# EE320 Assignment

Xiaohan Huang | Reg No. 201920667

## Q1.1: Using pen & paper, calculate the squared magnitude response of this system.

Because $H_1(z) = 1 + z^{-1}$ $\quad H(z)|_{z=e^{j\Omega}} = H(e^{j\Omega})$

So the squared magnitude response is

$$|H(e^{j\Omega})|^2 = H(e^{j\Omega}) \times H^*(e^{j\Omega}) = (1+e^{-j\Omega})(1+e^{j\Omega})$$

$$= 2 + e^{j\Omega} + e^{-j\Omega} = 2 + 2\cos\Omega$$

## Q1.2: A new system H2(z) = H1(zN ), with N ∈ N (given as a return parameter N from the function call room()), arises through expansion of h1[n]. Its impulse response h2[n] is obtained through expansion from h1[n]: h2[n] = h1[Nn], i.e. h2[n] contains the coeffiffifficients of h1[n], but each separated by (N 1) zero coeffiffifficients:

```
2h1 = [1 1];
h2 = zeros(1,N*length(h1));
h2(1:N:end) = h1;
```

## Using pen & paper, What is the squared magnitude response of the expanded system h2[n]?

```
x=audioread('FarEndSignal.wav');
sound(x, 8000);
xf=x;
[y, N]=room(xf, 201920667);
N
```

```
>> EE320

N =

    30

fx >>
```

According to the function call room( ), we know $N=30$

So the squared magnitude response is

$$|H_2(e^{j\Omega})|^2 = \left|1+(e^{-j\Omega})^{30}\right|^2 = (1+e^{-30j\Omega})(1+e^{30j\Omega})$$

$$= 2+e^{30j\Omega}+e^{-30j\Omega} = 2+2\cos 30\Omega$$

## Q1.3: How do the magnitude responses of h1[n] and h2[n] related to each other?

① the magnitude responses of $h_1[n]$ is $H_1(e^{j\Omega}) = 1+e^{-j\Omega}$

② the magnitude responses of $h_2[n]$ is $H_2(e^{j\Omega}) = 1+e^{-30j\Omega}$

Similar to the $H_1(z^N) = H_2(z) \circ\!\!-\!\!\bullet\; H_2(e^{j\Omega}) = H_1(e^{Nj\Omega})$

$\therefore H_2(e^{j\Omega}) = H_1(e^{32j\Omega})$

## Q2.1: Design a fifilter h3[n] and its N-fold expansion h[n] = h3[nN].

**h3 = fir1(23,0.4);**

**h = zeros(1,N*length(h3));**

**h(1:N:end) = h3;**

## Plot its impulse and magnitude responses.

```
%Design a filter h3 and its N-fold expansion and plot its impulsse and
%magnitude responses
h3 = fir1(23,0.4);
h = zeros(1,N*length(h3));
h(1:N:end) = h3;

figure(1);
impz(h3);

figure(2);
fs=8000;
f=(0:length(h)-1)/length(h)*fs;
plot(f,abs(fft(h)));
xlabel('Frequency/[Hz]');
ylabel('Magnitude');
```

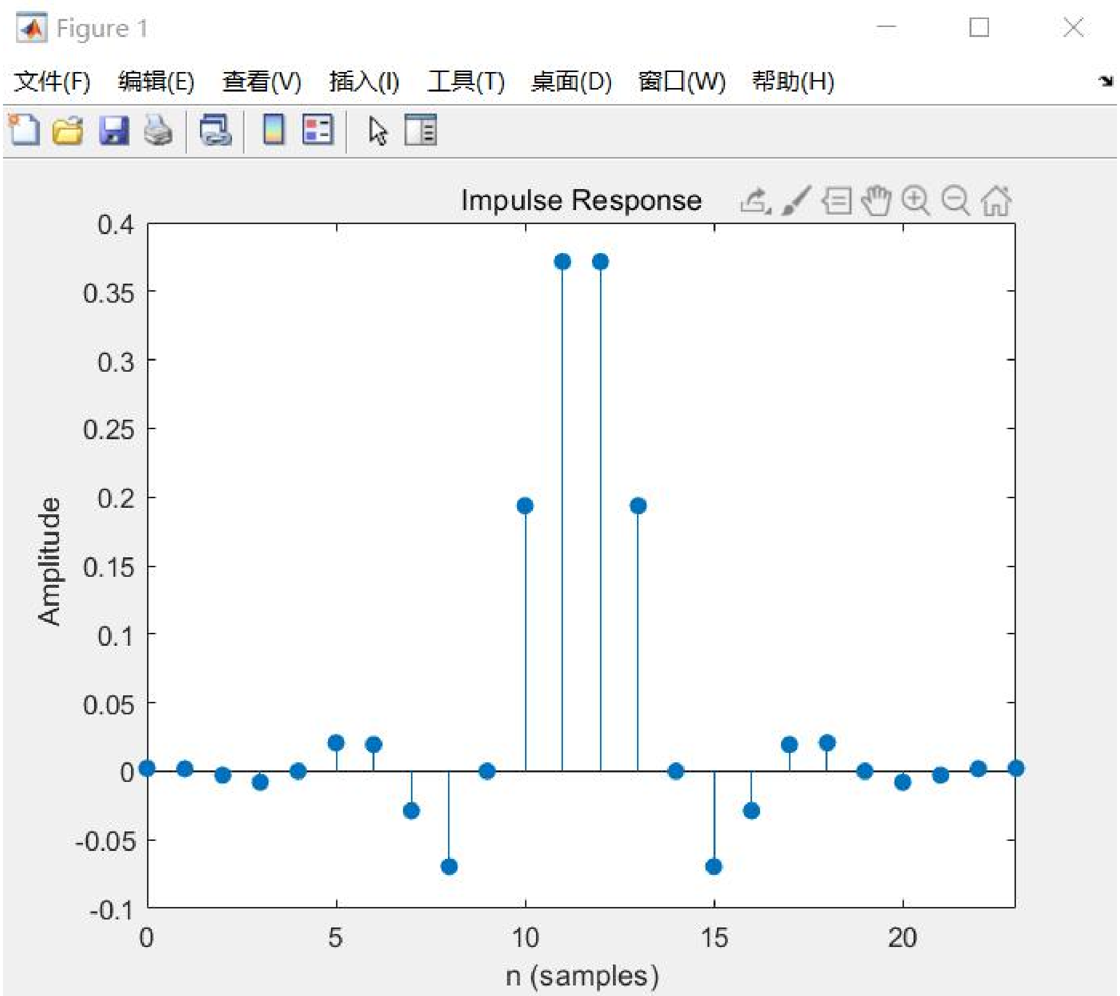


Figure 1
文件(F) 编辑(E) 查看(V) 插入(I) 工具(T) 桌面(D) 窗口(W) 帮助(H)
Impulse Response
Amplitude
n (samples)
0.4
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
-0.05
-0.1
0
5
10
15
20

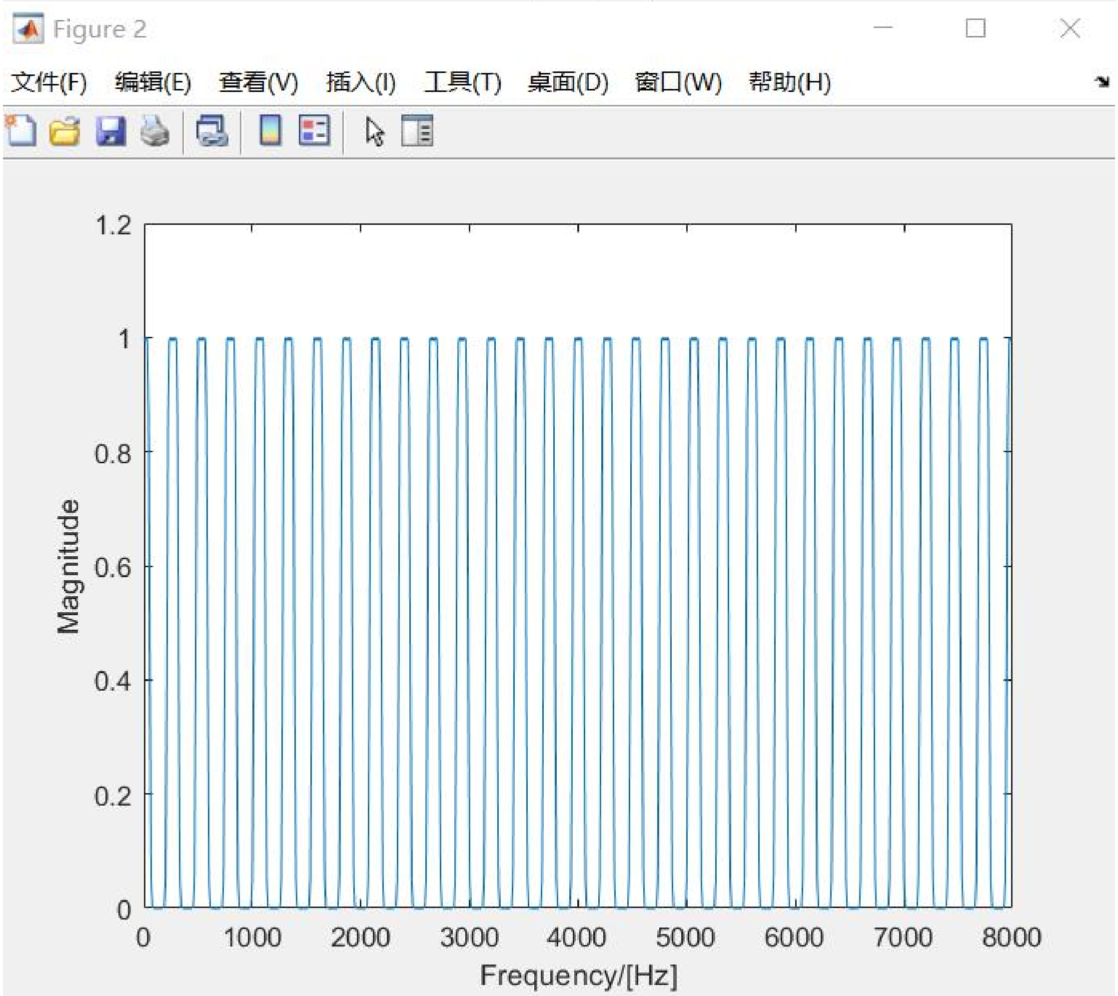
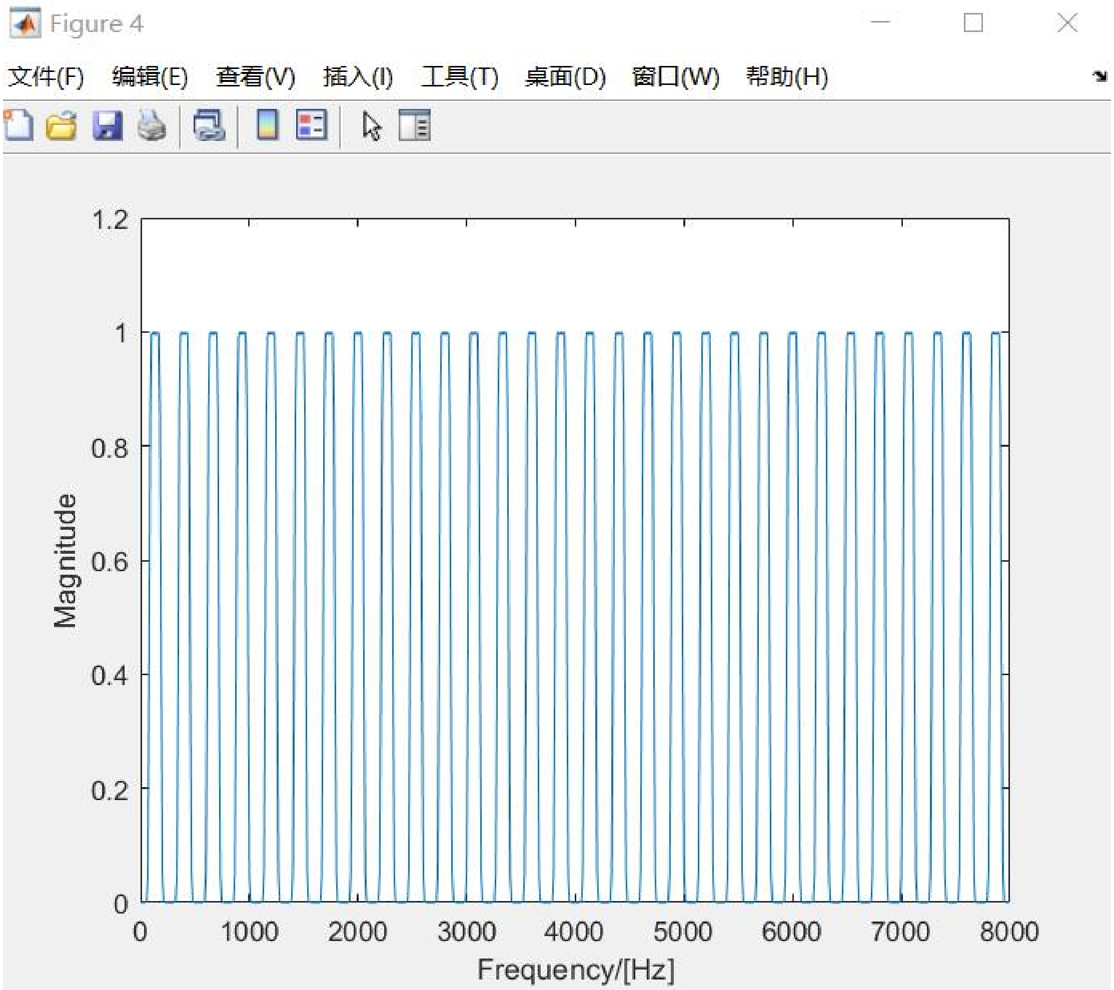


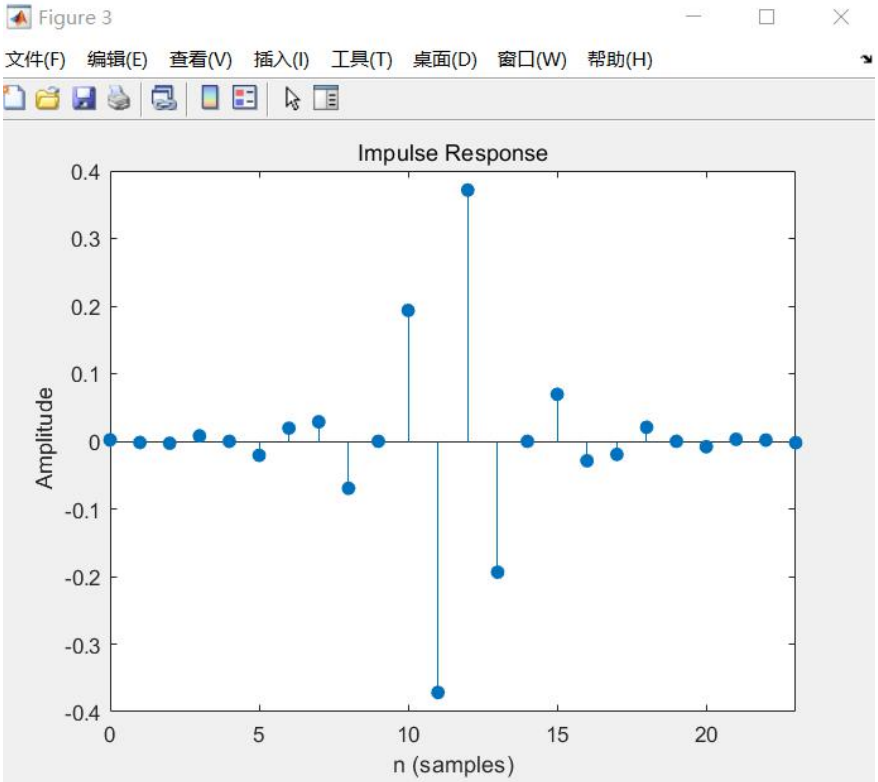
Figure 2
文件(F) 编辑(E) 查看(V) 插入(I) 工具(T) 桌面(D) 窗口(W) 帮助(H)
Magnitude
Frequency/[Hz]
1.2
1
0.8
0.6
0.4
0.2
0
0
1000
2000
3000
4000
5000
6000
7000
8000

**Q2.2: Create a fifilter g3[n] = (( 1)nh3[n] and its N-fold expansion g[n] = g3[nN]. Plots its impulse and magnitude responses. How do the magnitude responses of h[n] and g[n] relate to each other?**

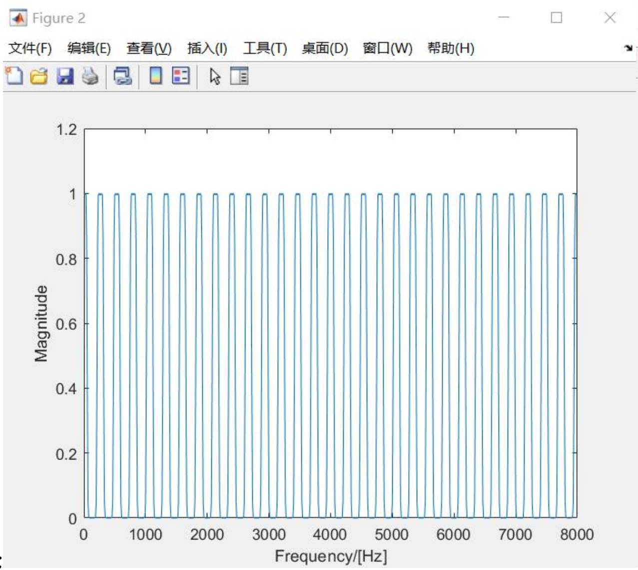
*Because the odd terms should be negative so we code like this* `g3 = (-1).^(0:23).*h3;`

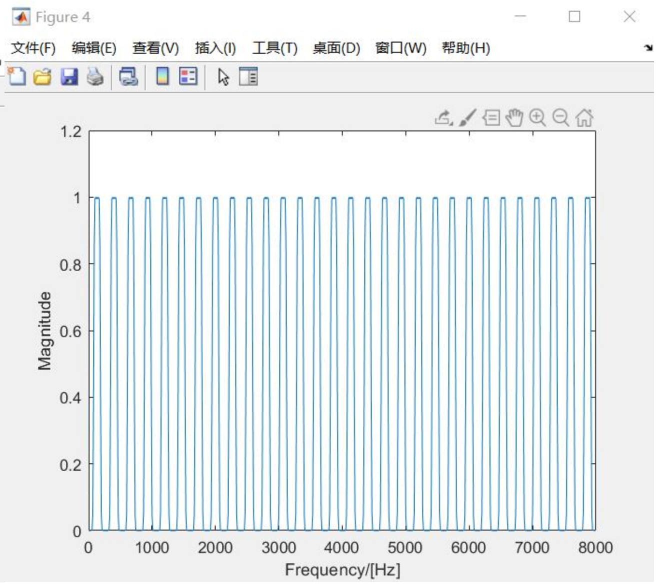
```
%Design a filter g3[n] = (( 1)nh3[n] and its N-fold expansion g[n] = g3[nN]
%Plots its impulse and magnitude responses.
g3=h3;
g3(2:2:end) = -g3(2:2:end);
g3 = (-1).^(0:23).*h3;
g = zeros(1,N*length(g3));
g(1:N:end) = g3;

figure(3);
impz(g3);

figure(4);
fs=8000;
f=(0:length(g)-1)/length(g)*fs;
plot(f,abs(fft(g)));
xlabel('Frequency/[Hz]');
ylabel('Magnitude');
```






The relationship between h[n] and g[n]: Comparing two magnitude responses figure




below:

we can see that: The Magnitude is same but the Frequency is a little different. As for h[n] when the frequency is 0, the magnitude is 1. As for the g[n] ,  when the frequency is 0 , the magnitude is 0.

**Q3.1: Listen to the near-end signal y[n] without any processing. What do you hear? By implementing the overall system according to Fig. 1 including the systems h[n] and g[n] as defiined in Q2. (you may use Matlab's filter( ) command), what does yf[n] sound like?**

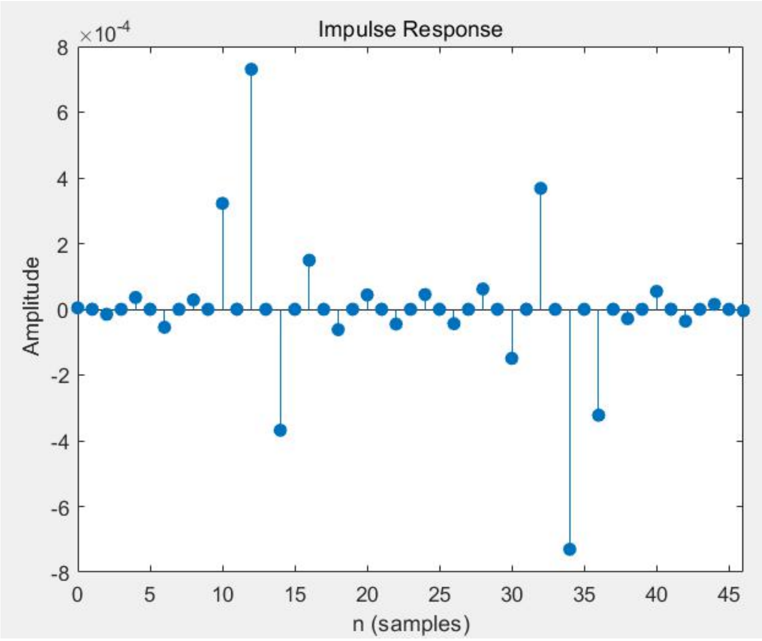
Listen to the near-end signal y[n] without any processing: I can hear the echo and noise.

Listen to the signal yf[n]: I can hear that the noise is hardly to hear but it exits. As for the echo, it is reduced.

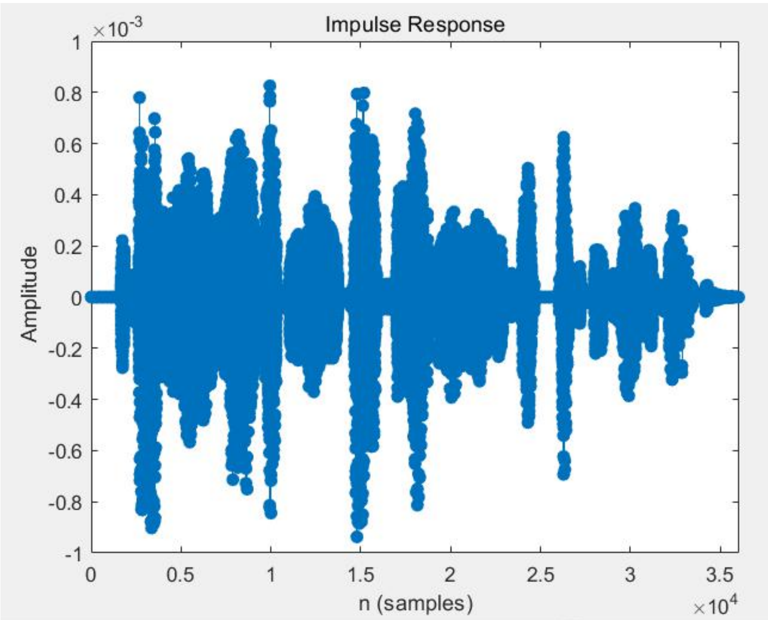
**Q3.2: If so, why is the echo of Bob reduced? Also, in case Prof Stewart's voice is distorted, why could this be the case?**

```
% %calculate CONVOLUTION
C1=conv(g3,h3);
figure(5);
impz(C1);

C2=conv(C1,x);
figure(6);
impz(C2)
```

The reason of why the echo of Bob reduced: let g3[n] and h3[n] be convoluted. Then we get a figure C1:




And then convolution of the below figure C1 and x, and we get a figure C2:

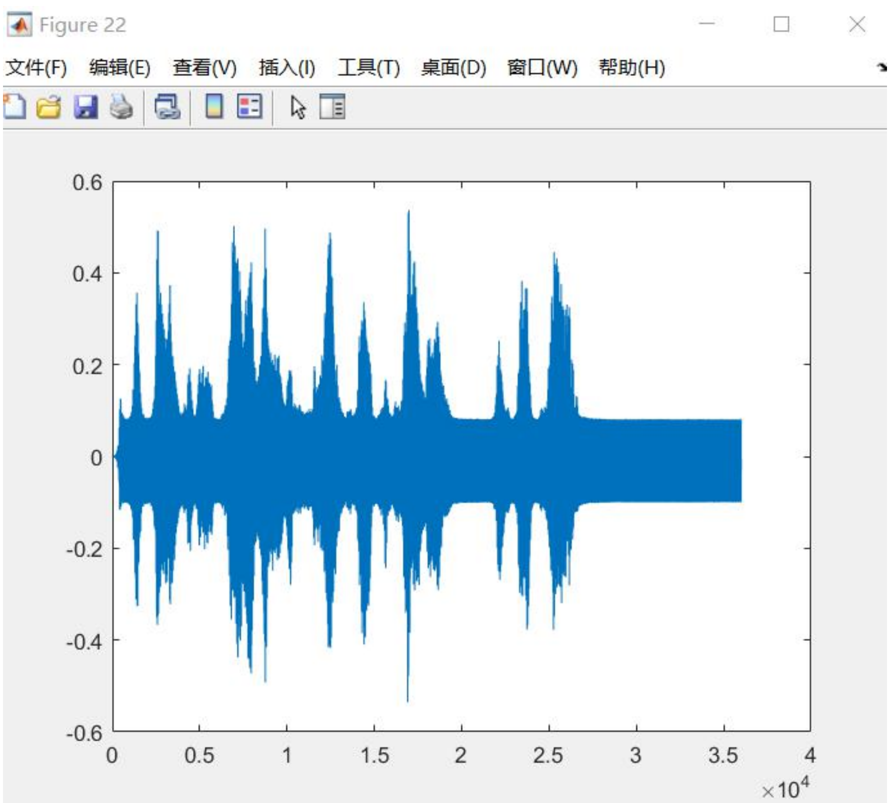


We can see that the voice has already been reduced.

The distortion of Prof Stewart's voice : We also can see that there are some times with zeros the voice is disappeared from the figure C2.

**Q3.3: Implement the fifilter without using any of Matlab's provided functions. How can you minimise the computational complexity of your implementation?**

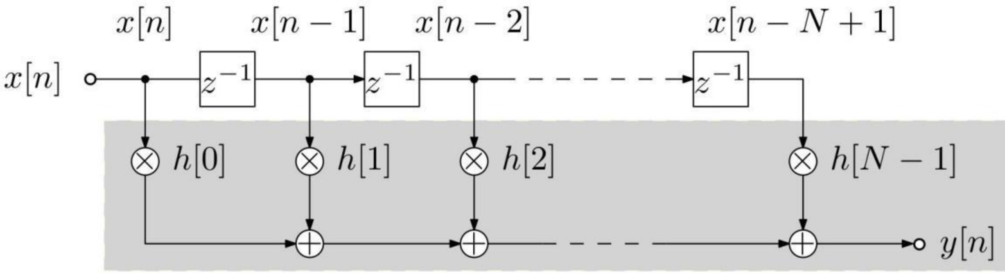
```
%recit the FarEndSignal.wav file and filter it
x=audioread('FarEndSignal.wav');
xf=filter(g,1,x);
[y,N]=room(xf,201920667);
yf=filter(h,1,y);
% sound(yf,8000);

%3.3 NOTE 264/306 FIR Filter Implementation
N1=length(h);                               % filter length
y_tdl=zeros(N1,1);                          % TDL vector initialisation
for n=1:length(y)                           % iteration --- once per sampling period

% step 1: update TDL with latest samples
y_tdl = [y(n); y_tdl(1:(N1-1))];

% step 2: calculate output y[n] using scalar product
yfn(n) = h*y_tdl;
end;
figure(22);
plot(yfn);
```



Use FIR Filter

## 11.4 FIR Filter Implementation

- To implement the FIR structure below, we require 2 steps per sampling period

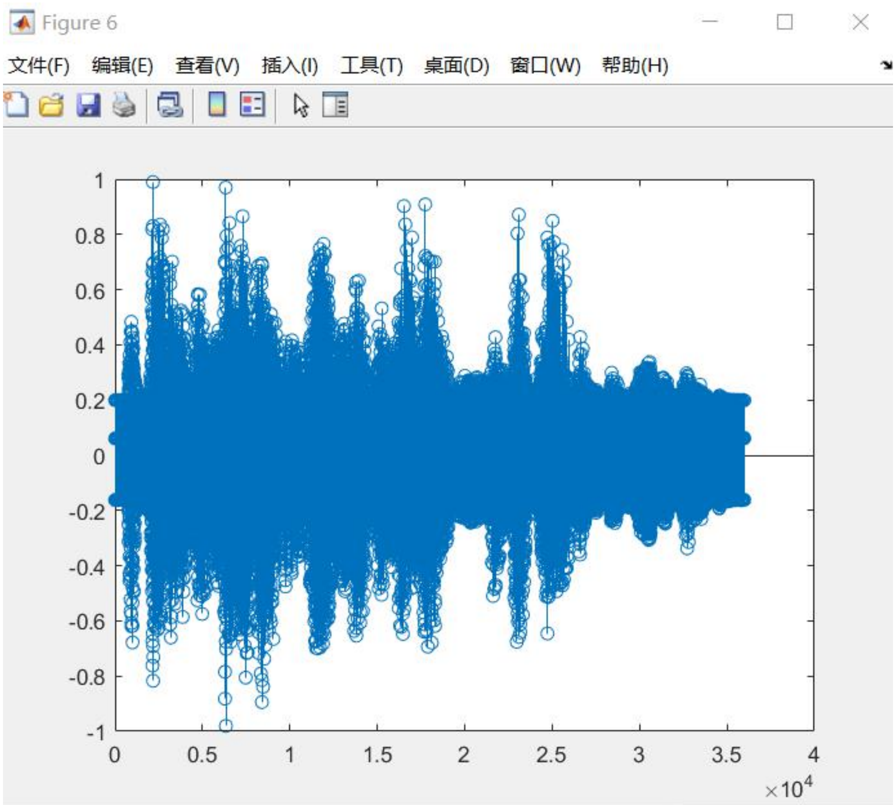


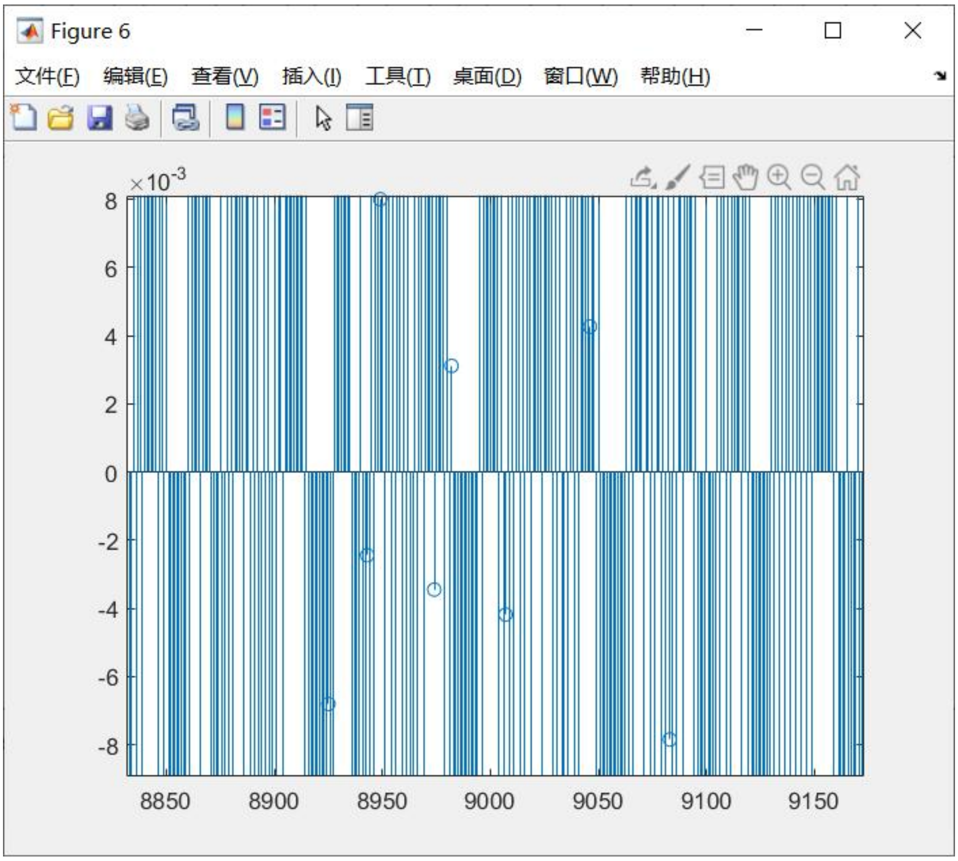

- *Step 1:* update the tapped-delay line with the most recent sample $x[n]$;
- *Step 2:* multiply the data samples with the coefficients, and accummulate the results to determine the output $y[n]$;
- then increment the time index $n \longrightarrow n+1$ and repeat ...



**Q4.1: inspecting the time domain waveform y[n] by fifinding a segment where the signal is otherwise quiet (i.e. interference dominates)**

```
%inspecting the time domain waveform y[n]
figure(6);
stem(y);
```






## Q4.2: evaluating the discrete-time Fourier transform

**fs = 8000;** **% sampling frequency in Hertz**

**f = (0:length(y)-1)/length(y)*fs ;** **% frequency scale plot(f,abs(fft(y)));**

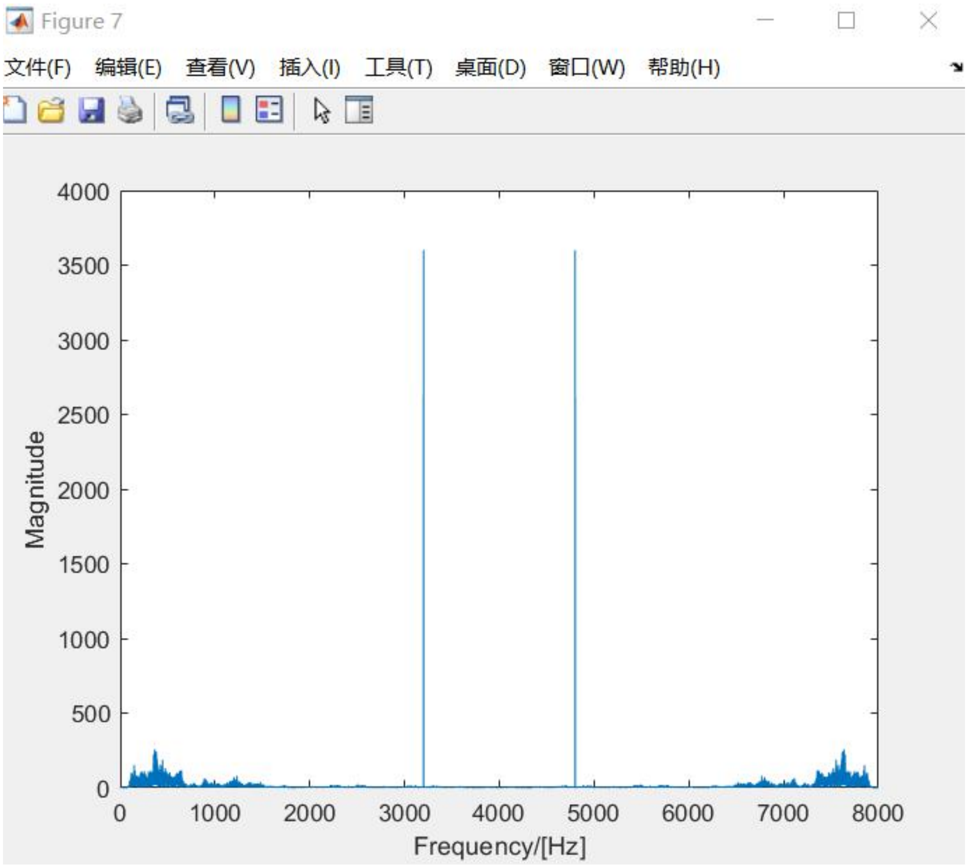
**Plot(f,abs(fft(y)))** **% discrete Fourier transform of x**

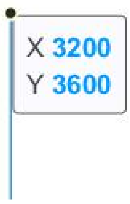
**xlabel('frequency f / [Hz]');** **% label axes**

**ylabel('magnitude');**

### Provide plots for the time domain segment in Q4.1 and the magnitude spectrum in

```
%Provide plots for the time domain segment in Q4.1 and the magnitude spectrum in Q4.2
figure(7);
fs=8000;
f=(0:length(y)-1)/length(y)*fs;
plot(f,abs(fft(y)));
xlabel('Frequency/[Hz]');
ylabel('Magnitude');
```




*So we can see that:* $f_0 = 3200$

*and then calculate the normalized angular frequency:* $\Omega_0 = 2\pi \frac{f_0}{f_s} = 2\pi \frac{3200}{8000} \approx 2.5133$

## Fourier Transform

- Fourier transform of $x_s(t)$:

$$X_s(j\omega) = \int_{-\infty}^{\infty} x_s(t) e^{-j\omega t} dt = \int_{n-\infty}^{\infty} \sum_{n=-\infty}^{\infty} x[n]\delta(t - nT_s) e^{-j\omega t} dt$$

$$= \sum_{n=-\infty}^{\infty} x[n] \int_{n-\infty}^{\infty} \delta(t - nT_s) e^{-j\omega t} dt \quad (161)$$

$$= \sum_{n=-\infty}^{\infty} x[n] e^{-j\omega n T_s} \quad (162)$$

with the step from (161) to (162) exploiting the sifting property of $\delta(t)$;

- we define the *normalised angular frequency*

$$\Omega = \omega T_s = 2\pi \frac{f}{f_s} = 2\pi \frac{\omega}{\omega_s} \quad (163)$$

- note that the normalised angular sampling frequency $\Omega_s = 2\pi$, which defines the periodicity of the spectrum.

**Q5.1: Determine the denominator and numerator coeffiffifficients ai and bi, i = 0, 1, 2 when written as**

To suppress the sinusoidal interference, we want to utilise an IIR notch filter with transfer function

$$Q(z) = \frac{(1 - e^{j\Omega_0} z^{-1})(1 - e^{-j\Omega_0} z^{-1})}{(1 - \rho e^{j\Omega_0} z^{-1})(1 - \rho e^{-j\Omega_0} z^{-1})}$$

to operate on $y_f[n]$ and generate an output $y_{ff}[n]$. For implementations and analysis below, please assume a value $\rho = 0.9$.

$$Q(z) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2}}{a_0 + a_1 z^{-1} + a_2 z^{-2}}$$

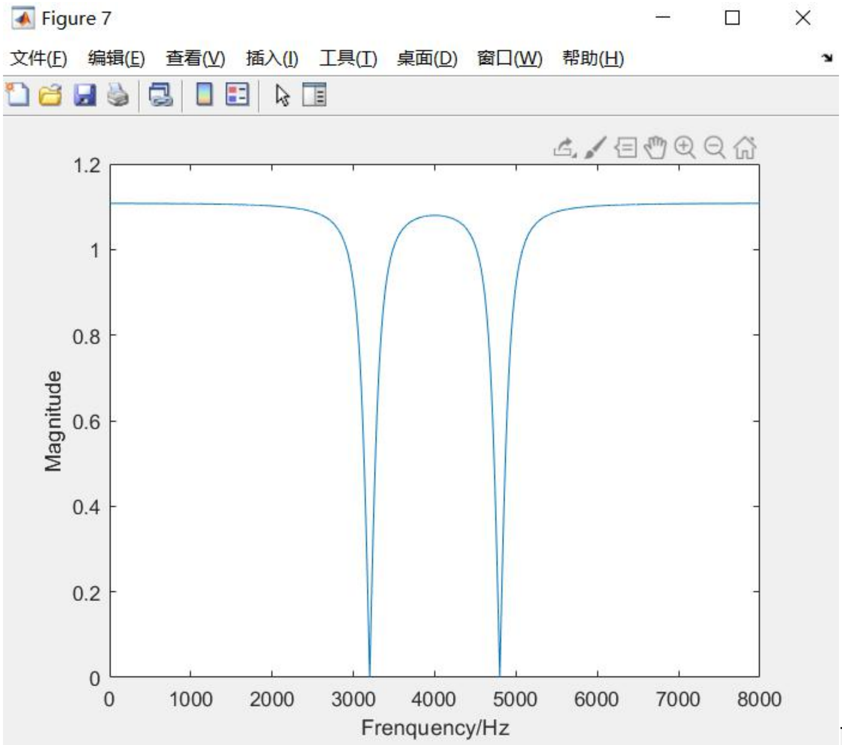
**and plot the magnitude response |H2(ejΩ)| in Matlab;**

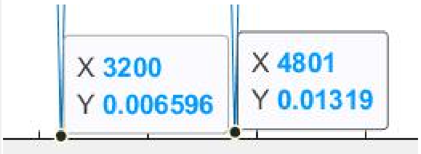
$$Q(z) = \frac{(1 - e^{j\Omega_0} z^{-1})(1 - e^{-j\Omega_0} z^{-1})}{(1 - \rho e^{j\Omega_0} z^{-1})(1 - \rho e^{-j\Omega_0} z^{-1})} \quad (\rho = 0.9)$$

$$= \frac{1 + z^{-2} - e^{j\Omega_0} z^{-1} - e^{-j\Omega_0} z^{-1}}{1 + \rho^2 z^{-2} - \rho e^{-j\Omega_0} z^{-1} - \rho e^{j\Omega_0} z^{-1}}$$

$$= \frac{1 - 2\cos\Omega_0 z^{-1} + z^{-2}}{1 - 1.8\cos\Omega_0 z^{-1} + 0.81 z^{-2}}$$

```
%Plot the magnitude response |H2(ejΩ)| in Matlab
f1=(0:0.0001:1)*2*pi;
c=cos(3200*2*pi/8000);
H2=(1-2*c*exp(f1*i*(-1))+exp(i*f1*(-2)))./(1-1.8*c*exp(i*(-1)*f1)+0.81*exp(i*f1*(-2)));
f=(0:length(H2)-1)/length(H2)*fs;
plot(f,abs(H2));
xlabel('Frenquency/Hz');
ylabel('Magnitude');
```






We can see that when the frequency is 3200 and 4801 the magnitude is zero.

**Q5.2: with the Matlab coeffiffifficient vectors a and b containing the coeffiffifficients determined in (2), fifilter your music signal y with the IIR notch fifilter, and listen to it:**

```
yff = filter(b,a,yf);        % yf is the input of filter q
sound(yff,8000);             % play back the filtered signal
```

**Is the sinusoidal interference suppressed in yff?**

We can know that:

a=[1 -18*cos(2*pi*3200/8000) 0.81]

b=[1 -2*cos(2*pi*3200/8000) 1]

From the yff signal we can see that the sinusoidal interference has been reduced in yff without noise.

```
%fifilter music signal y with the IIR notch fifilter
a=[1 -1.8*cos(2*pi*3200/8000) 0.81];
b=[1 -2*cos(2*pi*3200/8000) 1];
yff = filter(b,a,yf);          % yf is the input of filter q
sound(yff,8000);               % play back the filtered signal

figure(8);
fs=8000;
f=(0:length(yff)-1)/length(yff)*fs;
plot(f,abs(fft(yff)));
xlabel('Frequency/[Hz]');
ylabel('Magnitude');

figure(9);
fs=8000;
f=(0:length(yf)-1)/length(yf)*fs;
plot(f,abs(fft(yf)));
xlabel('Frequency/[Hz]');
ylabel('Magnitude');

figure(10);
stem(yf);

figure(11);
stem(yff);
```

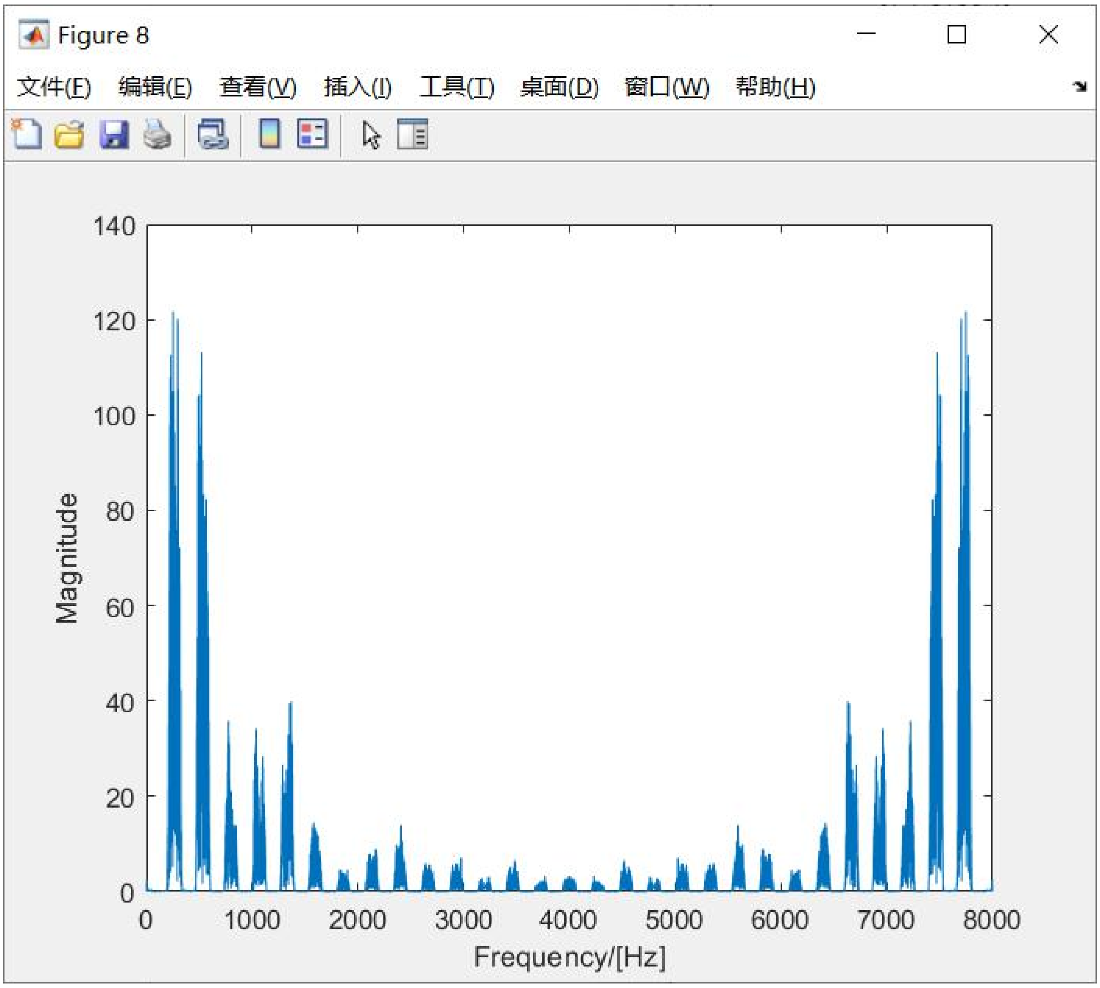


Figure 8
文件(F) 编辑(E) 查看(V) 插入(I) 工具(T) 桌面(D) 窗口(W) 帮助(H)
Magnitude
Frequency/[Hz]
0
20
40
60
80
100
120
140
0 1000 2000 3000 4000 5000 6000 7000 8000

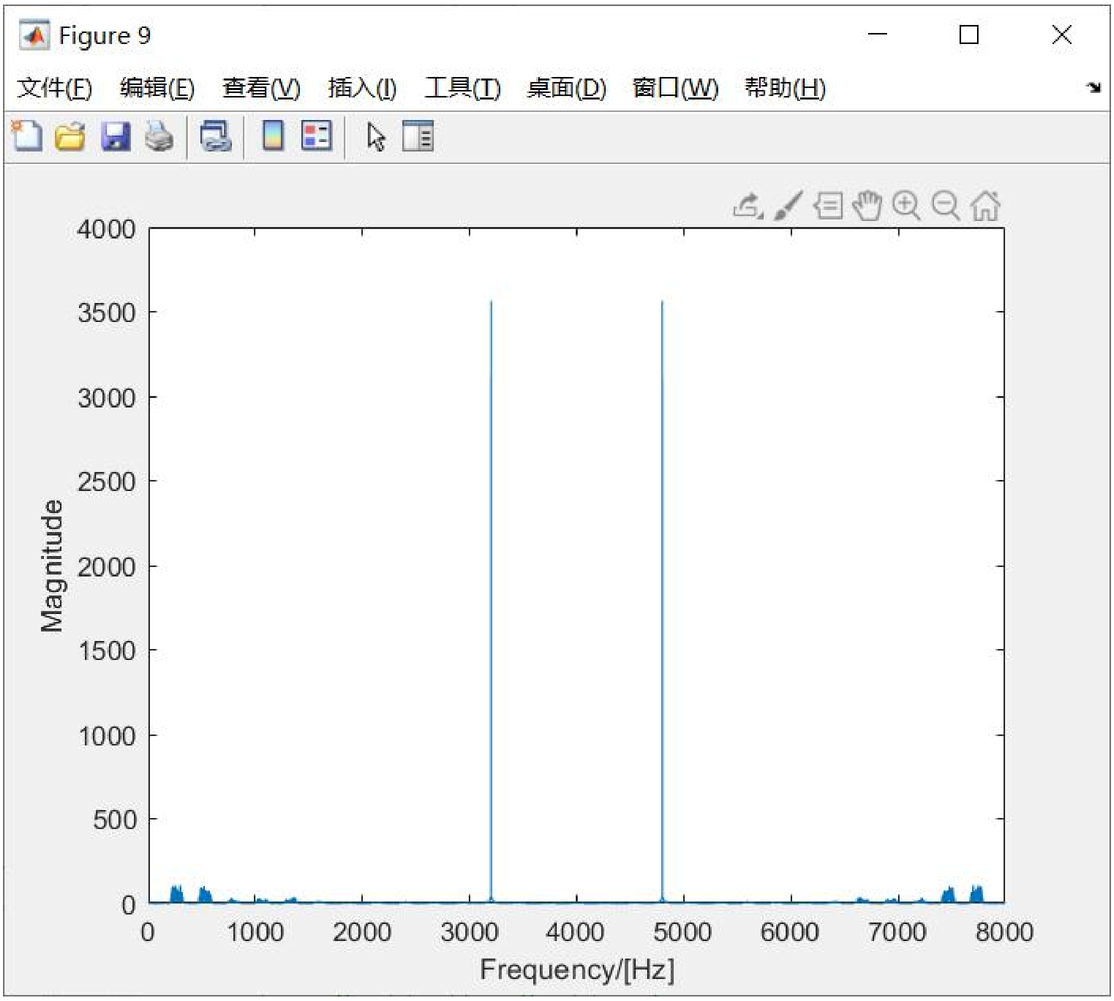


Figure 9
文件(F) 编辑(E) 查看(V) 插入(I) 工具(T) 桌面(D) 窗口(W) 帮助(H)
Magnitude
Frequency/[Hz]
0
500
1000
1500
2000
2500
3000
3500
4000
0 1000 2000 3000 4000 5000 6000 7000 8000

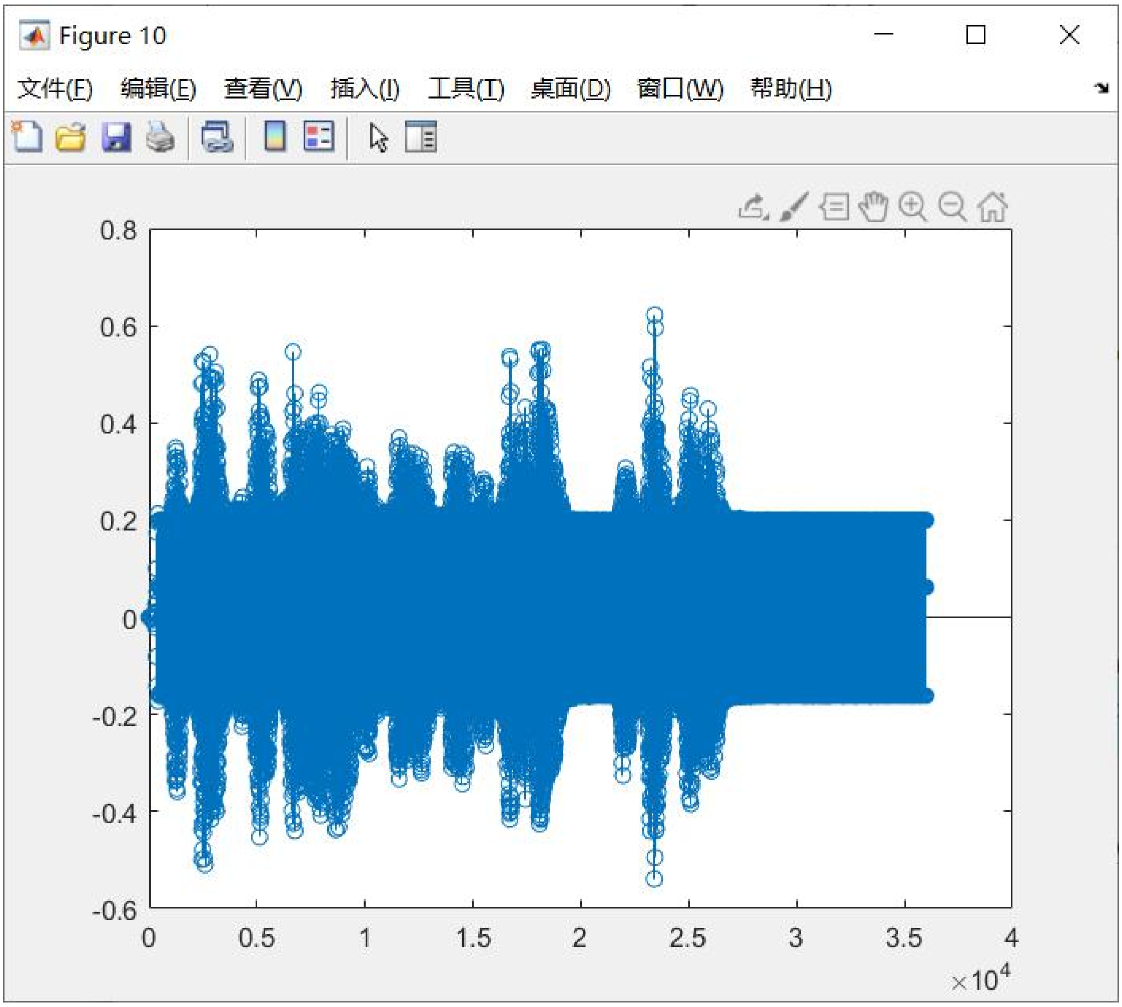


Figure 10
文件(F) 编辑(E) 查看(V) 插入(I) 工具(T) 桌面(D) 窗口(W) 帮助(H)
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
0 0.5 1 1.5 2 2.5 3 3.5 4
$\times 10^4$

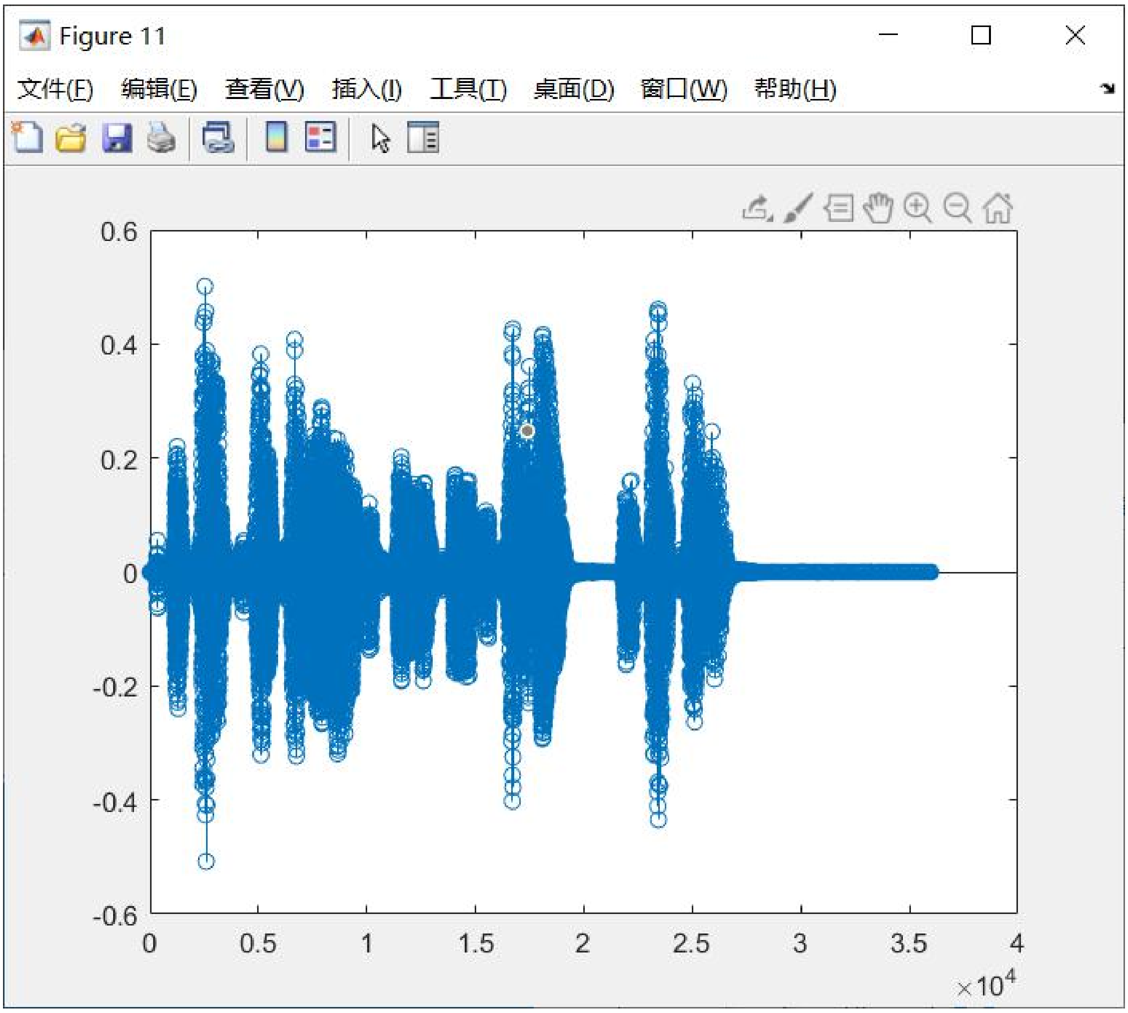
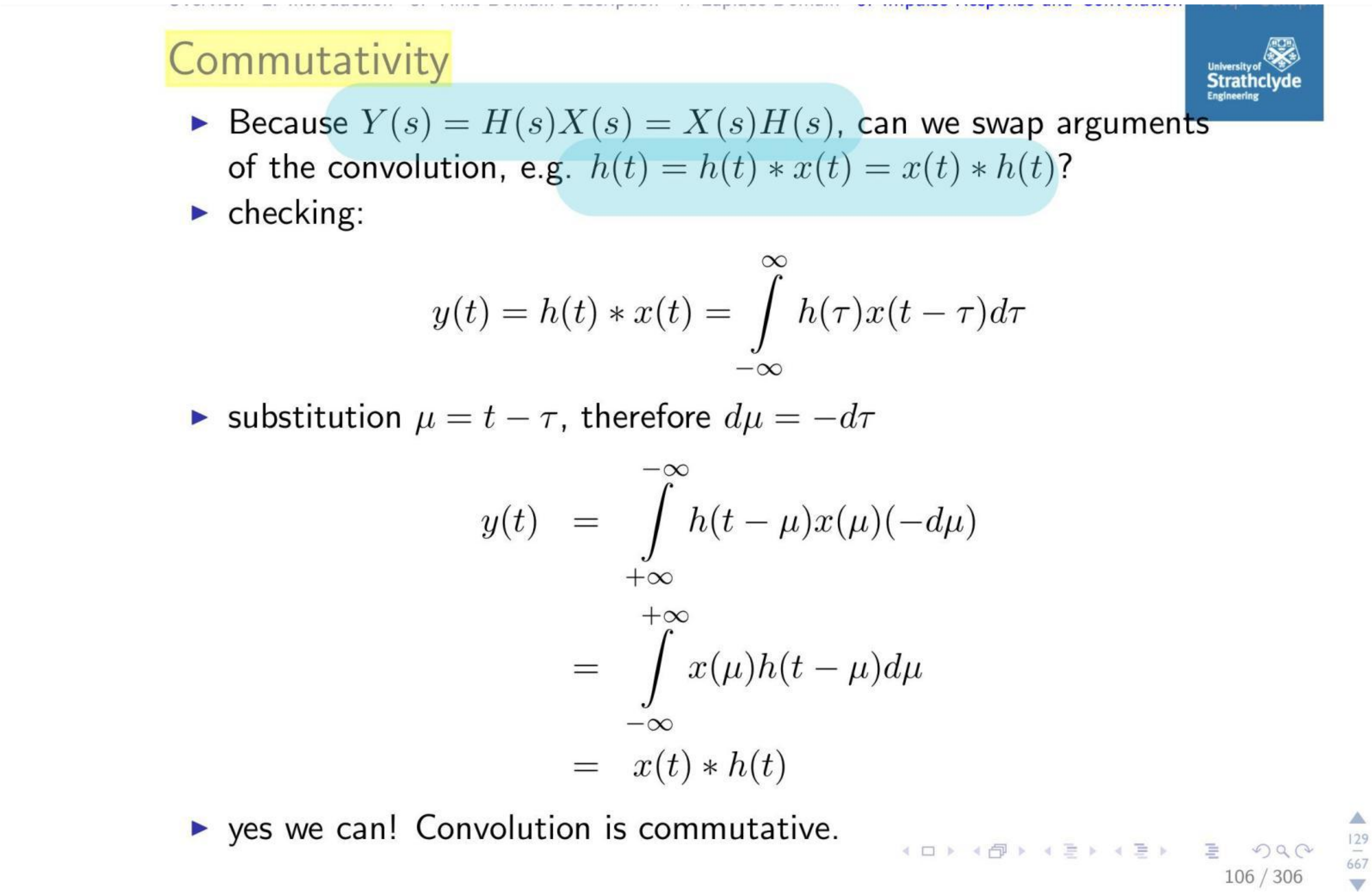


Figure 11
文件(F) 编辑(E) 查看(V) 插入(I) 工具(T) 桌面(D) 窗口(W) 帮助(H)
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
0 0.5 1 1.5 2 2.5 3 3.5 4
$\times 10^4$

**Q5.3: Can you swap the fifilters h[n] and q[n] without affffecting the overall performance? Is so, why?**

*Yes, the reason can be shown from the lecture slide.*

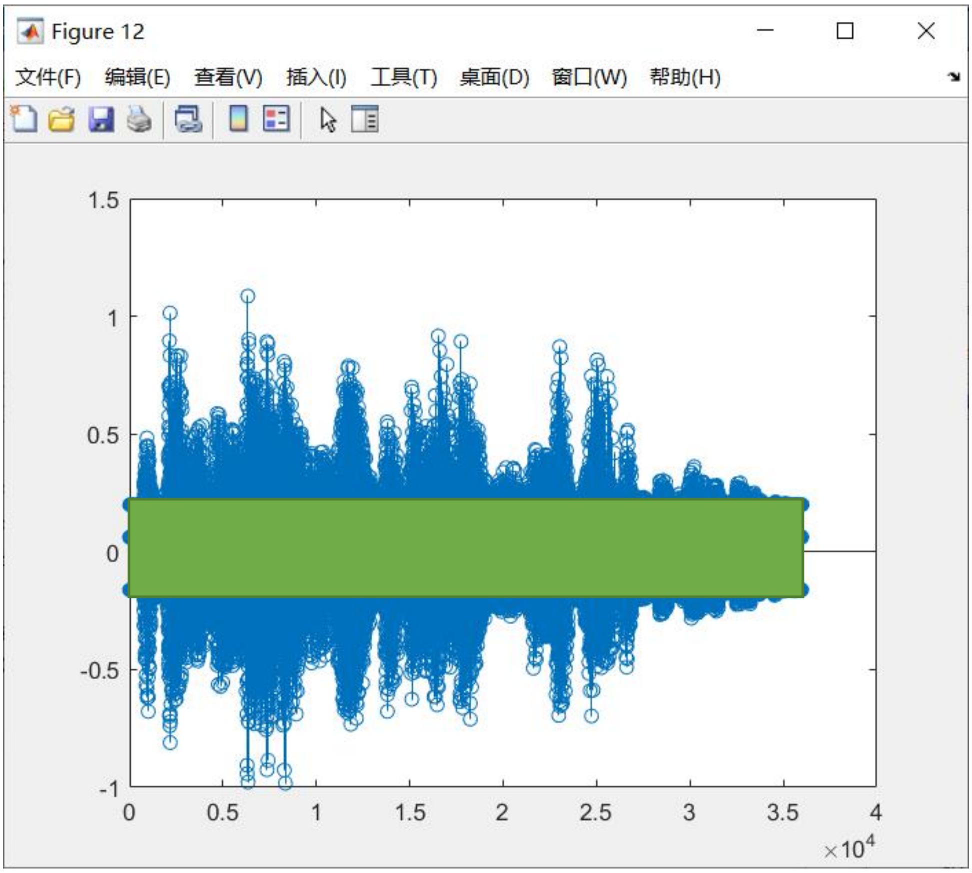



**Q5.4: Could you attenuate the interference without using q[n], and without further deteriorating Prof Stewart's voice?**

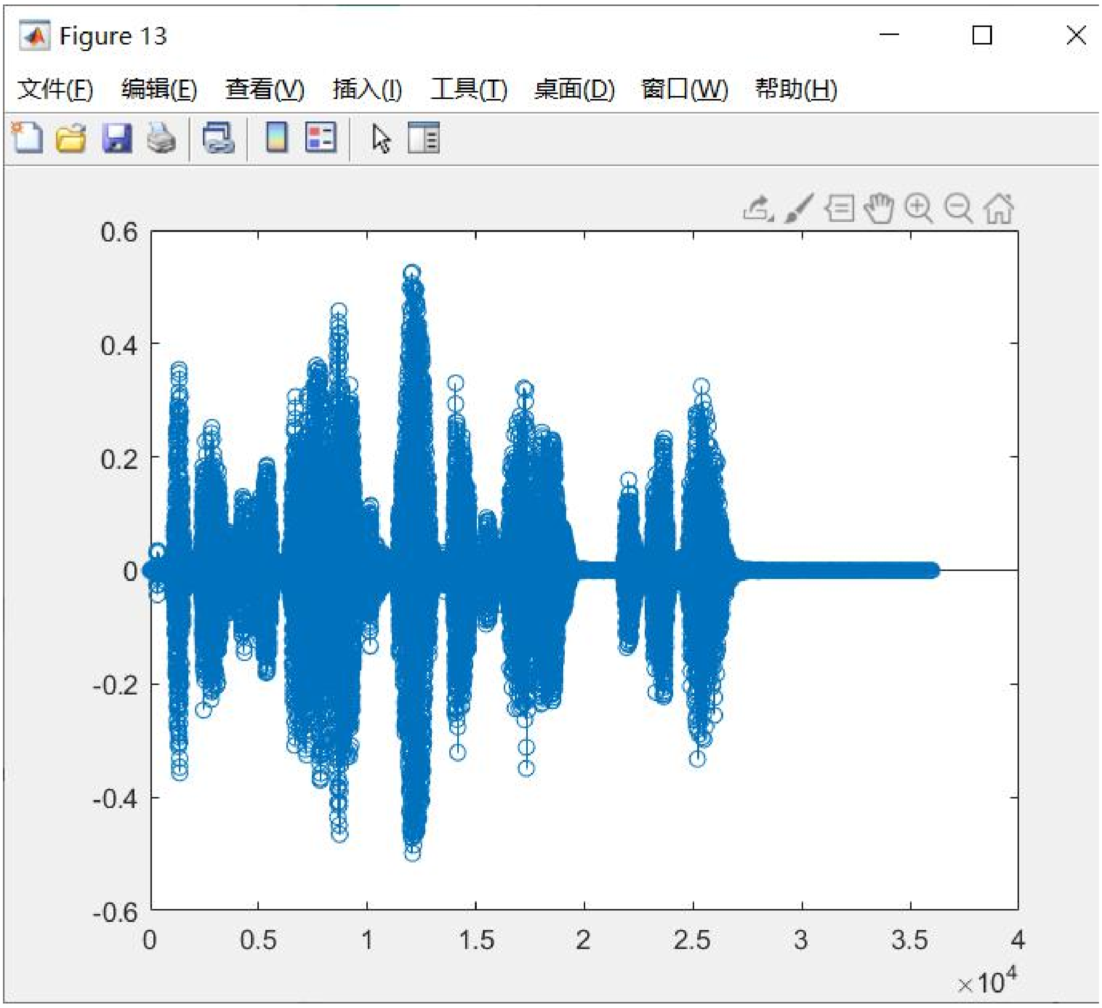
*We can know that the g[n] can not allow the noise go. So we can use the filter g[n] to attenuate the noise.*

```
%Q5.5 without q[n] to attenuate the noise
x=audioread('FarEndSignal.wav');
xf=filter(h,1,x);
[y,N]=room(xf,201920667);
yf=filter(g,1,y);
sound(yf,8000);

figure(12);
stem(y);

figure(13);
stem(yf);
```

*And the figure (12) is the y[n] which is not after filter.*




We can see that the figure(12) before the filter, has noise which is the be circled by green pen. But after the filter g[n] this changes to the figure(13) :



The noise part is not like the one in figure(12), it becomes smaller. This because when convolution with g[n], when the frequency equal 3200 the magnitude response is zero.

26, Jan 2020