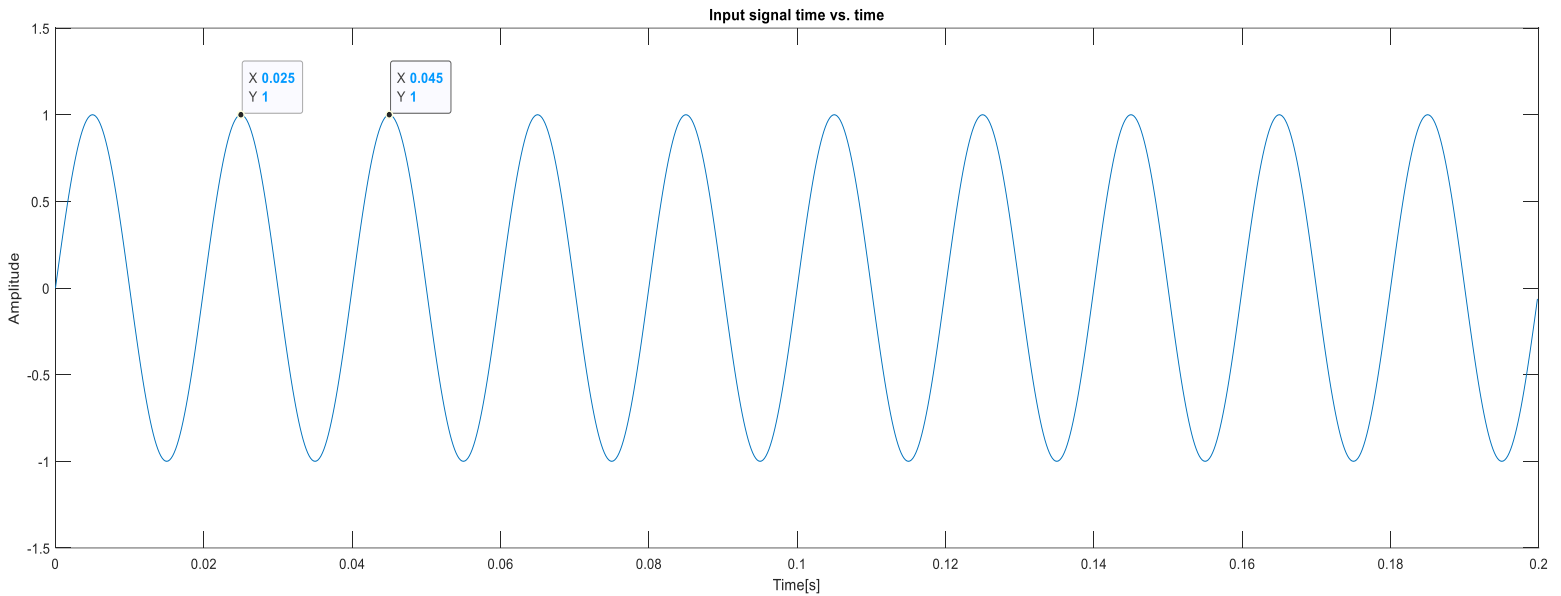


## DSP-Project

Rom Hirsch Id: 313288763, Yrom Swisa Id: 203675814

a. גרף של אות הכניסה בזמן:



גרף 1 - input signal vs. time

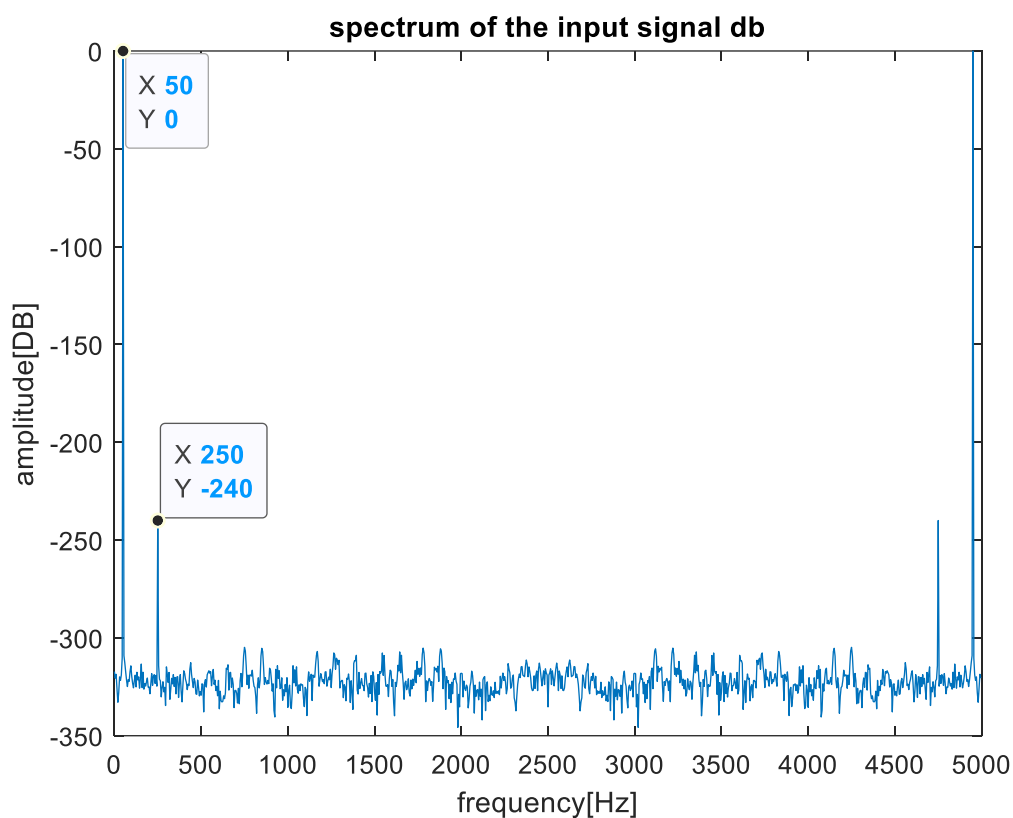
נמצא את התדר לפי הנקודות שסימנו כלומר:

$$f_{signal} = \frac{1}{T} = \frac{1}{0.45 - 0.025} = 50Hz$$

האמפליטודה לפי הגרף :

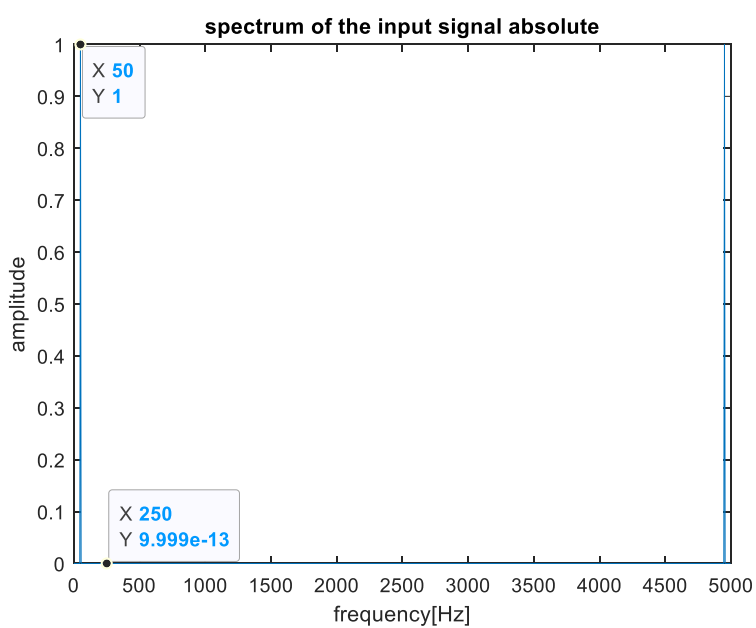
$$Amplitude = 1$$

b. ספקטרום אות הכניסה:



גרף 2 - spectrum of the input signal.

ספקטרום האות *absolute*:



גרף 3 - spectrum of the input signal absolute.

תדר האות:

$$f_{signal} = 250Hz$$
$$Amplitude = -240[db]$$

תדר הרעש – כפי שחישבנו בסעיף 1:

$$f_{noise} = 50Hz$$
$$Amplitude = 0[db]$$
$$Amplitude(absolute) = 1$$

c. תיכנון מסנן מסוג notch מסדר שני:  
 בתכנון נדרוש להוסיף אפסים בתדר הרעש כדי לאפסו, ולהוסיף קטבים במרחק של 0.9 לאותם  
 אפסים לפי הנתון.  
 נגדיר את האפסים וקטבים לפי הנוסחאות הבאות:

cut off -  $\Omega_c$   
 $a = 0.9$

$$\Omega_c = 2\pi * \frac{frequency_{noise}}{frequency_{samp}}$$

$$Zero1 = e^{i\Omega_c}$$

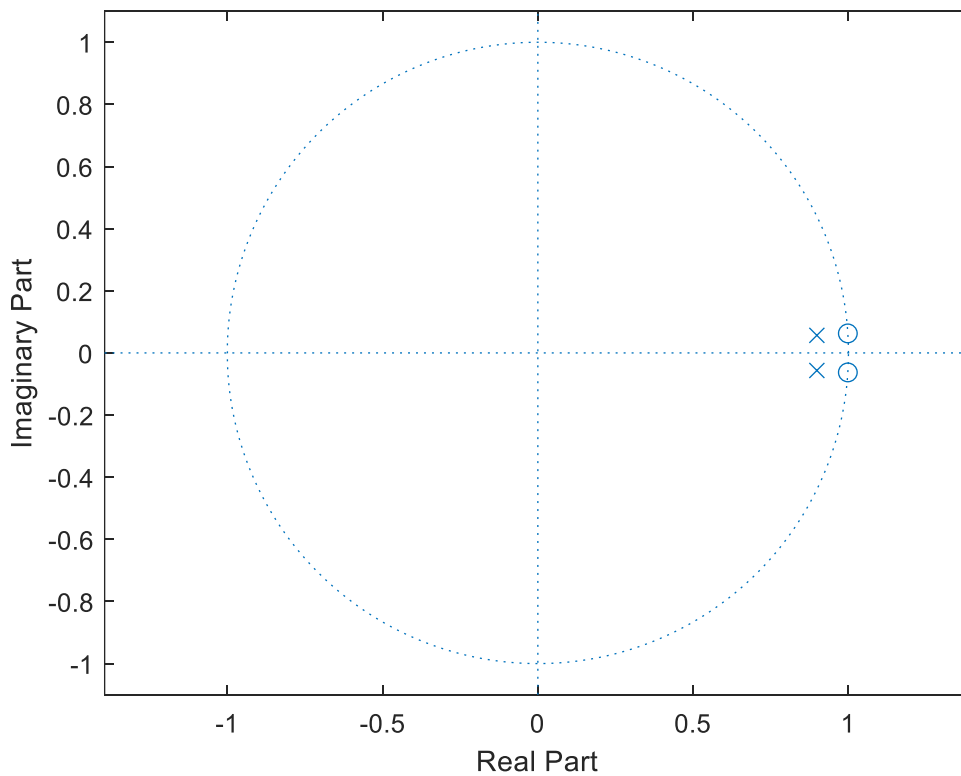
$$Zero2 = e^{-i\Omega_c}$$

$$Pole1 = a * e^{i\Omega_c}$$

$$Pole2 = a * e^{-i\Omega_c}$$

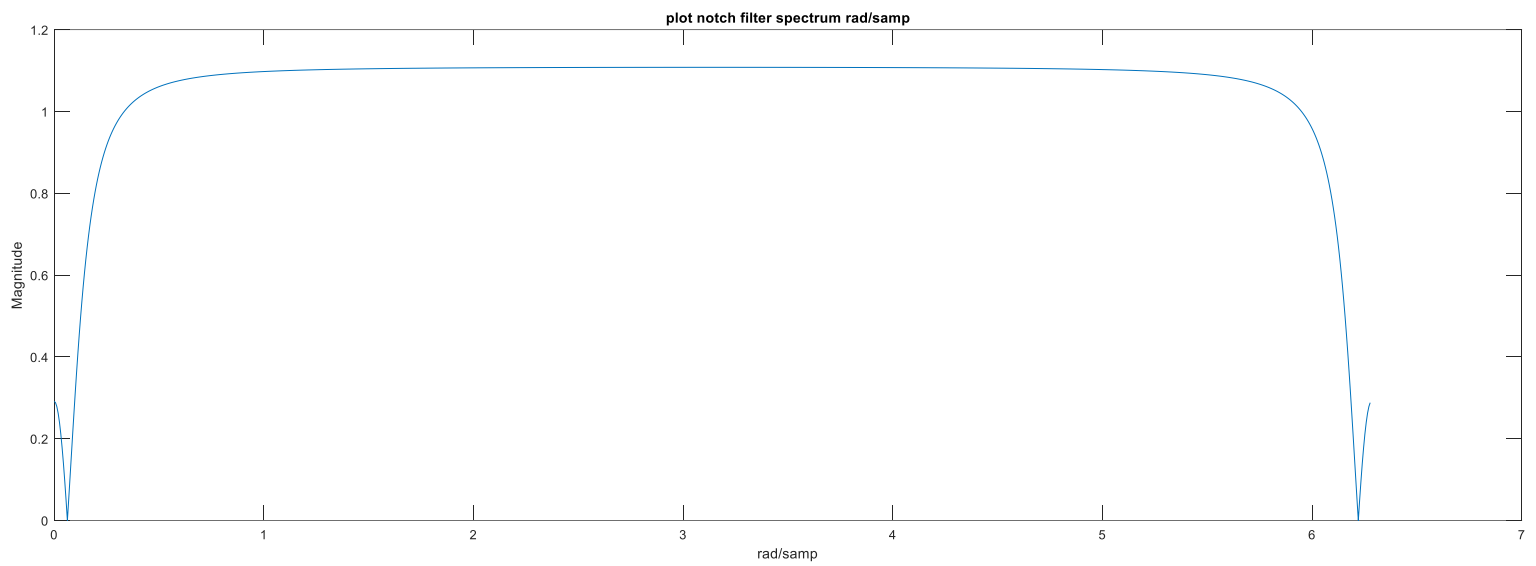
$$Transfer\ function(Notch) = \frac{(z - Zero1) * (z - Zero2)}{(z - Pole1) * (z - Pole2)}$$

: Pole zero diagram .d



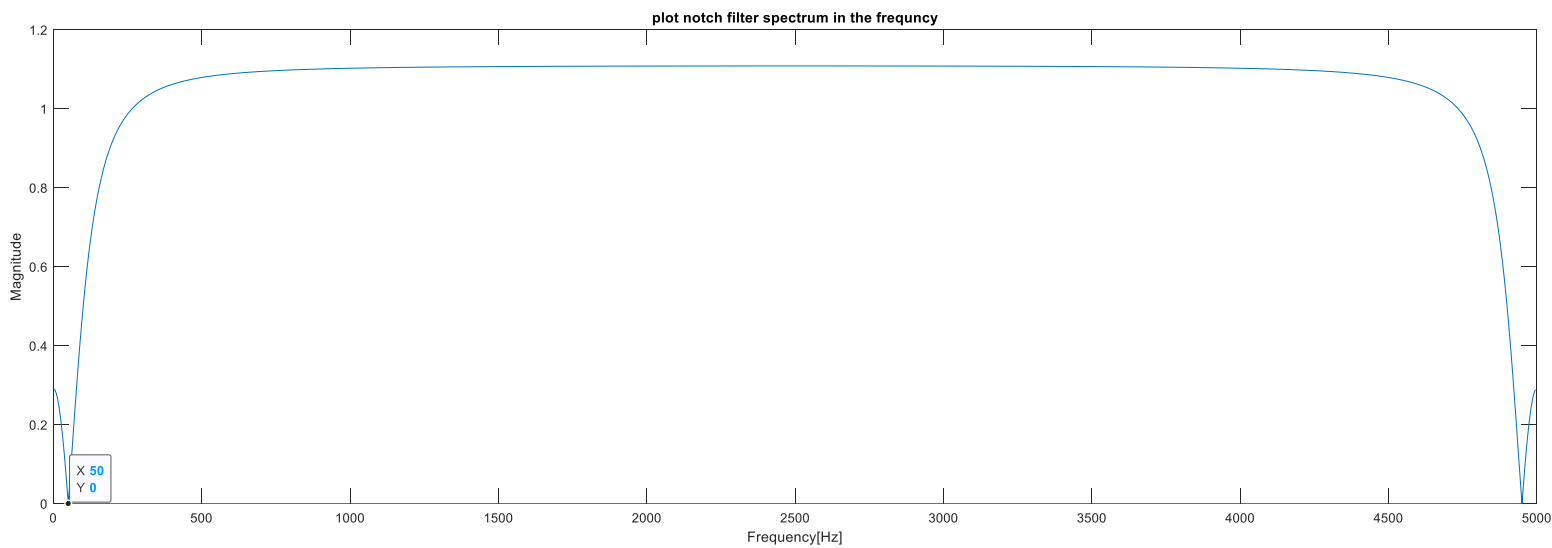
גרף 4 - pole zero diagram

e. הגרף של התגובה לתדר של הפילטר:



גרף 5 - *filter vs. normalized frequency form 0 to  $2\pi$  [rad/smp]*

גרף ספקטרום המסנן ביחס לתדר:



גרף 6 - *filter vs. frequency*

ניתן לראות שהמסנן לא יעביר את תדר הרעש כפי שתיכננו.

f. המשוואה הדפראנציאלית של המערכת:

$$\frac{Y}{X} = \frac{(z - \text{Zero1}) * (z - \text{Zero2})}{(z - \text{Pole1}) * (z - \text{Pole2})}$$

$$X * (z - \text{Zero1}) * (z - \text{Zero2}) = Y * (z - \text{Pole1}) * (z - \text{Pole2}) \rightarrow$$

$$\begin{aligned} X * (z^2 - z * (\text{Zero1} + \text{Zero2}) + (\text{Zero1} * \text{Zero2})) \\ = Y * (z^2 - z * (\text{Pole1} + \text{Pole2}) + (\text{Pole1} * \text{Pole2})) \rightarrow \end{aligned}$$

נמיר את האות לזמן:

$$\begin{aligned} x[n+2] - x[n+1] * (\text{Zero1} + \text{Zero2}) + x[n](\text{Zero1} * \text{Zero2}) \\ = y[n+2] - y[n+1] * (\text{Pole1} + \text{Pole2}) + y[n] * (\text{Pole1} * \text{Pole2}) \rightarrow \end{aligned}$$

מערכת סיבתית כמו זו אינה תלויה בערכי העתיד לכן נבצע הזזה בזמן כלומר נציב

$$n = n - 2$$

נקבל:

$$\begin{aligned} x[n] - x[n-1] * (\text{Zero1} + \text{Zero2}) + x[n-2](\text{Zero1} * \text{Zero2}) \\ = y[n] - y[n-1] * (\text{Pole1} + \text{Pole2}) + y[n-2] * (\text{Pole1} * \text{Pole2}) \rightarrow \end{aligned}$$

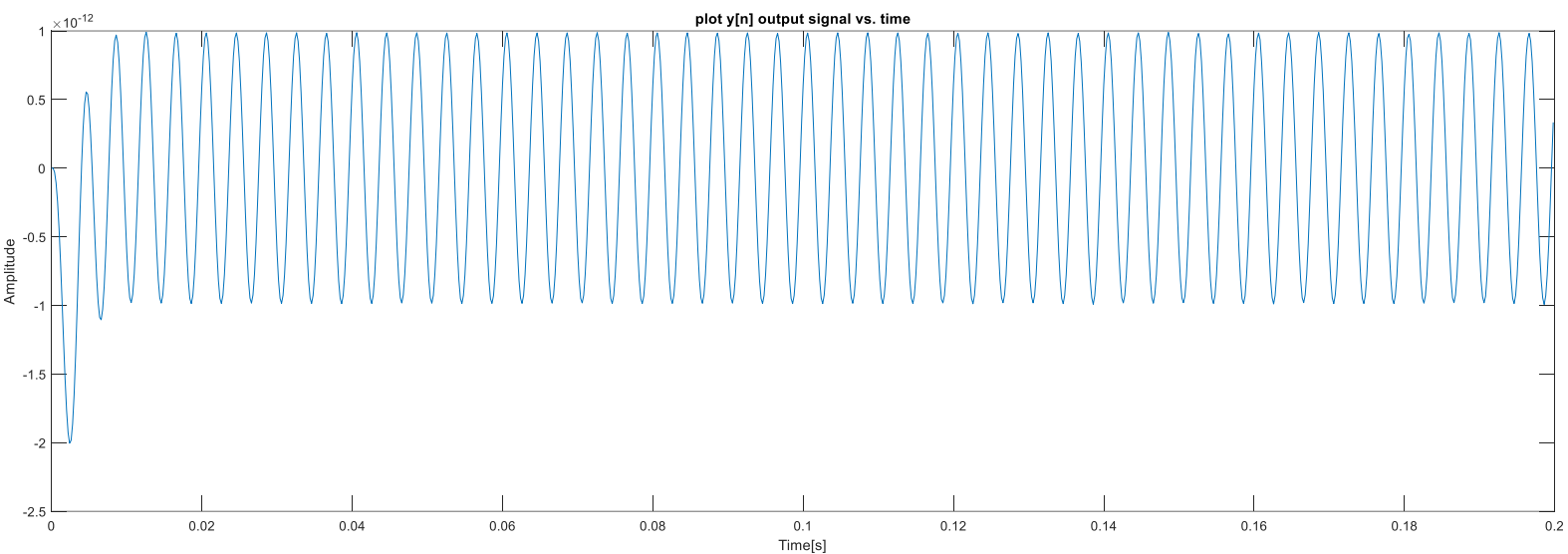
לכן המשוואה הדפראנציאלית של המערכת שווה:

$$\begin{aligned} y[n] = x[n] - x[n-1] * (\text{Zero1} + \text{Zero2}) + x[n-2](\text{Zero1} * \text{Zero2}) + y[n-1] \\ * (\text{Pole1} + \text{Pole2}) - y[n-2] * (\text{Pole1} * \text{Pole2}) \end{aligned}$$

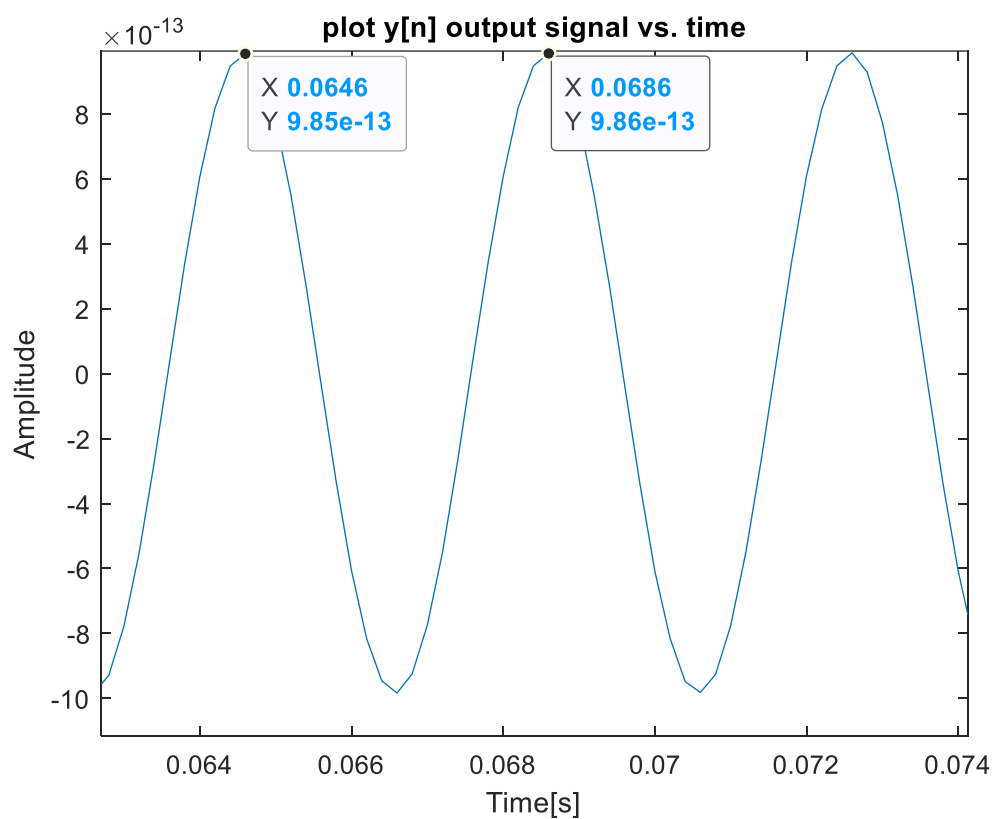
g. מימשנו את המערכת הדפראנציאלית בקוד matlab .

h. בוצע בקוד matlab cell של question i.

i. גרף .output signal vs. time



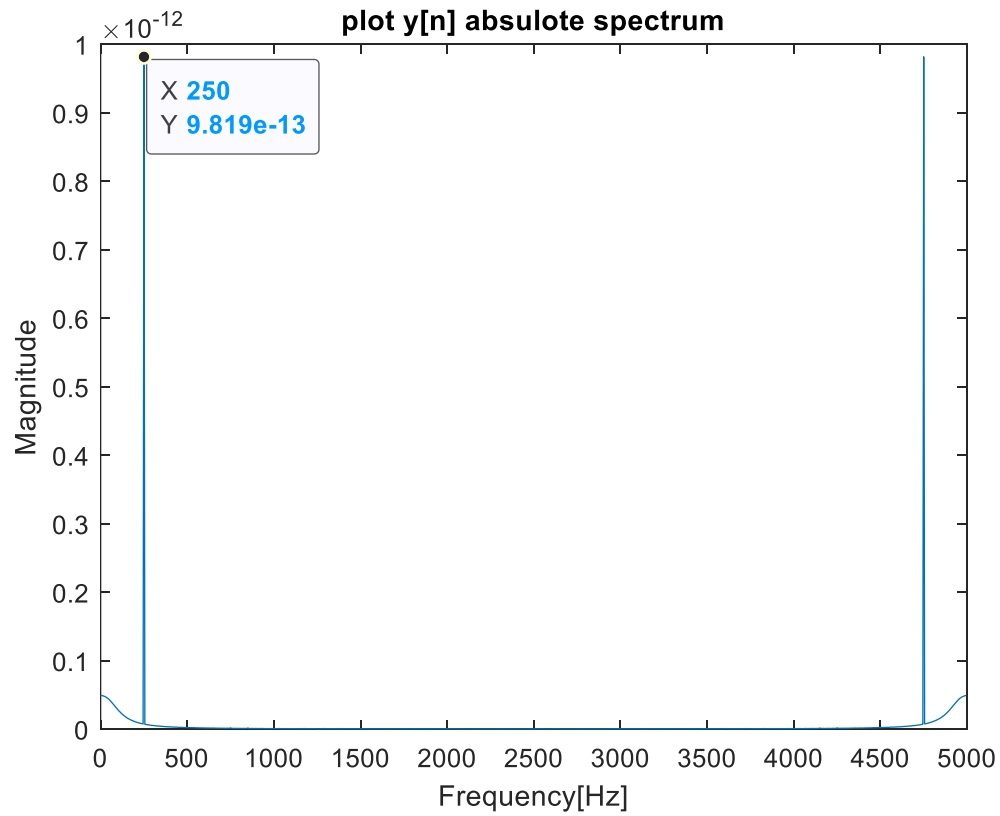
ן. תדר ואמפליטודה של האות



תדר האות לאחר הוספת הפילטר הוא :

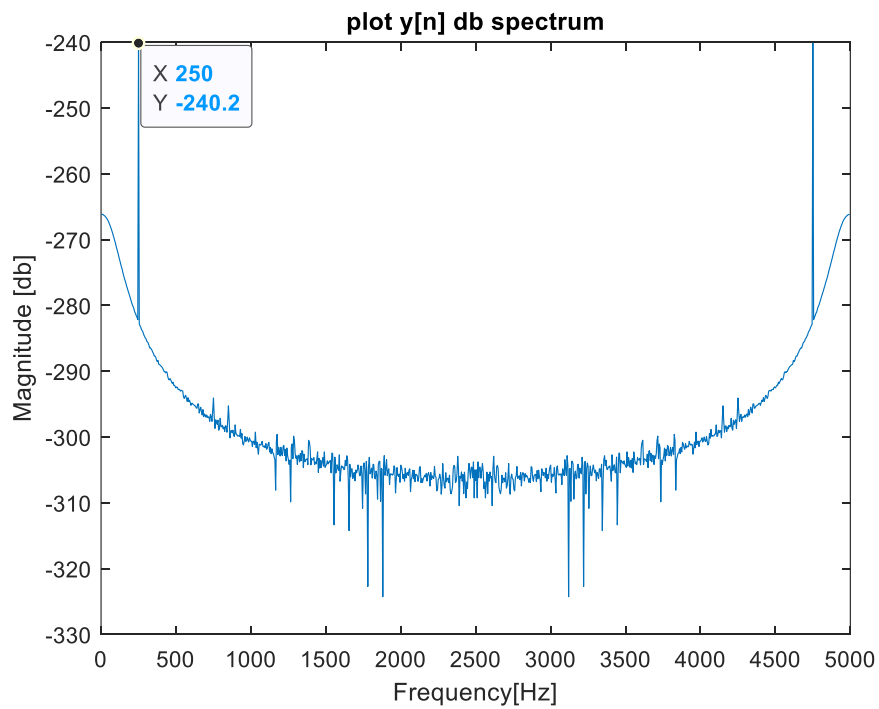
$$f_{signal} = 250Hz$$
$$Amplitude = 9.819e^{-13}$$

בנוסף ניתן לראות את התדר גם בעזרת הספקטרום להלן של גרף של הספקטרום:



ניתן לראות את התדר ב 250hz.

ספקטרום האות ב db :





לסיכום ניתן לראות שלאחר העברת האות דרך ה- notch filter כבר אין לנו את הרעש שהיה ב 50hz ועכשיו ניתן לראות את האות המתוכנן ב 250hz. בזמן ניתן לראות בתחילה תופעת מעבר של המסנן ולאחר מכן את האות הנדרש.

```
% Id: 313288763,203675814
clear all ;
clc;
close all;
clear figure;
% init vars
data = load("input_signal.mat").s;
N = length(data) ;% samples number
fs = 5e3 ;% sampled rate
Ts = 1/fs;
T0 = N*Ts ;% total sampling time
df = 1/T0; % resolution we can get in the spectrum figure()
t = (0:length(data)-1).*Ts; % time domain
%% Question a: plot Input signal time domain
figure()
plot(t,data);
title("Input signal time vs. time")
xlabel("Time[s]")
ylabel("Amplitude")

%% Question b: plot Input signal spectrum frequency Domain
spectrum = fft(data,N)/(N*0.5);
freq_axis = 0:length(spectrum)-1; % calc frequency array
freq_axis = freq_axis .*df; % calc frequency array
omega = 2.* pi .*(freq_axis/fs); % Omega = 2pi*(f/fs)
% plot Input signal spectrum db
figure()
plot(freq_axis,db(abs(spectrum)));
title("spectrum of the input signal db")
xlabel("frequency[Hz]")
ylabel("amplitude[DB]")
% plot Input signal spectrum absolute
figure()
plot(freq_axis,abs(spectrum));
title("spectrum of the input signal absolute")
xlabel("frequency[Hz]")
ylabel("amplitude")

%% Question d: notch filter design
noiseFreq = 50; % noise frequency we found
N = 1000; % samples number
a = 0.9; % notch magnitude
z = exp(1j.*omega); %z Transpose param z=e^-jomega
omegaCutOff = 2 * pi*(noiseFreq/fs); %calc frequency cut off
%Calc zeros and poles
zero1 = exp(1j*omegaCutOff);
zero2 = exp(-1j*omegaCutOff);
pole1 = a * exp(1j*omegaCutOff);
pole2 = a * exp(-1j*omegaCutOff);
notchFilter = ((z - zero1) .* (z-zero2) )./((z-pole1) .* (z-pole2)); %
notch filter equation

%% Question e: plot zero and pole map
zplane([zero1;zero2],[pole1;pole2]);% plot zero and pole map

%% Question f: plot notch filter spectrum
% plot notch filter spectrum between 0 - 2pi
figure()
```

```

plot(omega,abs(notchFilter))
title("plot notch filter spectrum rad/samp")
xlabel("rad/samp")
ylabel("Magnitude")
% plot notch filter spectrum in the frequency domain
figure()
plot(freq_axis,abs(notchFilter))
title("plot notch filter spectrum in the frequency")
xlabel("Frequency[Hz]")
ylabel("Magnitude")

%% Question h and i: y[n] difference calc and pass the input through the
filter.
y = zeros (1,N); %init y[n]
x=data;
%calc difference
for n = 3 : N
    y(n) = x(n)- x(n-1).*(zero1+zero2)+x(n-2).*(zero1.*zero2)+y(n-
1).*(pole1+pole2)-y(n-2).*(pole1.*pole2);
end
Y = fft(y)/(N*0.5);

%% Question j and k: plot y[n] output signal vs. time.
plot(t,y)
title("plot y[n] output signal vs. time")
xlabel("Time[s]")
ylabel("Amplitude")

% plot y[n] spectrum (Y[n])
figure()
plot(freq_axis,abs(Y))
title("plot y[n] absolute spectrum")
xlabel("Frequency[Hz]")
ylabel("Magnitude")

% plot y[n] spectrum db (Y[n])
figure()
plot(freq_axis,db(abs(Y)))
title("plot y[n] db spectrum")
xlabel("Frequency[Hz]")
ylabel("Magnitude [db]")

```