w(abs(w)>=2.2),20*log10(H(abs(w)>=2.2)));title('|w| >= 2.2 in dB
    ');
f = conv(h2,nspeech2);
[F,w] = DTFT(f,512);



figure(3);
plot(w,20*log10(F)); title('magnitude of filtered signal in dB');
xlim([-3.5 3.5]);
```

Question 1:

Running Result:

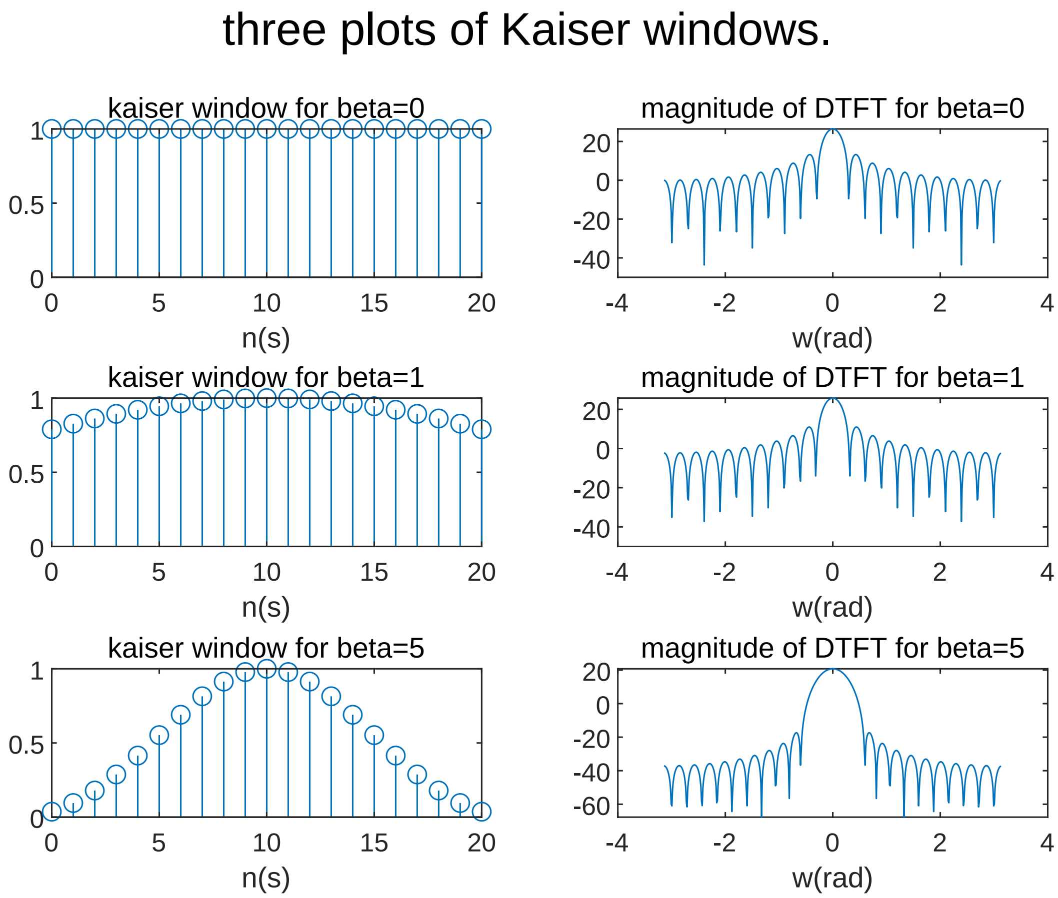

Figure 11: plots of the 3 Kaiser windows and the magnitude of their DTFT's indecibels.

Comment:
As β increases, the time domian window becomes no more flat(Two sides of the window decrease.). Moreover, in the frequency domian, sidelobes gradually decrease.

Question 2:

Calculate β and N:

$$\omega[n] = \begin{cases} I_0\left(\beta\frac{\sqrt{n(N-1-n)}}{N-1}\right) & \\ \frac{1}{I_0(\beta)} & n = 0, 1, \ldots, N-1 \\ 0 & \text{otherwise} \end{cases}$$

$$\beta = \begin{cases} 0.1102(A-8.7) & A > 50 \\ \{0.5842(A-21)^{0.4} + 0.07886(A-21) & 21 \leq A \leq 50 \\ 0.0 & A < 21 \end{cases}$$

$$N = \left\lceil 1 + \frac{A-8}{2.285\left(\omega_s - \omega_p\right)} \right\rceil$$

$\therefore \beta = 4.0909, N = 51$. Running Result:

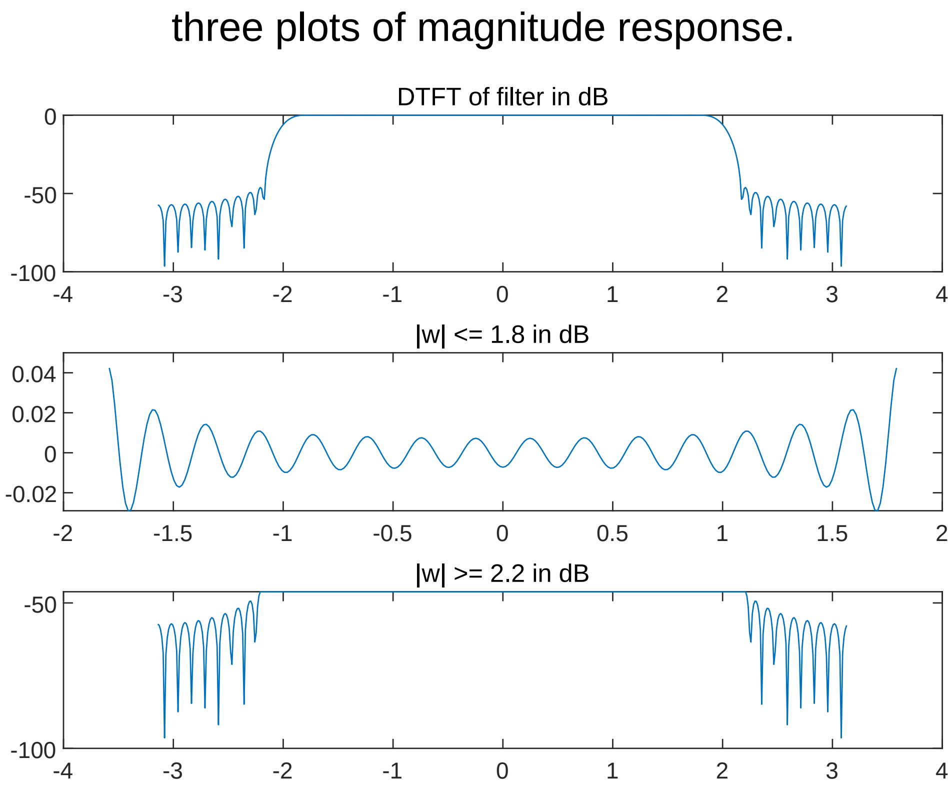

Figure 12: three plots of magnitude response.

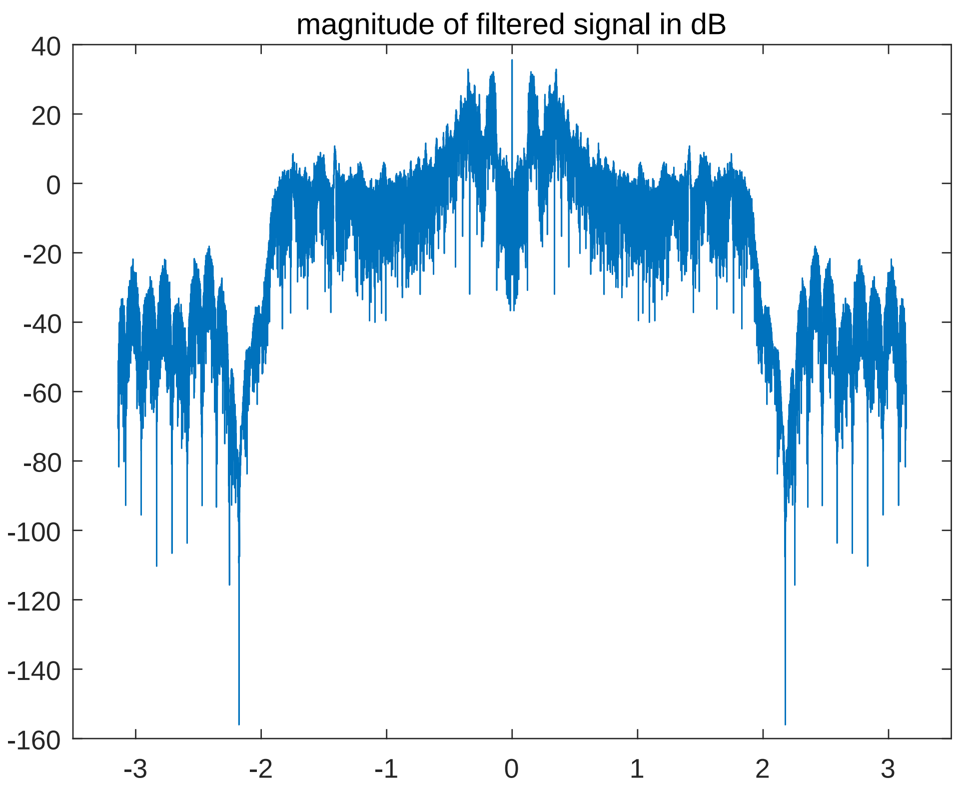

Figure 13: magnitude of filtered signal in dB

Analysis:

1. After filtering, noise and sidelobe decrease. The DTFT becomes smoother.
2. Filtered sounds are better and clearer.

Summary

Info: 这里拿中文写了 ◡̈。

- **关于这篇报告：**
对于这个篇报告，我有一点小遗憾。其实最初原本想拿 latex 写一个像之前几次 Lab 一样较为完美的学术报告，但是由于各种期末考试的压力，我这篇报告在考试后才完成，因此显得有一些虎头蛇尾。但是总得来说，通过这次实验的各种小型实验，我对 IIR 和 FIR 滤波器的设计都有了一定程度的理解，也获得不了不少宝贵的编程经验以及分析问题的能力。显然这是更为重要的。
- **关于整个学期：**
  - 说实话，虽然这个学期学业比较繁重，但是我还是收获了不少东西：从 MATLAB，Simulink 的基本使用，再到对信号处理的一些函数的复现，以及后来深入的滤波器设计和计算机生成音乐的 Mini Project, 这都是我前所未有的经验，也让我感受到了作为一个"工程师"的快乐。
  - 另外，对于每次 LAB 课报告的撰写，我也收获很多。譬如，为了科学严谨的叙述问题，我开始采用更多的数学语言描述问题或是使用 visio 绘制流程图。为了让自己的排版更加精美，我也逐渐尝试了从 word，到 markdown 再到 Latex 的排版方法。同样的，我也逐渐熟悉了 Latex 公式的编写并设置自己的 Latex 模板。我认为，这些都是我自己这学期超越分数最为宝贵的经验，为自己将来的科研和学习打下坚实的基础。
- 最后，感谢老师和学助对我们的指导和关心！

Appendix

DTFT.m

```
function [X,w] = DTFT(x,M)
% This function computes samples of the DTFT of x.
% To compute the DTFT of x, use
%
%            [X,w] = DTFT(x,0)
%
% where X is the vector of DTFT samples and w is the
% vector of radial frequencies. To compute at least
% M samples of the DTFT, you may use the command
%
%            [X,w] = DTFT(x,M)
%
% This is useful when the plot of X versus w does
% not contain a sufficient number of points.
N = max(M,length(x));
N = 2^(ceil(log(N)/log(2)));
% Take the padded fft
X = fft(x,N);
w = 2*pi*( (0:(N-1))/N );
w = w - 2*pi*(w>=pi);
% Shift FFT to go from -pi to pi
X = fftshift(X);w = fftshift(w);
```