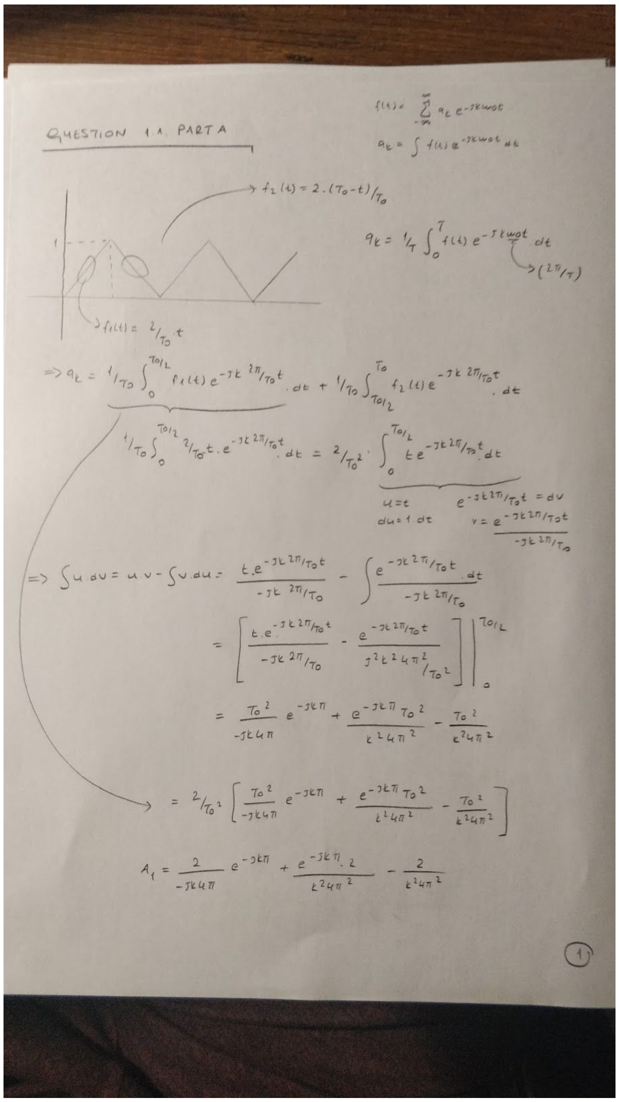
CMPE 362: Project 1 — Due:March 8st 23:59

Note: Prepare a report (pdf file) includes your code, explanations and comments of your code for each question. You will compress everything into a zip file. Name it as YourNumber-CmpE362-HW2.zip and submit it via Moodle.

Question 1.PART A:



QUESTION 1.A. PART A

$f(t) = \sum_{-\infty}^{\infty} a_k e^{-jkw_0 t}$

$a_k = \int f(t) e^{-jkw_0 t} dt$

$f_2(t) = 2.(T_0 - t)/T_0$

$a_k = 1/T \int_0^T f(t) e^{-jkw_0 t} dt$, $\; (2\pi/T)$

$f_1(t) = 2/T_0 \, t$

$\Rightarrow a_k = 1/T_0 \int_0^{T_0/2} f_1(t) e^{-jk 2\pi/T_0 t} . dt + 1/T_0 \int_{T_0/2}^{T_0} f_2(t) e^{-jk 2\pi/T_0 t} . dt$

$1/T_0 \int_0^{T_0/2} 2/T_0 t . e^{-jk 2\pi/T_0 t} . dt = 2/T_0^2 \int_0^{T_0/2} t e^{-jk 2\pi/T_0 t} dt$

$u = t \quad du = 1.dt \quad e^{-jk2\pi/T_0 t} = dv \quad v = \frac{e^{-jk2\pi/T_0 t}}{-jk 2\pi/T_0}$

$\Rightarrow \int u\, dv = u.v - \int v . du = \frac{t . e^{-jk 2\pi/T_0 t}}{-jk\, 2\pi/T_0} - \int \frac{e^{-jk 2\pi/T_0 t}}{-jk\, 2\pi/T_0} . dt$

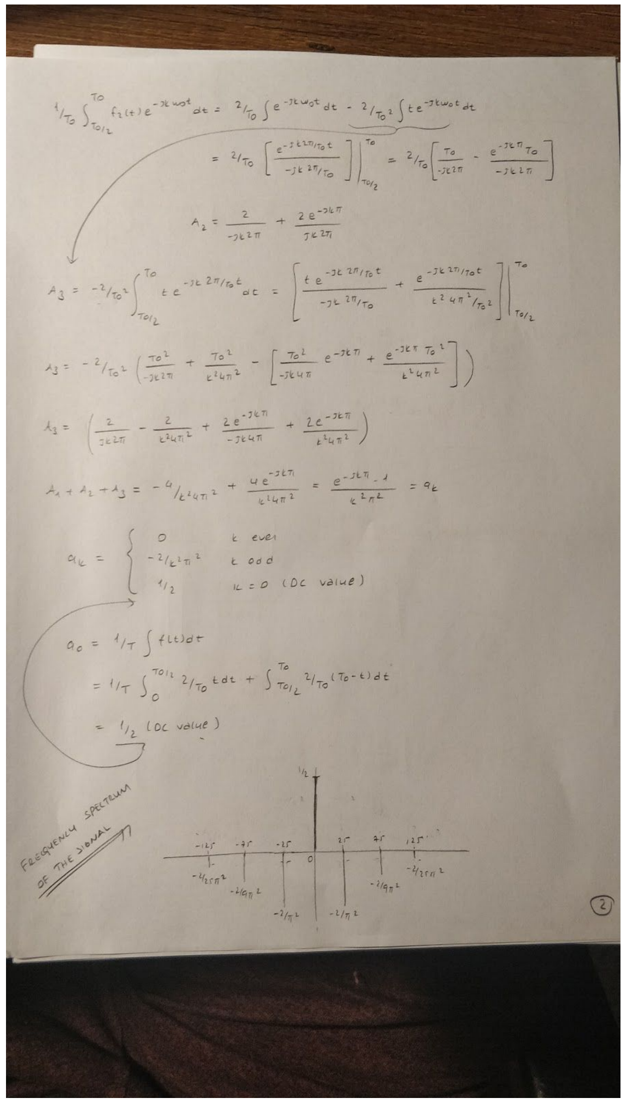
$= \left[\frac{t . e^{-jk 2\pi/T_0 t}}{-jk\, 2\pi/T_0} - \frac{e^{-jk 2\pi/T_0 t}}{j^2 k^2 4\pi^2 / T_0^2} \right] \Big|_0^{T_0/2}$

$= \frac{T_0^2}{-jk 4\pi} e^{-jk\pi} + \frac{e^{-jk\pi} T_0^2}{k^2 4\pi^2} - \frac{T_0^2}{k^2 4\pi^2}$

$= 2/T_0^2 \left[\frac{T_0^2}{-jk4\pi} e^{-jk\pi} + \frac{e^{-jk\pi} T_0^2}{k^2 4\pi^2} - \frac{T_0^2}{k^2 4\pi^2} \right]$

$A_1 = \frac{2}{-jk4\pi} e^{-jk\pi} + \frac{e^{-jk\pi} . 2}{k^2 4\pi^2} - \frac{2}{k^2 4\pi^2}$

1



$1/T_0 \int_{T_0/2}^{T_0} f_2(t) e^{-jkw_0 t} dt = 2/T_0 \int e^{-jkw_0 t} dt - 2/T_0^2 \int t e^{-jkw_0 t} dt$

$= 2/T_0 \left[\frac{e^{-jk 2\pi/T_0 t}}{-jk\, 2\pi/T_0} \right] \Big|_{T_0/2}^{T_0} = 2/T_0 \left[\frac{T_0}{-jk2\pi} - \frac{e^{-jk\pi} T_0}{-jk 2\pi} \right]$

$A_2 = \frac{2}{-jk2\pi} + \frac{2e^{-jk\pi}}{jk2\pi}$

$A_3 = -2/T_0^2 \int_{T_0/2}^{T_0} t e^{-jk 2\pi/T_0 t} dt = \left[\frac{t e^{-jk 2\pi/T_0 t}}{-jk\, 2\pi/T_0} + \frac{e^{-jk 2\pi/T_0 t}}{k^2 4\pi^2/T_0^2} \right] \Big|_{T_0/2}^{T_0}$

$A_3 = -2/T_0^2 \left(\frac{T_0^2}{-jk2\pi} + \frac{T_0^2}{k^2 4\pi^2} - \left[\frac{T_0^2}{-jk4\pi} e^{-jk\pi} + \frac{e^{-jk\pi} T_0^2}{k^2 4\pi^2} \right] \right)$

$A_3 = \left(\frac{2}{jk2\pi} - \frac{2}{k^2 4\pi^2} + \frac{2e^{-jk\pi}}{-jk4\pi} + \frac{2e^{-jk\pi}}{k^2 4\pi^2} \right)$

$A_1 + A_2 + A_3 = -4/k^2 4\pi^2 + \frac{4e^{-jk\pi}}{k^2 4\pi^2} = \frac{e^{-jk\pi} - 1}{k^2 \pi^2} = a_k$

$a_k = \begin{cases} 0 & k \text{ even} \\ -2/k^2\pi^2 & k \text{ odd} \\ 1/2 & k = 0 \text{ (DC value)} \end{cases}$

$a_0 = 1/T \int f(t) dt$

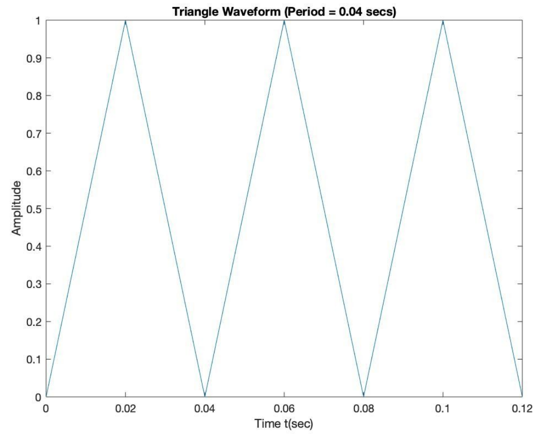
$= 1/T \int_0^{T_0/2} 2/T_0 \, t\, dt + \int_{T_0/2}^{T_0} 2/T_0 (T_0 - t) dt$

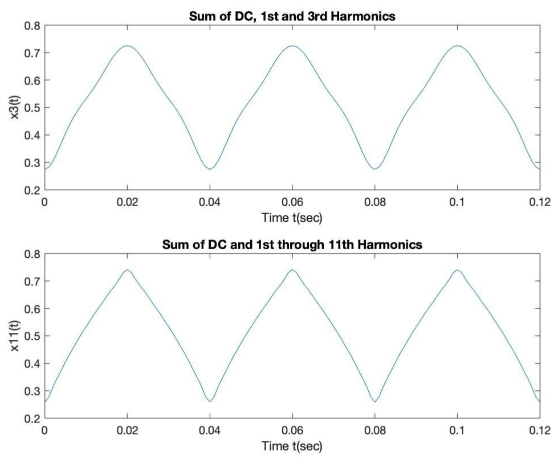
$= 1/2$ (DC value)

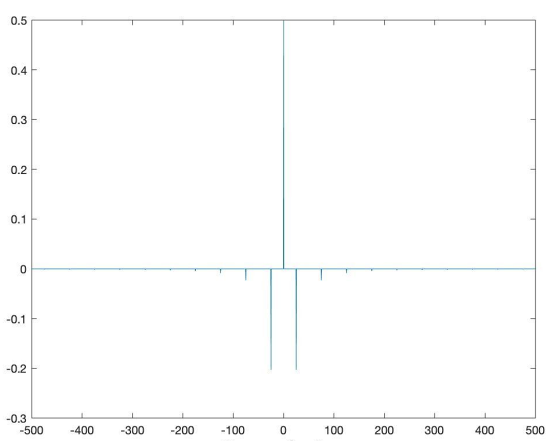
FREQUENCY SPECTRUM OF THE SIGNAL

2

Question 1.PART B:







Code:

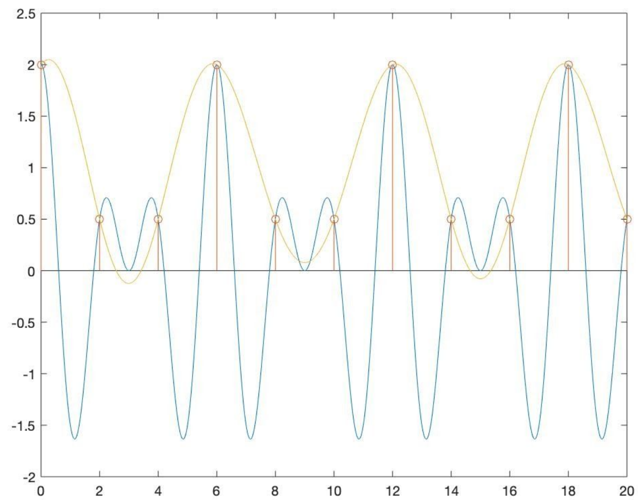
```
close all;

%% First part
figure('Name','Triangle Wave','NumberTitle','off');
T = 1/25;   % period
t = 0:0.001:3*T; % time variable with 0.001 step size
x = sawtooth(2*pi*1/T*t,0.5)/2 + 0.5; % signal

plot(t,x);
title('Triangle Waveform (Period = 0.04 secs)');
xlabel('Time t(sec)'); ylabel('Amplitude');

%% Second part

figure('Name','Harmonic Summation','NumberTitle','off');

% DC & 1st harmonic & 3rd Harmonic
wZero = 50 * pi;     % wZero value
harmonicOne = -(2 / (pi ^ 2)) * exp(wZero * t * 1j); % 1st harmonics
harmonicThree = -(2 / (3 ^ 2 * pi ^ 2)) * exp(3 * wZero * t * 1j);  % 3rd harmonics
x3 = harmonicOne + harmonicThree + 0.5; % Sum of DC & 1st harmonic & 3rd Harmonic

subplot(2,1,1);
plot(t, abs(x3));
title('Sum of DC, 1st and 3rd Harmonics');
xlabel('Time t(sec)'); ylabel('x3(t)');

% DC & 1st through 11th Harmonic
harmonicFive = -(2 / (5 ^ 2 * pi ^ 2)) * exp(5 * wZero * t *1j);     % 5th harmonics
harmonicSeven = -(2 / (7 ^ 2 * pi ^ 2)) * exp(7 * wZero * t *1j);    % 7th harmonics
harmonicNine = -(2 / (9 ^ 2 * pi ^ 2)) * exp(9 * wZero * t *1j);     % 9th harmonics
harmonicEleven= -(2 / (11 ^ 2 * pi ^ 2)) * exp(11 * wZero * t *1j); % 11th harmonics

x11 = harmonicOne + harmonicThree + harmonicFive + harmonicSeven + ...
   harmonicNine + harmonicEleven + 0.5;  % Sum of DC & 1st through 11th Harmonic
subplot(2,1,2);
plot(t,abs(x11));
title('Sum of DC and 1st through 11th Harmonics');
xlabel('Time t(sec)'); ylabel('x11(t)');

%% Third part

t = 0:0.001:T*50-0.001;  % time variable with 0.001 step size
x = sawtooth(2*pi*25*t,0.5)/2 + 0.5;  % signal
figure
y = fft(x);             % fft of x
n = length(x);          % number of samples

y0 = fftshift(y/n);         % shift y values
f0 = (-n/2:n/2-1)*(1000/n);
plot(f0,y0)
xlabel('Frequency Spectrum')
```

Question 2:
Code:

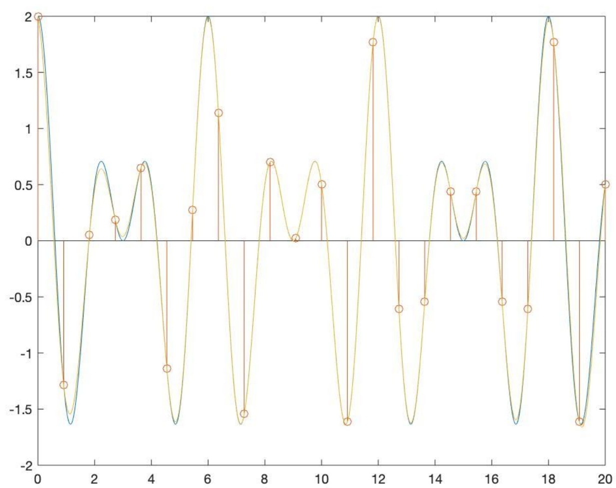
```
clear;
clc;
close all;

t = 0:0.001:20; % time variable with 0.001 step size

a = cos(2 * pi * 1 / 2 * t);    % first cos signal
b = cos(2 * pi * 1 / 3 * t);    % second cos signal

c = a + b;  % sum of these signals

%% UNDERSAMPLING

figure('Name','UNDERSAMPLING','NumberTitle','off');
plot(t, c); hold on;

Fs = 0.5;   % sampling rate

n = 0:1/Fs:20;

x = c(1:1/Fs*1000:20*1000+1);   % sampled points at fs 0.5

stem(n, x); hold on;

% sign interpolation
y = zeros(1,size(t,2));
for i = 1:size(x,2)

   y = y + x(i) * sinc((t - ((i - 1) * 1 /Fs)) * Fs);

end

plot(t, y);


%% NYQUIST SAMPLING

figure('Name','NYQUIST SAMPLING','NumberTitle','off');
plot(t, c); hold on;

Fs = 1.1;

n = 0:1/Fs:20;

x = c(1:1/Fs*1000:20*1000+1);   % sampled points at fs 1.1

stem(n, x); hold on;
```

```
% sign interpolation
y = zeros(1,size(t,2));
for i = 1:size(x,2)

    y = y + x(i) * sinc((t - ((i - 1) * 1 /Fs)) * Fs);

end

plot(t, y);

%% OVERSAMPLING

figure('Name','OVERSAMPLING','NumberTitle','off');
plot(t, c); hold on;

Fs = 5;

n = 0:1/Fs:20;

x = c(1:1/Fs*1000:20*1000+1);    % sampled points at fs 5.0

stem(n, x); hold on;

% sign interpolation
y = zeros(1,size(t,2));
for i = 1:size(x,2)

    y = y + x(i) * sinc((t - ((i - 1) * 1 /Fs)) * Fs);

end

plot(t, y);
```

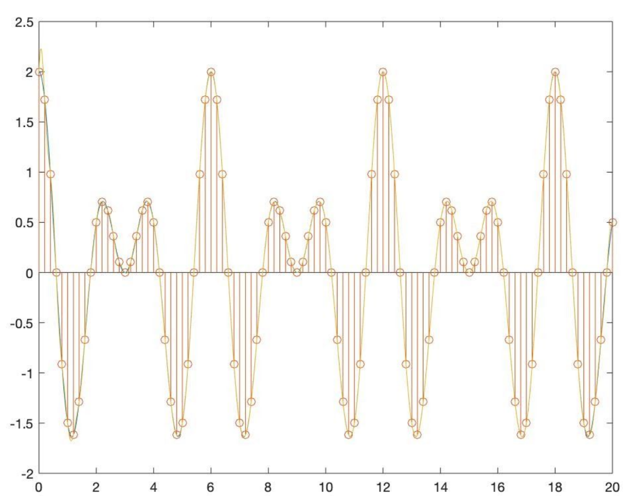
I take frequencies respectively ½ and ⅓ and also found that nyquist sampling rate is 1.1.



UNDERSAMPLING:



NYQUIST SAMPLING:



OVERSAMPLING:

Question 3:

This is ideal filter construct:
I select ”p232 090 clean.wav” and ”p232 090 noisy.wav”. Frequency spectrum for 2 signals can be seen in figure 5 : Then, I find the frequency response of my ideal filter by dividing FFT(clean signal)/FFT(noisy signal). The frequency response of my filter in figure 6: Now, let’s look how we could use this filter in time domain: I can find the time-response of my filter by using the functions ifftshift and ifft. There is a duality between time-domain and frequency domain. The multiplication operation in frequency domain is equivalent to the convolution operation in time domain. Therefore, I can obtain a new clean recording by taking convolution of noisy recording and time-response of your filter.

CODE:

```
%%  IDEAL FILTER SECTION
clear;
close all;
clc;
filename = "p232_090.wav";
[ySignal, Fs] = audioread(filename);

filename = "p232_090_noise.wav";
[yNoise, Fs] = audioread(filename);

T = 1 / Fs;


% fft transform of clean
fftY = fft(y);
fftY = fftshift(fftY);


% fft transform of noisy
fftYNoise = fft(yNoise);
fftYNoise = fftshift(fftYNoise);
% divide of clean / noise
fftDivide = fftY' ./ fftYNoise';
figure,plot(abs(fftDivide));
a = ifftshift(fftDivide);
a = ifft(a);
figure,plot(a);
c = cconv(a, yNoise');
figure, plot(y);
x=size(c);
c = c(1:ceil(x(2) / 2));
figure, plot(c);
```

My filter code:

```
clear;
close all;
clc;
filename = "p232_090.wav";
[ySignal, Fs] = audioread(filename);

filename = "p232_090_noise.wav";
[yNoise, Fs] = audioread(filename);
length = size(yNoise,1);
f0 = (-length/2:length/2-1)*(fs/length);

T = 1 / Fs;


%%

Y = fft(yNoise - ySignal);

L = 189269;

P2 = abs(Y/L);
P1 = P2(1:ceil(L/2));
P1(2:end-1) = 2*P1(2:end-1);

f = Fs*(0:(L/2))/L;

figure, plot(f,P1);


%%
Y = fft(ySignal);

L = 189269;

P2 = abs(Y/L);
P1 = P2(1:ceil(L/2));
P1(2:end-1) = 2*P1(2:end-1);

f = Fs*(0:(L/2))/L;

figure, plot(f,P1);

%%
zeroLele = zeros(length,1);
upper = 24000;
lower = 2000;
for i = 1:1:length
   if f0(i) > lower && f0(i) < upper
       zeroLele(i) = 1.0;
   end
   if f0(i) < -lower && f0(i) > -upper
       zeroLele(i) = 1.0;
   end
end
shift = ifftshift(zeroLele');
shift = ifft(shift);
c = cconv(abs(shift), y');
x = size(c);
c = c(1:ceil(x(2) / 2));
```