Alperen Bağ
2018400279

03.05.2020

CMPE 362 Homework 2

1.a.-

$$\bullet\; x(t) = \begin{cases} \dfrac{-t}{0.02} & \text{for } -0.02 \le t < 0 \\ \dfrac{t}{0.02} & \text{for } 0 \le t < 0.02 \end{cases} \longrightarrow \text{triangle wave eqn.}$$

$(T_0 = 0.04s)$

$\bullet$ Fourier Analysis Integral $\longrightarrow$ $a_k = \dfrac{1}{T_0} \displaystyle\int_0^{T_0} x(t) e^{-j\left(\frac{2\pi}{T_0}\right)kt} dt$

(We can shift the integral as long as the interval is T_0) $a_k = \dfrac{1}{T_0} \displaystyle\int_{-T_0/2}^{T_0/2} x(t) e^{-j\left(\frac{2\pi}{T_0}\right)kt} dt$

$$a_k = \frac{2}{T_0^2}\left[\int_{-T_0/2}^{0} (-t) e^{-jw_0kt} dt + \int_0^{T_0/2} (t) e^{-jw_0kt} dt\right]$$

$$\int t.e^{-jw_0kt} dt = \frac{t.e^{-jw_0kt}}{(-jw_0k)} - \int \frac{e^{-jw_0kt}}{(-jw_0k)} dt = \frac{t.e^{-jw_0kt} - \overbrace{\int e^{-jw_0kt} dt}^{\frac{e^{-jw_0kt}}{(-jw_0k)}}}{(-jw_0k)}$$

$u = t$	$du = dt$
$dv = e^{-jw_0kt} dt$	$v = \dfrac{e^{-jw_0kt}}{(-jw_0k)}$

$\boxed{(e^{-jw_0kt} = X) \text{ for simplicity}}$

$$= \frac{t.X - \frac{X}{(-jw_0k)}}{(-jw_0k)}$$

$$= \frac{t.X}{(-jw_0k)} - \frac{X}{(-jw_0k)^2} = \frac{-t.X}{jw_0k} - \frac{X}{(jw_0k)^2} = \frac{-X(jw_0kt + 1)}{(jw_0k)^2}$$

$$= \boxed{\frac{e^{-jw_0kt}(jw_0kt + 1)}{(w_0k)^2}} \qquad \boxed{w_0 = \frac{2\pi}{T_0}}$$

$$\int_0^{T_0/2} t.e^{-jw_0kt} dt = \frac{e^{-j\pi k}(j\pi k + 1) - 1}{(w_0k)^2}$$

$$\int_{-T_0/2}^{0} (-t) e^{-jw_0kt} dt = \frac{-[e^{j\pi k}(1 - j\pi k)] - 1}{(w_0k)^2}$$

$+$

$$= \frac{e^{-j\pi k} j\pi k + e^{-j\pi k} - 1 + e^{j\pi k} - e^{j\pi k} j\pi k - 1}{(w_0k)^2}$$

$$= \frac{j\pi k(e^{-j\pi k} - e^{j\pi k}) + e^{j\pi k} + e^{-j\pi k} - 2}{\frac{4\pi^2k^2}{T_0^2}}$$

$$a_k = \frac{\pi k \underbrace{\cancel{\sin(\pi k)}}_{\to 0 \text{ (for integer } k)} + \cos(\pi k) - 1}{\pi^2 k^2} = \boxed{\frac{\cos(\pi k) - 1}{\pi^2 k^2}}$$

$$\boxed{\begin{aligned} e^{j\pi k} &= \cos\pi k + j\sin\pi k \\ e^{-j\pi k} &= \cos(-\pi k) + j\sin(-\pi k) = \cos\pi k - j\sin(\pi k) \end{aligned}}$$

Alperen Bağ
2018400279

03.05.2020

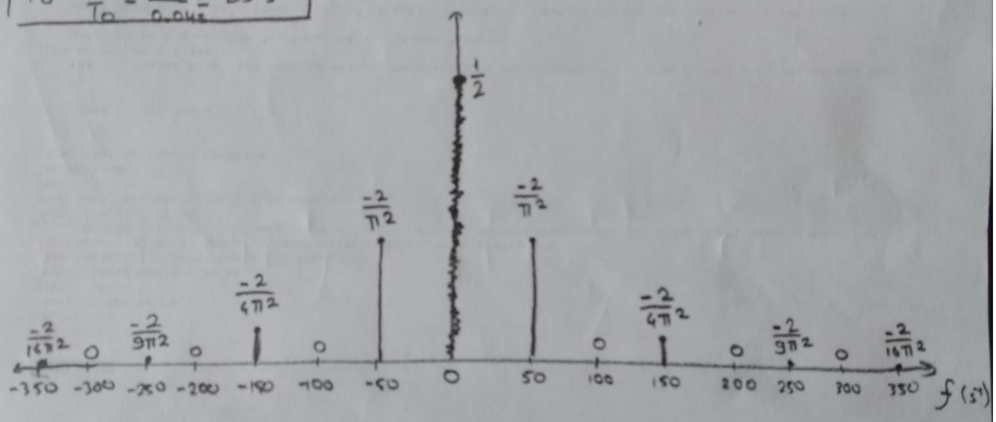
$$a_k = \frac{\cos(\pi k) - 1}{\pi^2 k^2}$$

$$\boxed{a_0 = \frac{2}{T_0^2}\left[-\int_{T_0/2}^{0} t\,dt + \int_{0}^{-T_0/2} t\,dt\right]}$$

$$\boxed{a_0 = \frac{2}{T_0^2}\left[+\frac{T_0^2}{8} + \frac{T_0^2}{8}\right] = \frac{1}{2}}$$

$$a_k = \begin{cases} \dfrac{-2}{\pi^2 k^2} & \text{when } k = \pm 1, \pm 3, \pm 5 \ldots \ (\cos(\pi k) = -1) \\ \dfrac{1}{2} & \text{when } k = 0 \\ 0 & \text{when } k = \pm 2, \pm 4, \pm 6 \ldots \ (\cos(\pi k) = 0) \end{cases}$$

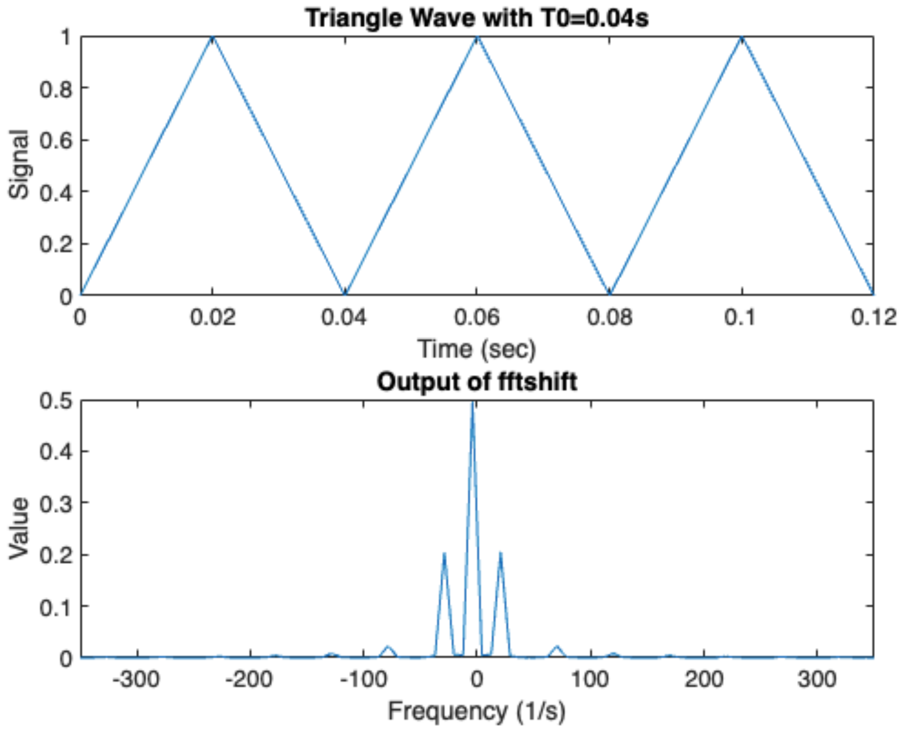
$$\boxed{F_0 = \frac{1}{T_0} = \frac{1}{0.04s} = 25\ s^{-1}}$$



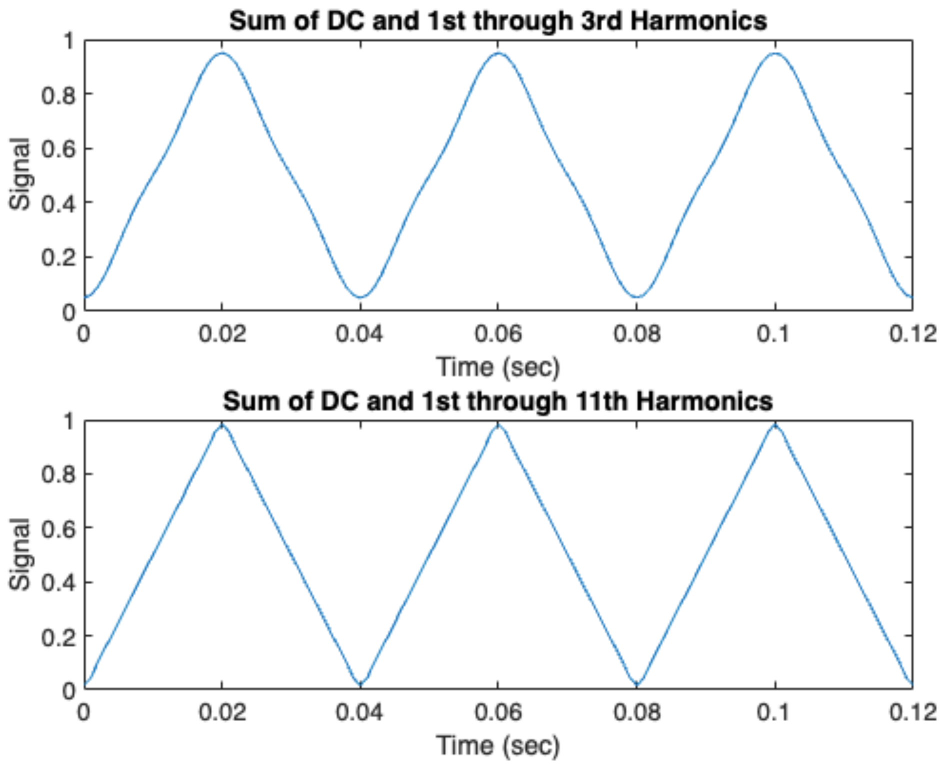
Alperen Bağ
2018400279

03.05.2020

1.b.-



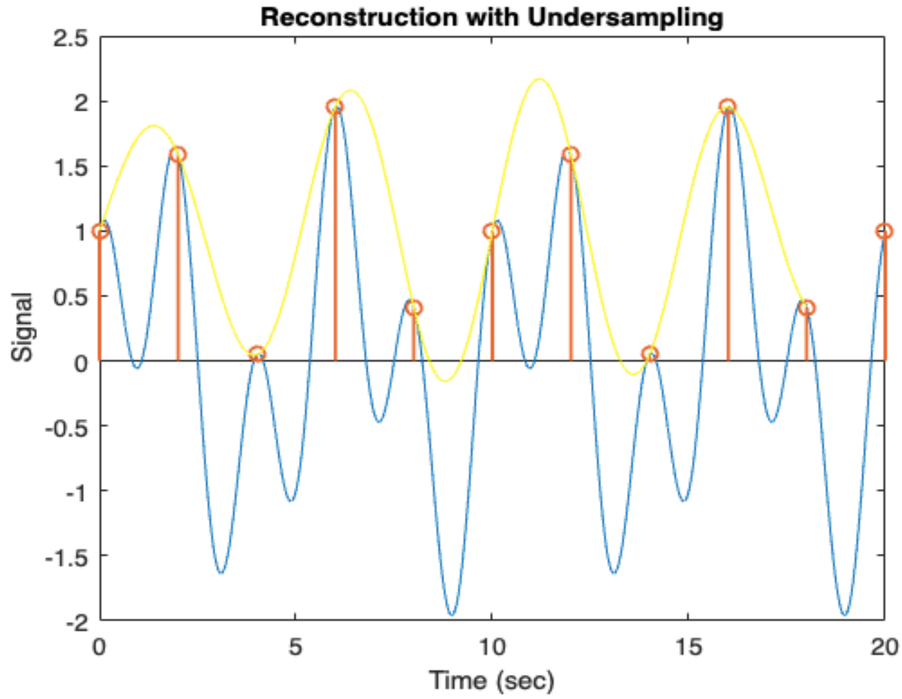
My calculations to find Fourier coefficients of given triangle waveform are consistent with the results obtained using discrete fast Fourier function of MATLAB. Reconstruction of the signal with the coefficients I found also proves that they are correct. Reconstructed signal is below.



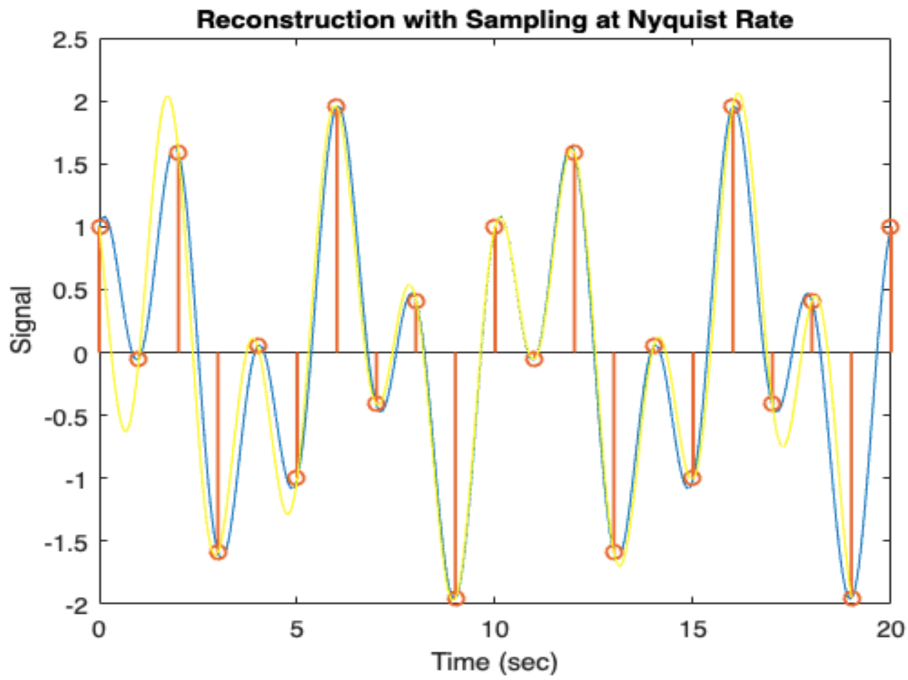
Alperen Bağ
2018400279 03.05.2020

2-

$$f_1 = 0.20\ s^{-1}$$
$$x_1 = \sin(2 * \pi * f_1 * t)$$
$$f_2 = 0.50\ s^{-1}$$
$$x_2 = \cos(2 * \pi * f_2 * t)$$

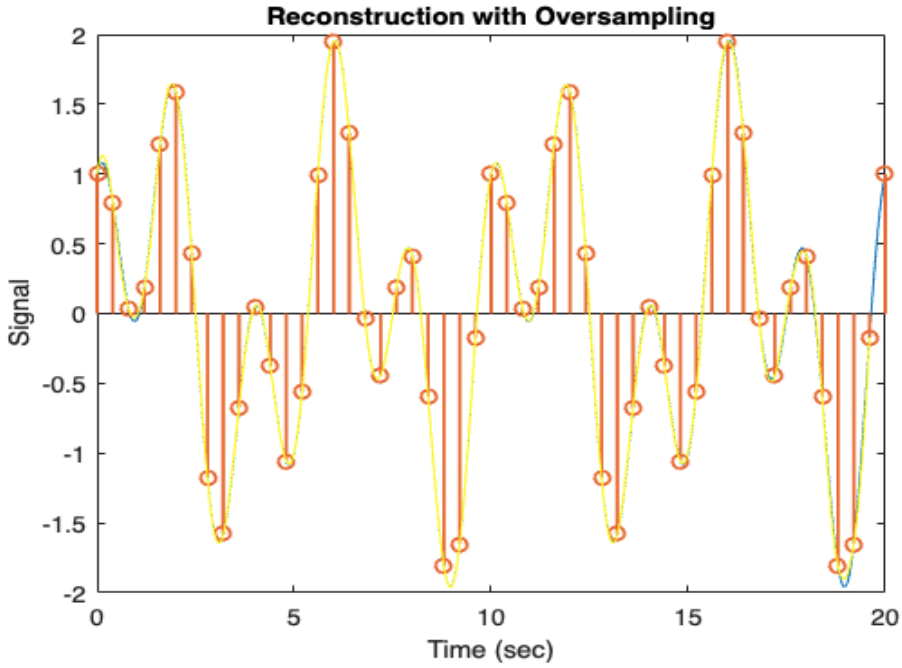


$$Sampling\ Rate = 0.50\ s^{-1}\ (under)$$



$$Sampling\ Rate = 1\ s^{-1}\ (under)$$

Alperen Bağ
2018400279 03.05.2020



$$\boldsymbol{Sampling\ Rate\ =\ 2.5\ s^{-1}\ (under)}$$

3-

I decided to use a filter with windows size 5. Initially, I implemented 5-pt moving averaging filter to see if it works well or not. Unfortunately, filter output was not good enough. It was almost the same as noisy input signal.

After the first try, I decided to train a 1-dimension filter using PyTorch. I wrote a simple train code and used the given test data as my train data. (Train script is included in my submission.) After 20 epoch training with batch size 1, I obtained the trained filter coefficients and put them in my MATLAB code. Result was not perfect again, however it was better than running averaging filter, when I listened output signals. However, according to FFT output below, we can see that my trained filter could not clean the low frequency noise completely.

Trained Filter Coefficients

$b_k = \{0.15, 0.24, 0.00, 0.35, 0.01\}$

Difference Equation

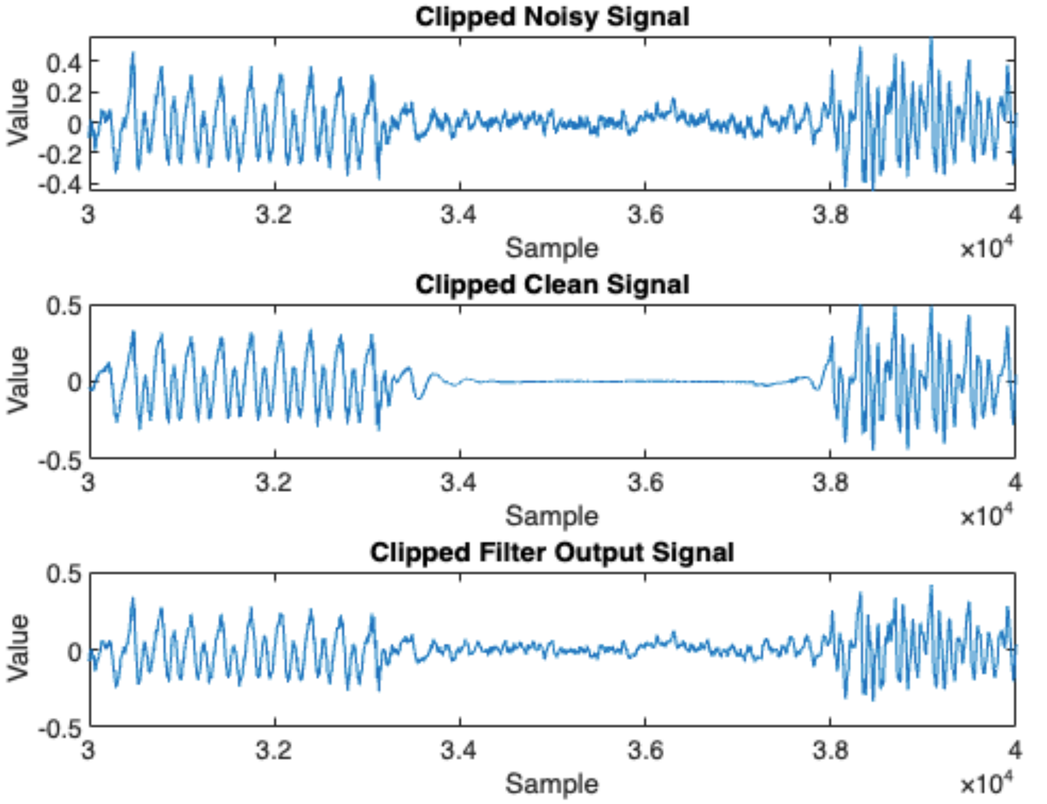
$$y[n] = 0.15 * x[n] + 0.24 * x[n-1] + 0.35 * x[n-3] + 0.01 * x[n-4]$$

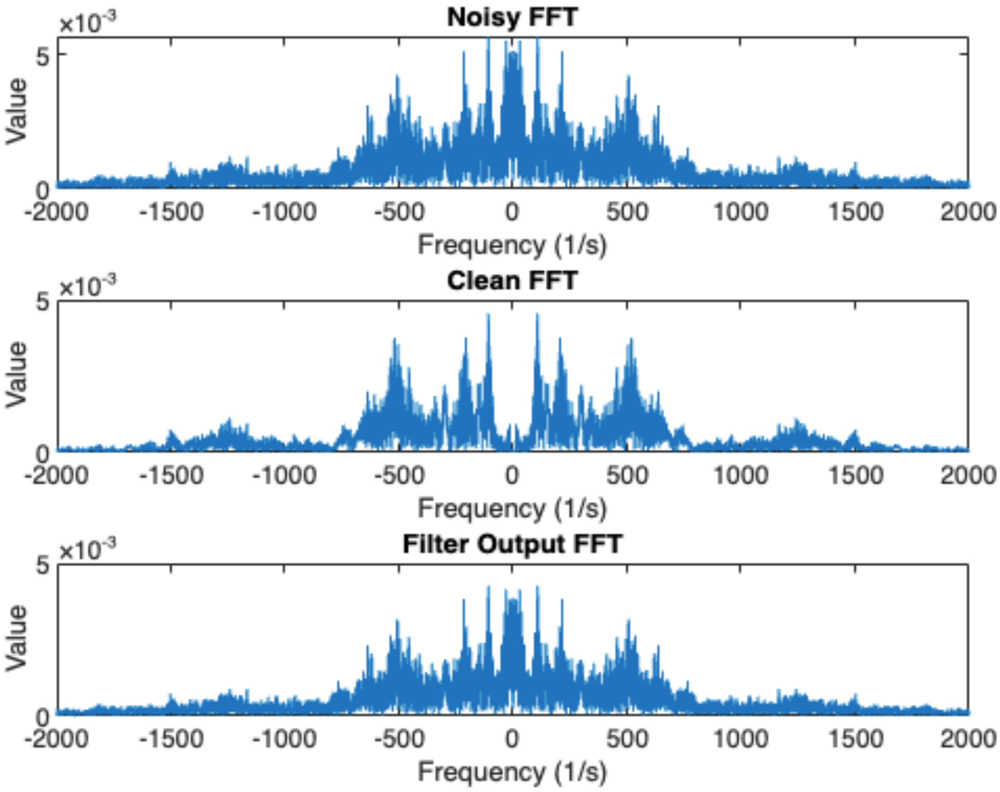
System Function

$$H(z) = \frac{Y(z)}{X(z)} = 0.15 + 0.24 * z^{-1} + 0.35 * z^{-3} + 0.01 * z^{-4}$$

$$H(z) = \frac{0.15 * z^4 + 0.24 * z^3 + 0.35 * z^1 + 0.01}{z^4}$$

Test with p232_010.wav





Alperen Bağ
2018400279

03.05.2020

Codes

1st Question

```
clear,clc,close all;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
figure;
subplot(2,1,1);
fs = 1000;
t = 0:1/fs:0.12;
x = (sawtooth(2*pi*25*t,0.5)+1)./2;
plot(t,x);
xlabel('Time (sec)');
ylabel('Signal');
title("Triangle Wave with T0=0.04s")
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
subplot(2,1,2);
y = fftshift(fft(x));
n = length(x);
f0 = (-n/2:n/2-1)*(fs/n);
normalized = abs(y/n);
plot(f0,normalized);
xlabel('Frequency (1/s)');
ylabel('Value');
title("Output of fftshift")
xlim([-350 350])
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
figure;
fs = 1000;
t = 0:1/fs:0.12;

subplot(2,1,1)
x = t.*0;
x = x + 0.5; % DC component
for k = 1:3
    if mod(k,2)==1
        x = x + (-4/((pi*k)^2))*cos(2*pi*25*k*t);
    end
end
plot(t,x);
xlabel('Time (sec)');
ylabel('Signal');
title("Sum of DC and 1st through 3rd Harmonics")
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
subplot(2,1,2)
x = t.*0;
x = x + 0.5; % DC component
for k = 1:11
    if mod(k,2)==1
        x = x + (-4/((pi*k)^2))*cos(2*pi*25*k*t);
    end
end
plot(t,x);
xlabel('Time (sec)');
ylabel('Signal');
title("Sum of DC and 1st through 11th Harmonics")
```

Alperen Bağ
2018400279
03.05.2020

2nd Question

```
clear,clc,close all;

f1 = 0.20;
f2 = 0.50;
t = 0:0.001:20;      % CT time
x1 = sin(2*pi*f1*t); % First CT signal
x2 = cos(2*pi*f2*t); % Second CT signal

% Undersampling
figure;
plot(t,x1+x2);
hold on;
fs_under = 1*f2;      % Sampling Rate
Ts_under = 1/fs_under;
td_under = 0:Ts_under:20;
x_under = sin(2*pi*f1*td_under) + cos(2*pi*f2*td_under);
stem(td_under, x_under,"Color",[0.91 0.41 0.17], "LineWidth",1.3);
% Reconstruction
y1 = zeros(1,length(t));
samples = length(td_under);
for i = 1:1:length(t)
    for n = 1:1:samples
        y1(i) = y1(i) + x_under(n)*sinc((t(i)-n*Ts_under)/Ts_under);
    end
end
hold on;
plot(t-Ts_under,y1,"Color","y");
xlim([0 20]);
xlabel('Time (sec)');
ylabel('Signal');
title("Reconstruction with Undersampling")
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Sampling at Nyquist Rate
figure;
plot(t,x1+x2);
hold on;
fs_nyquist = 2*f2;     % Sampling Rate
Ts_nyquist = 1/fs_nyquist;
td_nyquist = 0:Ts_nyquist:20;
x_nyquist = sin(2*pi*f1*td_nyquist) + cos(2*pi*f2*td_nyquist);
stem(td_nyquist, x_nyquist,"Color",[0.91 0.41 0.17],"LineWidth",1.3);
% Reconstruction
y2 = zeros(1,length(t));
samples = length(td_nyquist);
for i = 1:1:length(t)
    for n = 1:1:samples
        y2(i) = y2(i) + x_nyquist(n)*sinc((t(i)-n*Ts_nyquist)/Ts_nyquist);
    end
end
hold on;
plot(t-Ts_nyquist,y2,"Color","y");
xlim([0 20]);
xlabel('Time (sec)');
ylabel('Signal');
title("Reconstruction with Sampling at Nyquist Rate")
```

Alperen Bağ
2018400279 03.05.2020

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Oversampling
figure;
plot(t,x1+x2);
hold on;
fs_over = 5*f2;        % Sampling Rate
Ts_over = 1/fs_over;
td_over = 0:Ts_over:20;
x_over = sin(2*pi*f1*td_over) + cos(2*pi*f2*td_over);
stem(td_over, x_over,"Color",[0.91 0.41 0.17], "LineWidth",1.3);
% Reconstruction
y3 = zeros(1,length(t));
samples = length(td_over);
for i = 1:1:length(t)
    for n = 1:1:samples
        y3(i) = y3(i) + x_over(n)*sinc((t(i)-n*Ts_over)/Ts_over);
    end
end
hold on;
plot(t-Ts_over,y3,"Color","y");
xlim([0 20]);
xlabel('Time (sec)');
ylabel('Signal');
title("Reconstruction with Oversampling")
```

Alperen Bağ
2018400279

03.05.2020

3rd Question

```
clear,clc,close all;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
[x_noisy,fs_noisy] = audioread('noisy.wav');
[x_clean,fs_clean] = audioread('clean.wav');
l_noisy=length(x_noisy);
l_clean=length(x_clean);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% 5-pt Averaging Filter
%pt = 5;
%filter = ones([1 pt]).*(1/pt);

% Trained filter
filter = [0.15, 0.24, 0.00, 0.35, 0.01];

output = cconv(filter, x_noisy, length(x_noisy));
audiowrite("output.wav",output,fs_noisy);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
figure;
subplot(3,1,1);
X_noisy = fftshift(fft(x_noisy));
f_noisy = (-l_noisy/2:l_noisy/2-1)*(fs_noisy/l_noisy);
noisy_fourier = abs(X_noisy/l_noisy);
plot(f_noisy,noisy_fourier);
xlim([-2e3 2e3]);
xlabel('Frequency (1/s)');
ylabel('Value');
title("Noisy FFT")
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
subplot(3,1,2);
X_clean = fftshift(fft(x_clean));
f_clean = (-l_clean/2:l_clean/2-1)*(fs_clean/l_clean);
clean_fourier = abs(X_clean/l_clean);
plot(f_clean,clean_fourier);
xlim([-2e3 2e3]);
xlabel('Frequency (1/s)');
ylabel('Value');
title("Clean FFT")
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
subplot(3,1,3);
%plot(f_noisy, clean_fourier./noisy_fourier);
X_output = fftshift(fft(output));
f_output = (-l_noisy/2:l_noisy/2-1)*(fs_noisy/l_noisy);
output_fourier = abs(X_output/l_noisy);
plot(f_output,output_fourier);
xlim([-2e3 2e3]);
xlabel('Frequency (1/s)');
ylabel('Value');
title("Filter Output FFT")
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
figure;
subplot(3,1,1);
plot(x_noisy);
xlim([3e4 4e4]);
xlabel('Sample');
ylabel('Value');
title("Clipped Noisy Signal")
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
subplot(3,1,2);
plot(x_clean);
xlim([3e4 4e4]);
xlabel('Sample');
ylabel('Value');
title("Clipped Clean Signal")
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
subplot(3,1,3);
plot(output);
xlim([3e4 4e4]);
xlabel('Sample');
ylabel('Value');
title("Clipped Filter Output Signal")
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```