# CMPE 362 - SIGNAL PROCESSING

## HOMEWORK 2

## REPORT

**Problem 1**

① Part A

$$x(t) = \begin{cases} \dfrac{2t}{T_0} & 0 \le t < \dfrac{T_0}{2} \\ \dfrac{2(T_0 - t)}{T_0} & \dfrac{T_0}{2} < t < T_0 \end{cases}$$

$$a_k = \frac{1}{T_0}\int_0^{T_0/2} \left(\frac{2t}{T_0}\right) e^{-j\left(\frac{2\pi}{T_0}\right)kt}\,dt + \frac{1}{T_0}\int_{T_0/2}^{T_0} \left(\frac{2(T_0-t)}{T_0}\right) e^{-j\left(\frac{2\pi}{T_0}\right)kt}\,dt$$

$$= \frac{2}{T_0^2}\int_0^{T_0/2} t\, e^{-j\left(\frac{2\pi}{T_0}\right)kt}\,dt + \frac{2}{T_0^2}\int_{T_0/2}^{T_0} (T_0-t)\, e^{-j\left(\frac{2\pi}{T_0}\right)kt}\,dt$$

$$= \frac{2}{T_0^2}\left[\frac{e^{-j\left(\frac{2\pi}{T_0}\right)kt}}{-j\left(\frac{2\pi}{T_0}\right)k}\left(t - \frac{1}{-j\left(\frac{2\pi}{T_0}\right)k}\right)\Bigg|_0^{T_0/2}\right] + \frac{2}{T_0^2}\left[\frac{e^{-j\left(\frac{2\pi}{T_0}\right)kt}}{-j\left(\frac{2\pi}{T_0}\right)k}\left((T_0-t) - \frac{1}{-j\left(\frac{2\pi}{T_0}\right)k}\right)\right]_{T_0/2}^{T_0}$$

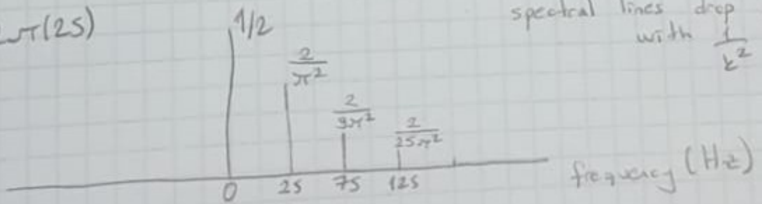
$$= \frac{-e^{-jk\pi}\left(-1 + e^{jk\pi} - jk\pi\right)}{2k^2\pi^2} + \frac{e^{-2jk\pi}\left(-1 + e^{jk\pi}(1 - jk\pi)\right)}{2k^2\pi^2}$$

$$= \frac{-e^{-jk\pi}\left(-1 + \cos[k\pi]\right)}{k^2\pi^2} \qquad e^{-jk\pi} = (-1)^k \qquad \cos(k\pi) = (-1)^k$$

$$a_k = \frac{(-1)^k\left[(-1)^k - 1\right]}{k^2\pi^2} \qquad a_0 = \frac{1}{T_0}\int_0^{T_0} x(t)\,dt = \frac{1}{T_0} \times \text{Area} = \frac{1}{T_0}\cdot\frac{T_0}{2} = \frac{1}{2}$$

$$a_k = \begin{cases} \dfrac{2}{k^2\pi^2}, & k = 1, 3, 5, \ldots \\ 0, & k = 2, 4, 6, \ldots \\ \dfrac{1}{2} & k = 0 \end{cases} \qquad x(t) = \frac{1}{2} + \frac{2}{\pi^2} + \frac{2}{9\pi^2} + \frac{2}{25\pi^2} + \ldots$$

$$w = \frac{2\pi}{0.04} = 2\pi(25)$$



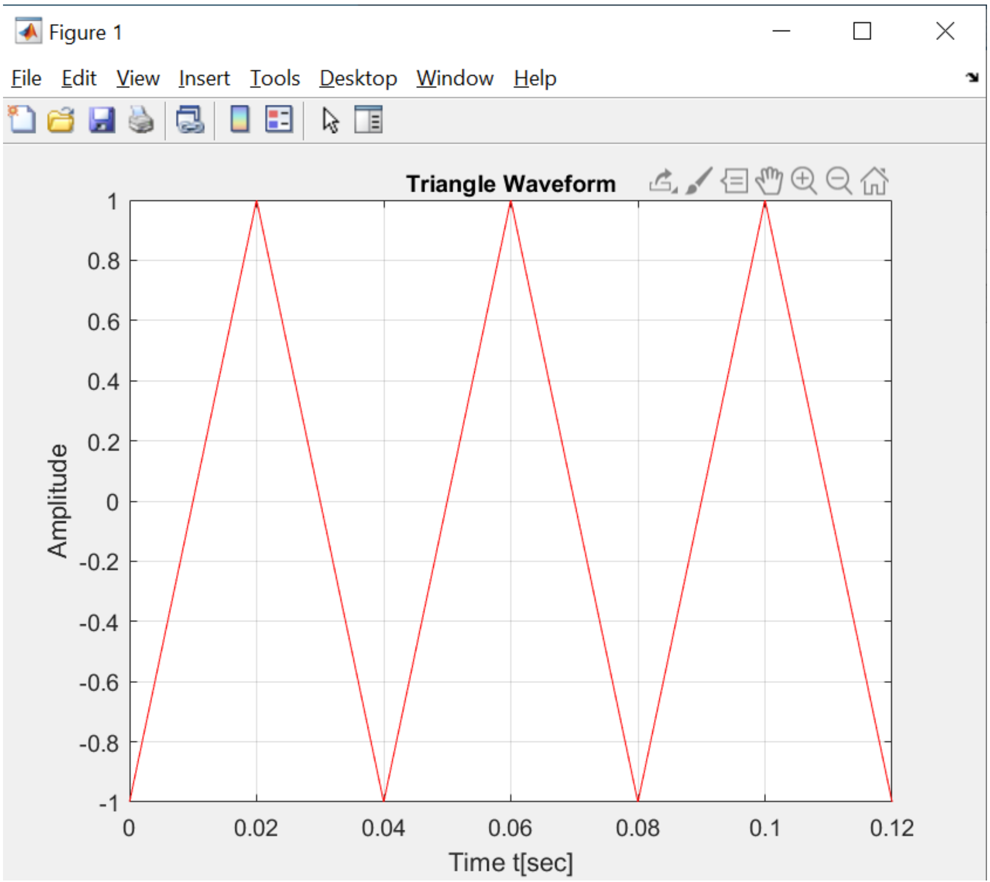

spectral lines drop off with $\frac{1}{k^2}$

**Part B**

```
%Construct triangular wave with sawtooth function
fs = 1000;  % sampling frequency
t = 0:1/fs:0.12;
x = sawtooth(2*pi*25*t,1/2);
plot(t,x,'r')
xlabel('Time t[sec]')
ylabel('Amplitude')
title('Triangle Waveform', 'FontSize', 10);
grid on

%Take fourier transform
fftSignal = fft(x);
l=length(fftSignal);
t=linspace(0,l/fs,l);
%apply fftshift
fftSignal = fftshift(fftSignal);

%calculate the frequency axis
f0 = (-l/2:l/2-1)*(fs/l); % 0-centered frequency range
figure();
plot(f0,abs(fftSignal))
ylabel('Time')
xlabel('Frequency')
```



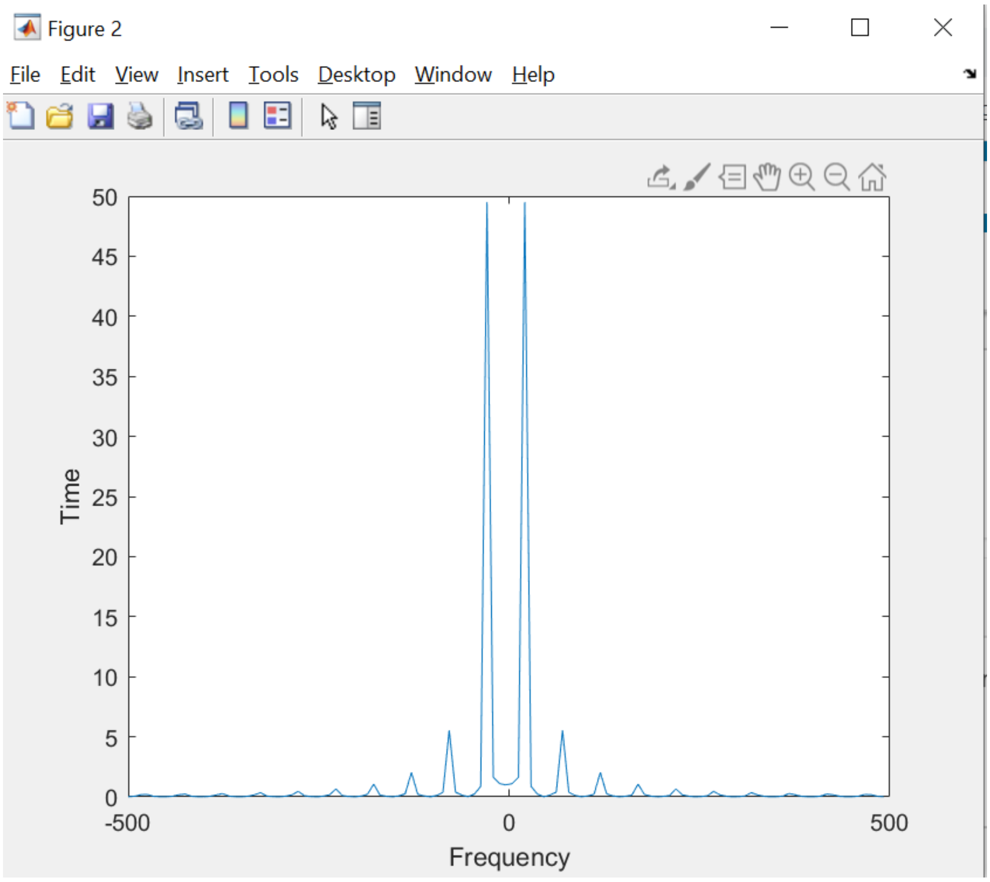
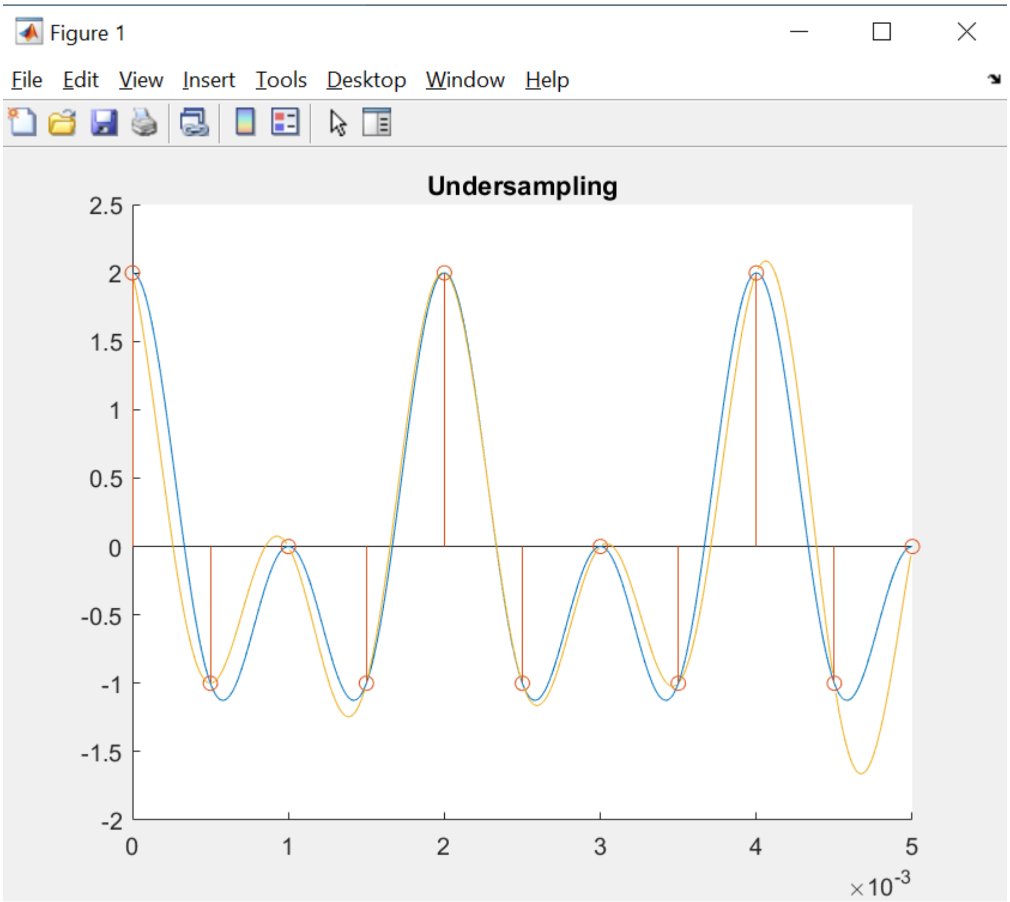


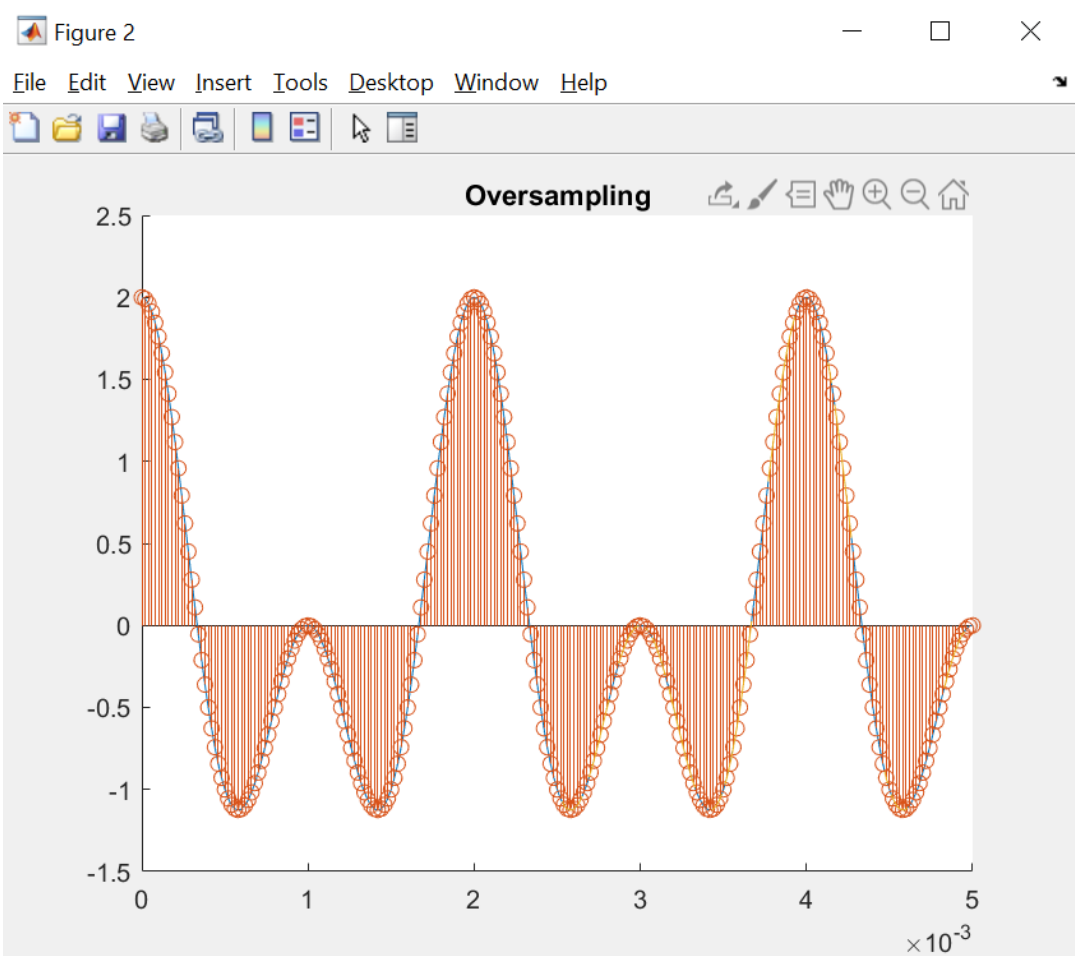
Figure 2
File Edit View Insert Tools Desktop Window Help
50
45
40
35
30
25
20
15
10
5
0
Time
-500
0
500
Frequency

## Problem 2

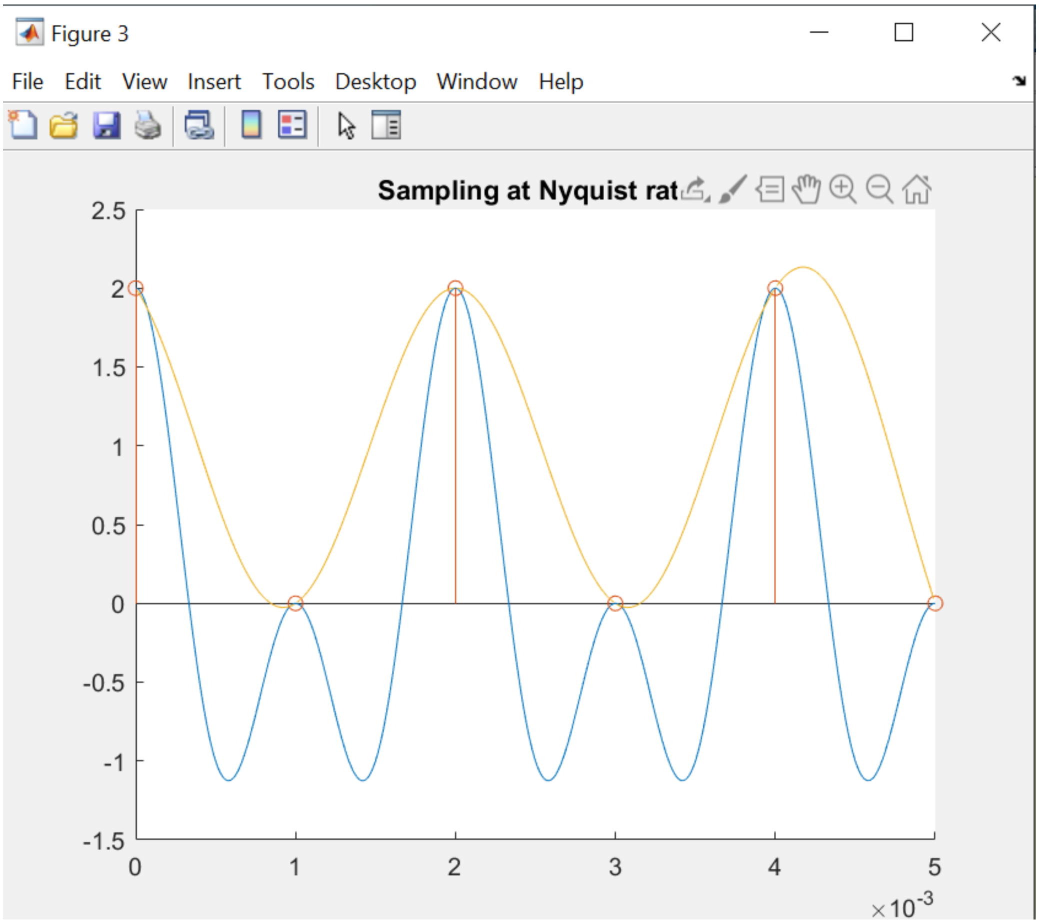
```matlab
%% Undersampling
F = 1000;      % frequency of signal
Fs = 2*F;    % sampling frequency
Ts = 1/Fs;  % sampling period
%Generate signals and sampling
tc = 0:0.00001:5/F;         % axis
y1=cos(pi*F*tc);  %first signal
y2=cos(2*pi*F*tc); %second signal
yc = y1+y2;     % sum of Continous time signals
td = 0:Ts:5/F;  %axis
y1=cos(pi*F*td);% first signal
y2=cos(2*pi*F*td); %second signal
yd = y1+y2;     % sum of discrete time signals
L = length(td);         % number of samples
% Reconstruction by using the formula:
Recns = zeros(size(tc));
sinc_train = zeros(L,length(tc));
for t = 1:length(tc)
    for i = 0:L-1
        % sinc(x) = sin(pi*x)/(pi*x) according to MATLAB
        sinc_train(i+1,:) = sin(pi*(tc-i*Ts)/Ts)./(pi*(tc-i*Ts)/Ts);
        Recns(t) = Recns(t) + yd(i+1)*sin(pi*(tc(t)-i*Ts)/Ts)/(pi*(tc(t)-i*Ts)/Ts);
    end
end
%Plot the signals
figure();title('Undersampling');
hold on
plot(tc,yc);stem(td,yd);plot(tc,Recns);
hold off
```



```
%% Oversampling
F = 1000;      % frequency of signal
Fs = 50*F;    % Increase the ratio of sampling frequency
Ts = 1/Fs;  % sampling period
%Generate signals and sampling
tc = 0:0.00001:5/F;         % axis
y1=cos(pi*F*tc);  %first signal
y2=cos(2*pi*F*tc); %second signal
yc = y1+y2;     % sum of Continous time signals
td = 0:Ts:5/F;  %axis
y1=cos(pi*F*td);% first signal
y2=cos(2*pi*F*td); %second signal
yd = y1+y2;     % sum of discrete time signals
L = length(td);         % number of samples
% Reconstruction by using the formula:
Recns = zeros(size(tc));
sinc_train = zeros(L,length(tc));
for t = 1:length(tc)
    for i = 0:L-1
        % sinc(x) = sin(pi*x)/(pi*x) according to MATLAB
        sinc_train(i+1,:) = sin(pi*(tc-i*Ts)/Ts)./(pi*(tc-i*Ts)/Ts);
        Recns(t) = Recns(t) + yd(i+1)*sin(pi*(tc(t)-i*Ts)/Ts)/(pi*(tc(t)-i*Ts)/Ts);
    end
end
figure();title('Oversampling');
hold on
plot(tc,yc);stem(td,yd); plot(tc,Recns);
hold off
```



```
%% Sampling at Nyquist rate
F = 1000;      % frequency of signal
Fs = 2*F;    % sampling frequency
Fn= Fs/2;    %nyquist frequency
Ts = 1/Fn;   % sampling period
%Generate signals and sampling
tc = 0:0.00001:5/F;          % axis
y1=cos(pi*F*tc);   %first signal
y2=cos(2*pi*F*tc); %second signal
yc = y1+y2;     % sum of Continous time signals
td = 0:Ts:5/F;   %axis
y1=cos(pi*F*td);% first signal
y2=cos(2*pi*F*td); %second signal
yd = y1+y2;     % sum of discrete time signals
L = length(td);          % number of samples
% Reconstruction by using the formula:
Recns = zeros(size(tc));
sinc_train = zeros(L,length(tc));
for t = 1:length(tc)
    for i = 0:L-1
        % sinc(x) = sin(pi*x)/(pi*x) according to MATLAB
        sinc_train(i+1,:) = sin(pi*(tc-i*Ts)/Ts)./(pi*(tc-i*Ts)/Ts);
        Recns(t) = Recns(t) + yd(i+1)*sin(pi*(tc(t)-i*Ts)/Ts)/(pi*(tc(t)-i*Ts)/Ts);
    end
end
%Plot the signals
figure();title('Sampling at Nyquist rate');
hold on
plot(tc,yc);stem(td,yd);plot(tc,Recns);
hold off
```



## Problem 3

```matlab
%load noise audio file
[y,Fs]= audioread('noise_p232_090.wav');
info = audioinfo('noise_p232_090.wav');
t = 0:seconds(1/Fs):seconds(info.Duration);
t = t(1:end-1);
subplot(2,1,1); plot(t,y); %plot the noise signal
title('noise signal');

%finding fft of noise audio
L=length(y); %size of the longest dimension of noise
NEFT = 2^nextpow2(L); %returns the smallest power of two
Y=abs(fft(y,NEFT)); %absolute value of fft
freq = Fs/2*linspace(0,1,NEFT/2+1); %linspace:generate logarithmically spaced values
subplot(2,1,2); plot(freq, Y(1:length(freq))) %plot the fft
title('noise fft')
```

```matlab
%load clean audio file
[clean_y,clean_Fs]= audioread('clean_p232_090.wav');
clean_info = audioinfo('clean_p232_090.wav');
clean_t = 0:seconds(1/clean_Fs):seconds(info.Duration);
clean_t = clean_t(1:end-1);
figure();
subplot(2,1,1); plot(clean_t,clean_y); %plot the clean signal
title('clean signal');

%finding fft of clean audio
clean_L=length(clean_y); %size of the longest dimension of clean
clean_NEFT = 2^nextpow2(clean_L); %returns the smallest power of two
clean_Y=abs(fft(clean_y,clean_NEFT));
clean_freq = clean_Fs/2*linspace(0,1,clean_NEFT/2+1); %linspace:generate logarithmically spaced values
subplot(2,1,2); plot(clean_freq, clean_Y(1:length(clean_freq))) %plot the fft
title('clean fft')
```

```matlab
%finding the time-response by using the functions ifft and ifftshift
F= clean_Y(1:length(clean_freq))./ Y(1:length(freq)); %FFT(clean signal)/FFT(noisy signal)
F=ifft(F,'symmetric'); %shift
%use circular convolution(cconv)
new=cconv(F,clean_Y(1:length(clean_freq)),length(Y(1:length(freq))));
%cconv(F,clean,length(noise));

new_L=length(clean_y); %size of the longest dimension of clean
new_NEFT = 2^nextpow2(new_L); %returns the smallest power of two
new_freq = clean_Fs/2*linspace(0,1,new_NEFT/2+1); %linspace:generate logarithmically spaced values
figure(); plot(new_freq, ifftshift(new(1:length(new_freq)))); %plot inverse zero-frequency shift
title('new signal')
```

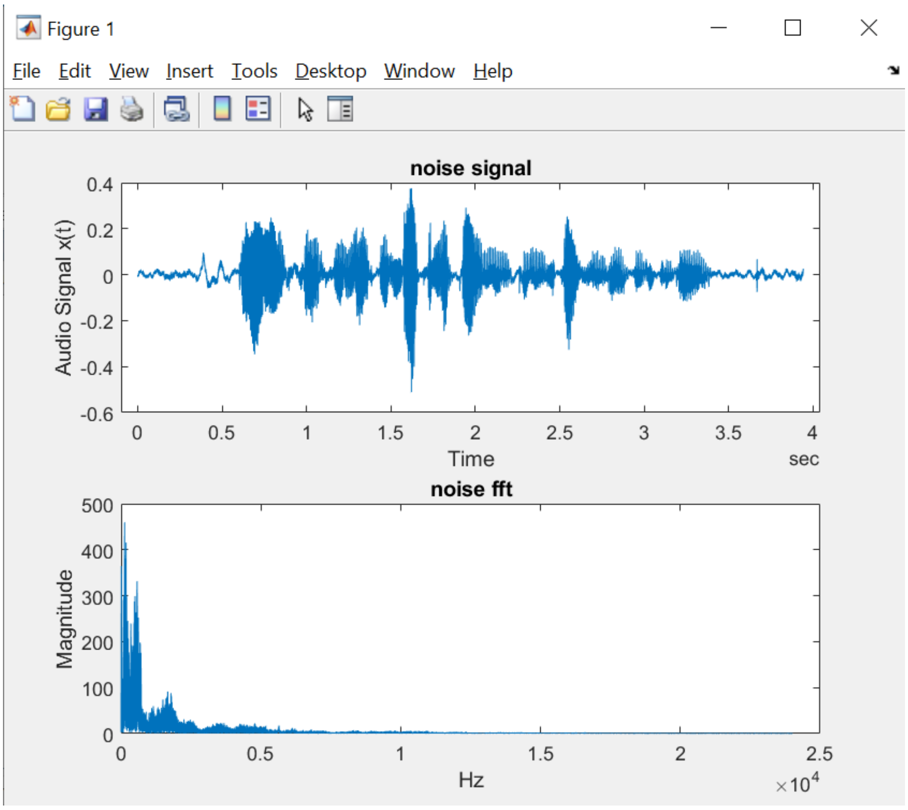


Figure 1
File Edit View Insert Tools Desktop Window Help
noise signal
Audio Signal x(t)
Time
sec
noise fft
Magnitude
Hz
$\times 10^4$

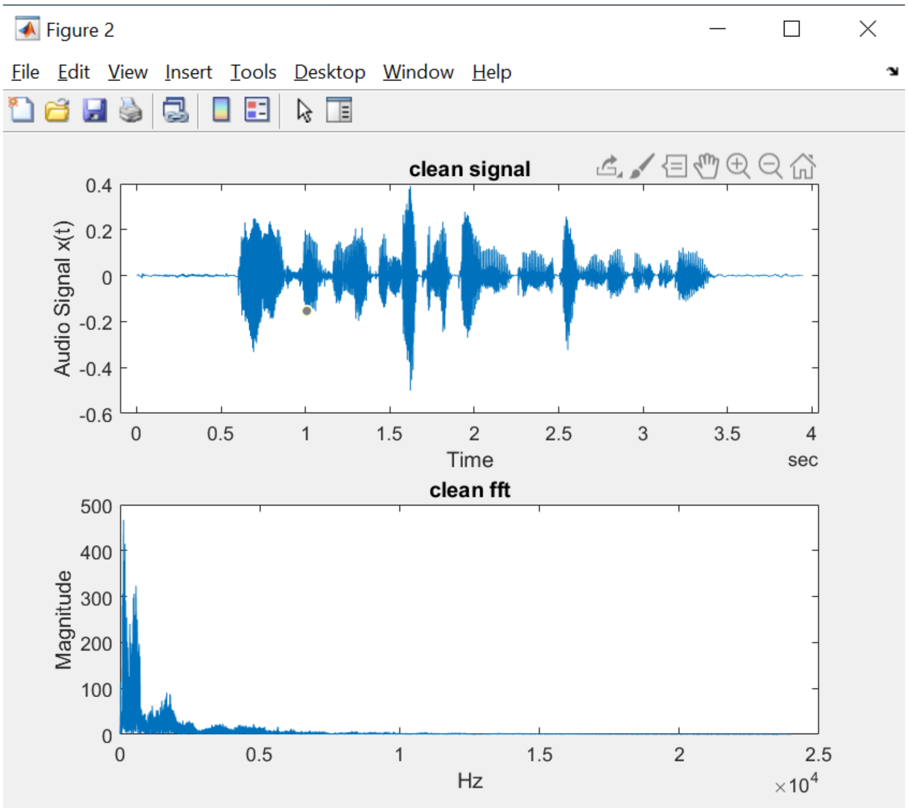
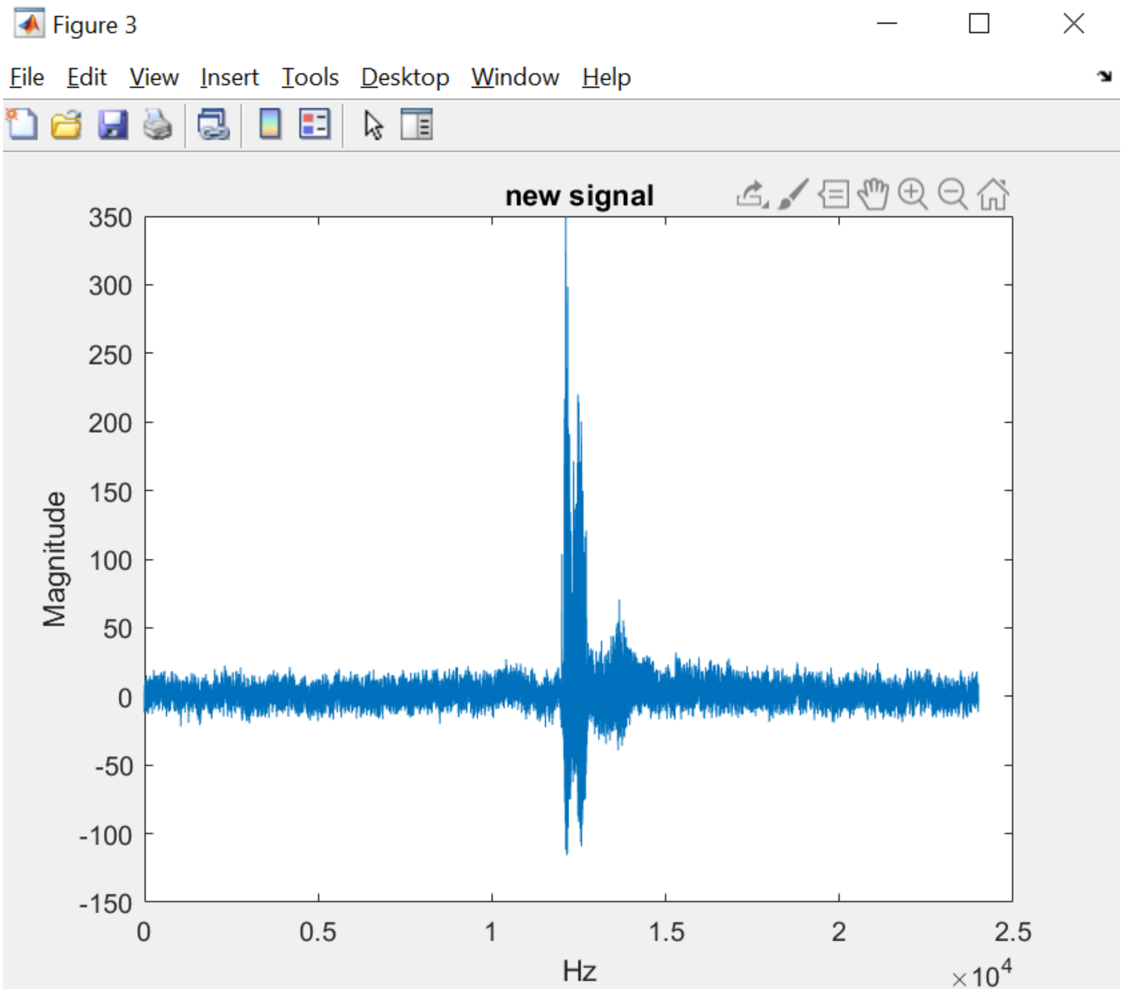


Figure 2
File Edit View Insert Tools Desktop Window Help
clean signal
Audio Signal x(t)
Time
sec
clean fft
Magnitude
Hz
$\times 10^4$




**Ali Batır**

**2015400261**