EECS2020 常安彥 107081028 Computer HW1 03/20/2020

Problem 1

(a) proof x_CT is a periodic signal with a frequency of $1/f_0$ mathematically and via MATLAB graphic illustration.

Mathematically:

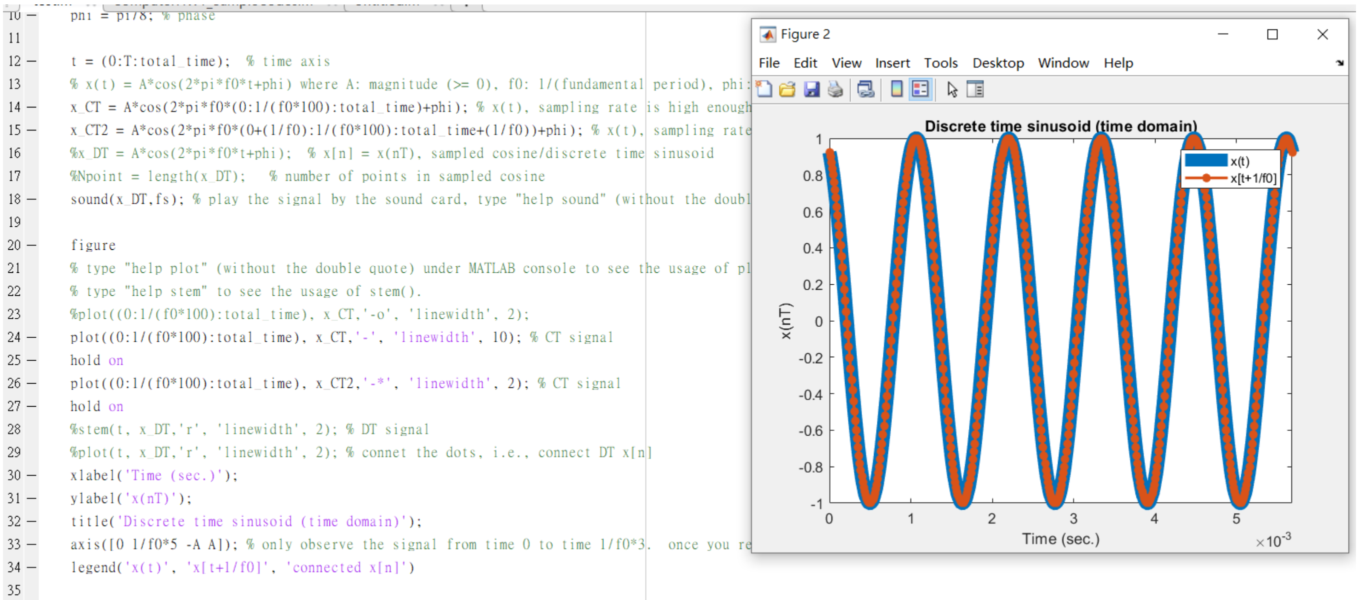
$$x(0) = Acos(2\pi f_0 \cdot 0 + \emptyset) = Acos(\emptyset)$$

$$x\left(\frac{1}{f_0}\right) = Acos\left(2\pi f_0 \cdot \left(\frac{1}{f_0}\right) + \emptyset\right) = Acos(\emptyset)$$

$$x\left(\frac{2}{f_0}\right) = Acos\left(2\pi f_0 \cdot \left(\frac{2}{f_0}\right) + \emptyset\right) = Acos(\emptyset)$$

$$x\left(\frac{n}{f_0}\right) = Acos\left(2\pi f_0 \cdot \left(\frac{n}{f_0}\right) + \emptyset\right) = Acos(2n\pi + \emptyset) = Acos(\emptyset) \quad n = 1, 2, 3 \ldots$$

MATLAB graphic illustration:



We drew another plot x_CT2 with $(1/f_0)$ slower at the beginning and the end. The result overlapped the original one. So it is proved that x_CT is a periodic signal with a frequency of $1/f_0$

x_CT2 = A*cos(2*pi*f0*(0 + (1/f0) : 1/(f0*100) : total_time + (1/f0))+phi);

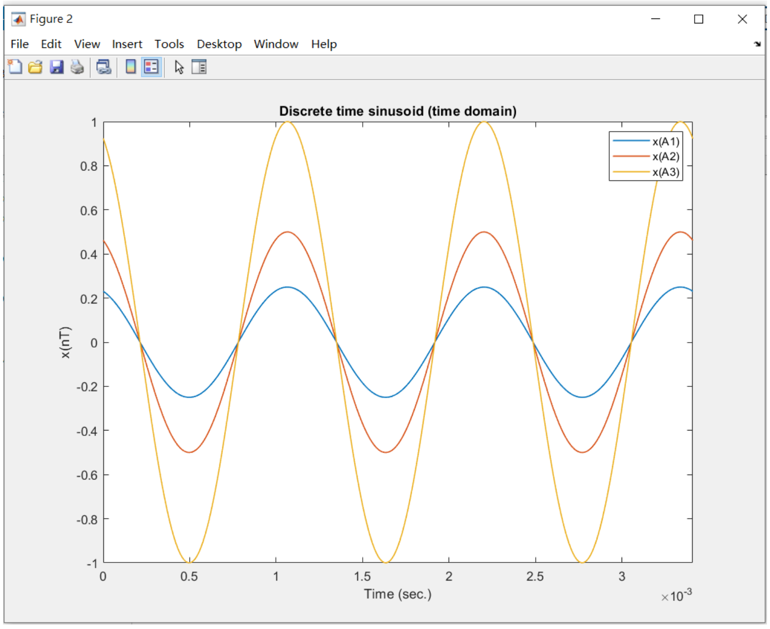
(b) Change A from 0.25 to 0.5, to 1, and tell the changes in the signal you observe and hear.

In the code, we changed the value of A from 0.25 to 0.5, to 1 through the line:

A = 0.25; % magnitude

The range of x(nT) increases as the amplitude increases. The amplitudes simply got larger, and the sound got louder although I couldn't distinguish the difference with my ear.

Plot:

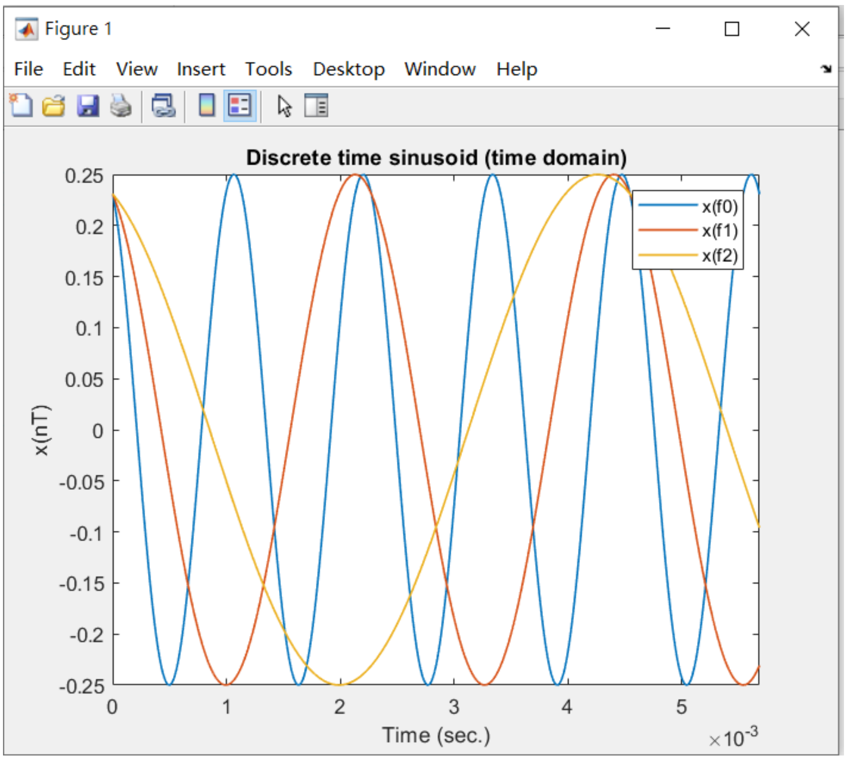


(c) Change f0 from 220 to 440, to 880, and tell the changes in the signal you observe and hear.

In the code, we changed the value of f0 from 220 to 440, to 880 through the line: f0 = 880; % frequency in Hz

We could observe that as f0 grows, wavelength (the interval of oscillation) gets shorter. As for hearing, the sound become more high pitched.

When f0 = 220, it sounded really low.

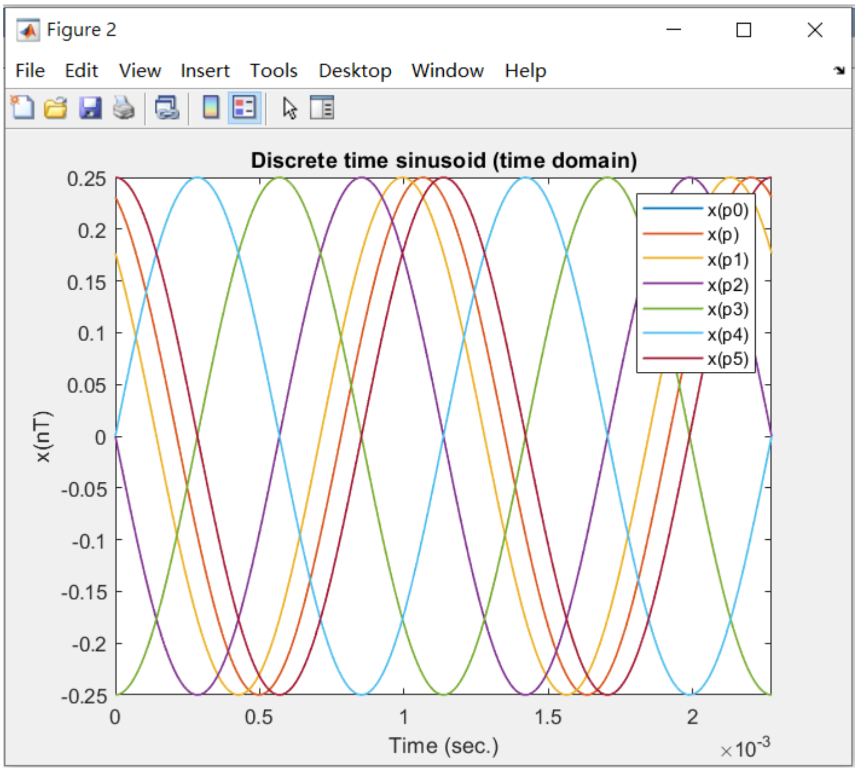


(d) Change phi from 0 to pi/8, to pi/4, to pi/2, to pi, to 3*pi/2, to 2*pi and tell the changes in the signal you observe and hear, what type of transformation of the independent variable? justify your answer. Is there any limitation for such a transformation to a sinusoidal signal by change of the phase phi? If your answer is yes, please describe the limitation.

In the code, we changed the value of phi from 0 to pi/8, to pi/4, to pi/2, to pi, to 3*pi/2, to 2*pi in the line:

phi = pi/2; % phase

Plot:



The phase difference makes the beginning point of the wave different.
Furthermore, If phi is greater than 2*pi, any plot of θ equals to (2*n*pi + θ) for n = 1, 2, 3… .
Therefore, the limitation is 2*pi.

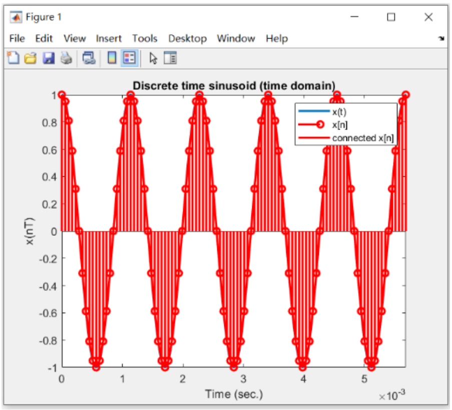
Problem 2

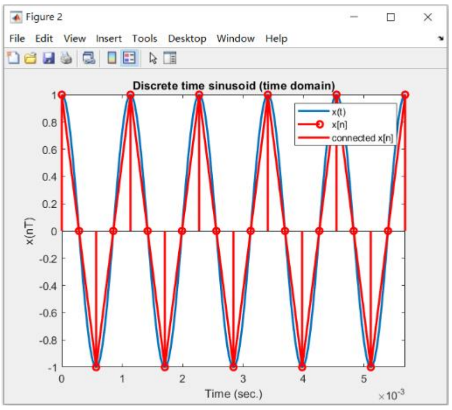
(a) change fsRatio from 20 down to 1.2 (at least try fsRatio = 20, 4, 2.5, 2.2, 2, 1.8, 1.4, and 1.2), and tell any differences among the sounds or any differnces among the DT signals

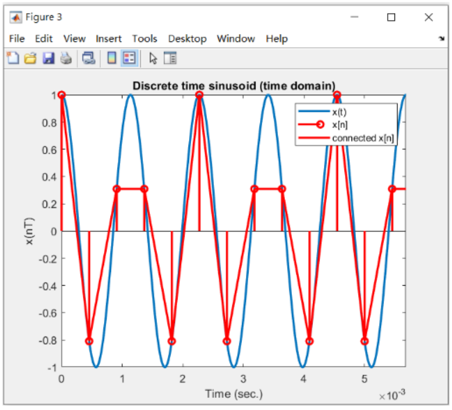
In the code, We changed the value of fsRatio from 20 down to 1.2 in the line:

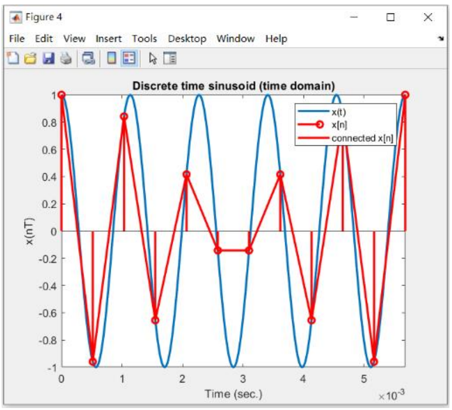
fsRatio = 20;

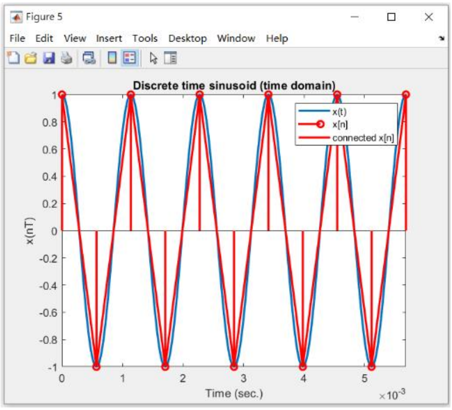
Following plots are in the queue: fsRatio = 20(figure1), 4(figure2), 2.5, 2.2, 2, 1.8, 1.4, 1.2

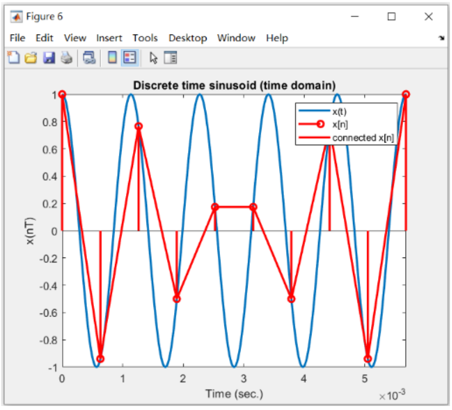


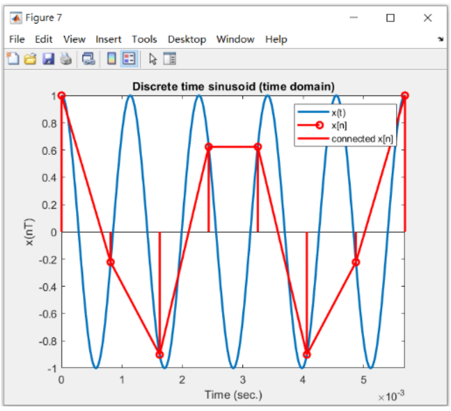


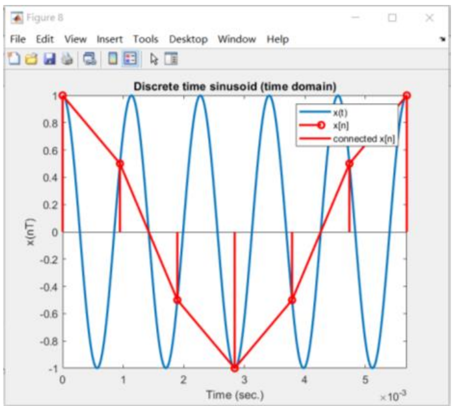












While fsRatio is still greater than 2, the sound sounded almost the same. Nevertheless, for those which are lower than 2, a significant drop on the frequency occurred. It sounded extremely low.

(b) following (a), please tell, what feature of the signal, magnitude, frequency, phase is changed after the sampling so that you hear the incorrect sound (i.e.,the sound is the not same as the sound of x(t))

Please tell once you lower down the fs, will the oscillation rate of the origial signal be kept? To kept the osillation rate, what should be the smallest fs

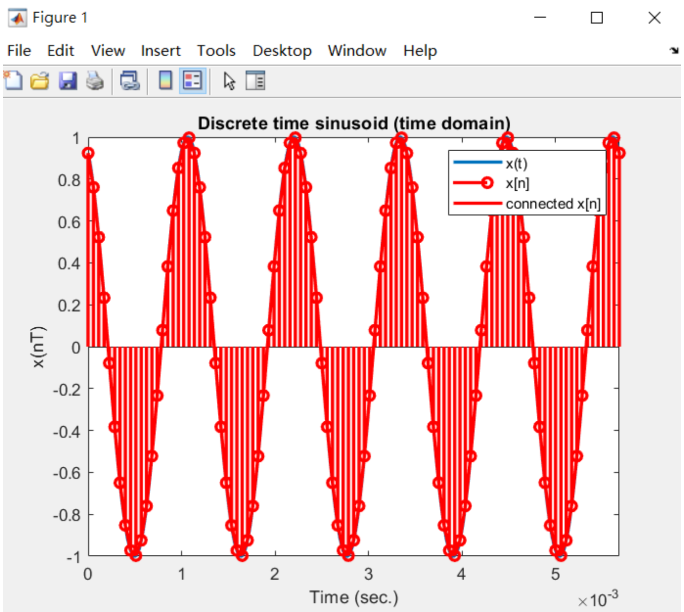
The rest features remained similar except for the frequency drop mentioned before. When the sampling rate was at least 2, we had at least sampled the highest and lowest points in every wave. For those between 2 and 4, if the average of points in every wave we sampled was greater than 2, the frequency would be preserved.

(c) Modify the code sound(x_DT,fs) at Line 29 to sound(x_DT,2*fs) and sound(x_DT,fs/2), respectively. Tell the changes and why.

The total time and the frequency changed. As the greater the fs timed to, the higher frequency and the shorter time occurred.

In the code, we changed the signal through the line: sound(x_DT,fs)

On the other hand, we didn't change the code of plotting, so the figure remained the same:



To explain the phenomenon, only changing the sampling rate affects the frequency of the signal and the velocity (total time). In other words, there are f1 signal points in the original form, after changing the sampling rate to 2*f1, the required time will reduce a half without changing the audio. Therefore, the velocity has changed. Furthermore, in order to sample the same quantity of samples, the frequency must be two times larger, otherwise the quantity of samples will not associate with the new sample rate due to the reduction of total time. Therefore, the frequency of the signal has changed.

Problem 3 (pure DT signal)

(a) please write down the math eq. of y, what mathematical operation is performed by the eq.? i.e., Generally, what do we call this mathematical operation, the name of this operation?

Calculating the dot product is performed by the eq. regarding function as vectors with infinite dimensions in Linear Algebra.

$$< f_1(t) \cdot f_2(t) > = \int_a^b f_1(t) * \overline{f_2(t)}\, dt$$

Conjunction is used because the answer is the produce of complexes.

(b) vary k and m from 1 to 50, respectively, and tell the change in y you observe and do you find any rule?

You can try the following codes which varies k and m via looping and shows you how to display y as a function of k and m.

In the code, we didn't change the details.

Code:

```
N = 100; % total time in normalized time
n = (0:1:N-1); % time axis, normalized time
k = 3; % integer, 1, 2, 3, 4, 5, 6, 7, 8, 9, ...
f1 = 1/N*k; % frequency
m = 5; % integer, from 1 to 2, 3, 4, 5, 6, 7, 8, 9, ...
f2 = 1/N*m; % frequency
x1 = exp(sqrt(-1)*(2*pi*f1*n));
x2 = exp(sqrt(-1)*(2*pi*f2*n));
y = sum(x1.*conj(x2));
K = 50;
M = 50;
y = zeros(K,M);
for k = 1:1:K
    for m = 1:1:M
        f1 = 1/N*k; % frequency
        f2 = 1/N*m; % frequency
        x1 = exp(sqrt(-1)*(2*pi*f1*n));
        x2 = exp(sqrt(-1)*(2*pi*f2*n));
        y(k,m) = sum(x1.*conj(x2));
    end
end
figure
```

```
imagesc(abs(y)) % display y as a function of k and m
colormap(gray)
colorbar
axis image
xlabel('m')
ylabel('k')
title('y')
```

As the result showed below, the value of the product would be zero (the black parts) unless k equaled to m. The set:

$e^{i2\pi Nt}$ $for\ N = 1, 2, 3, \ldots$ is an orthogonal basis.