EE386 Digital Signal Processing Lab **Jul-Dec 2021**

2: Sampling and Interpolation

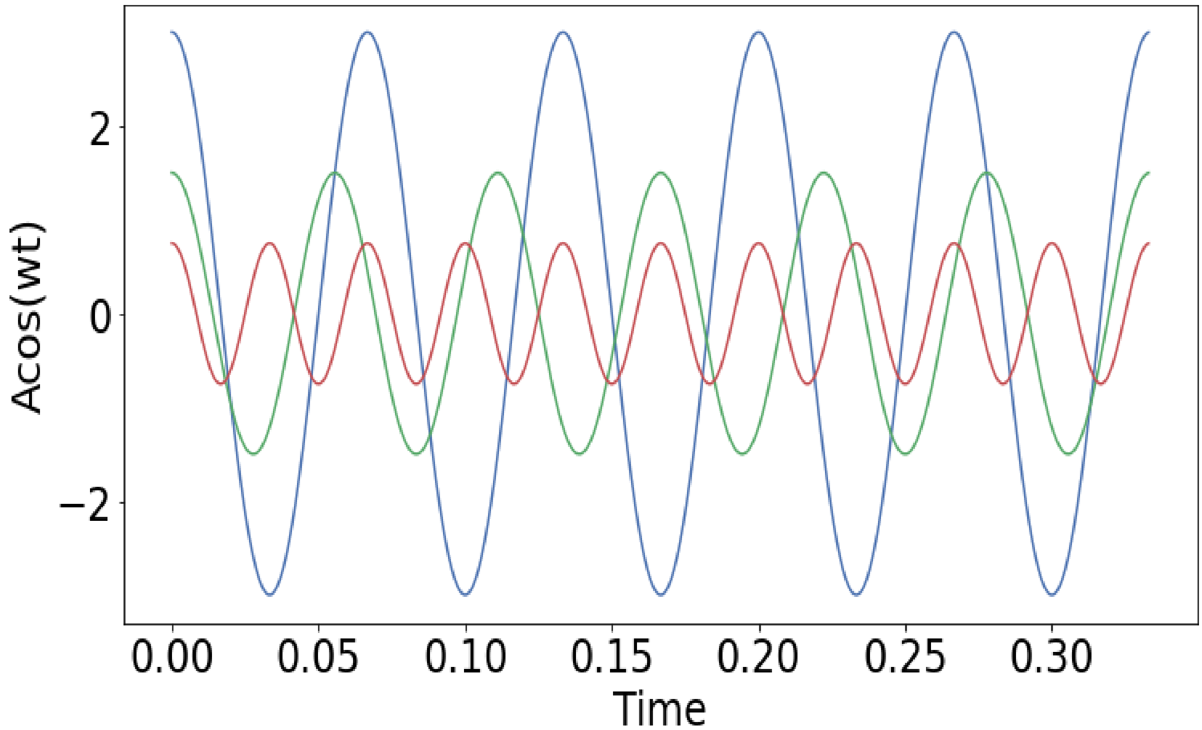
Author: Nikshep Manish Shah *Email:* nikshepshah.181ee231@gmail.com

1 Problem 1

(i) The given problem asks to plot a cosine function with given amplitude and frequency.
The below plot shows 3 different plots with different amplitudes and frequency.

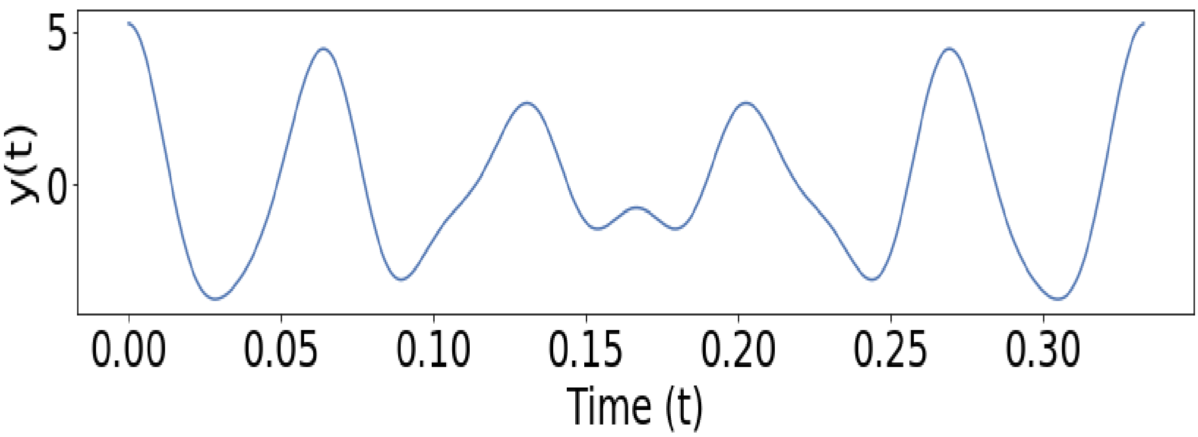
$$\text{Blue graph} : 3\cos(30\pi t)$$
$$\text{Green graph} : 1.5\cos(36\pi t)$$
$$\text{Blue graph} : 0.75\cos(60\pi t)$$

It is plotted using linspace for arranging samples at specific intervals.



(ii) The question asks to plot the sum of all three functions.
The below plot is of the following question :

$$\text{y(t)} = 3\cos(30\pi t) + 1.5\cos(36\pi t) + 0.75\cos(60\pi t)$$



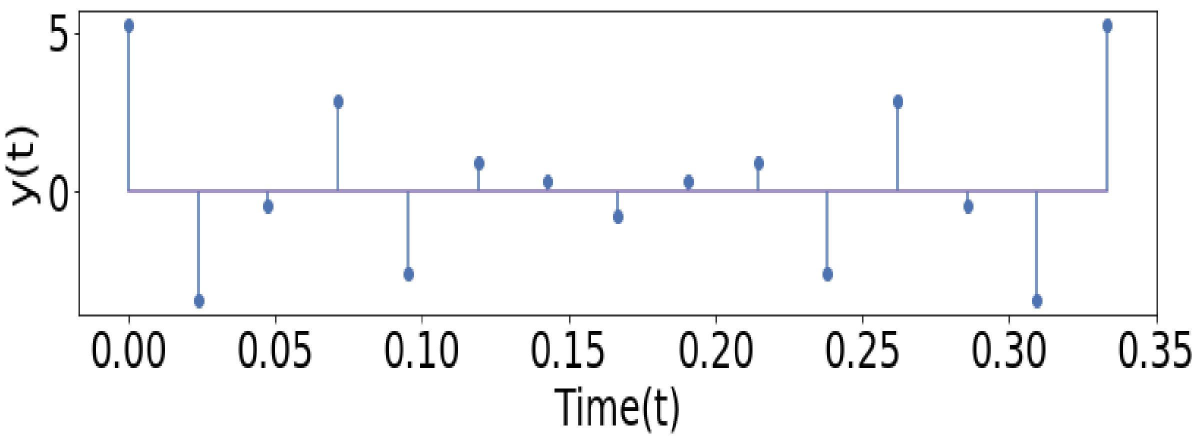
(iii) We need to stem plot the equation in part (ii) using different sampling frequencies as mentioned in the question.

So the first one says to plot at $14\alpha = 42$ samples/sec.

Therefore, Fs= 42 samples/s

Since we are plotting for 1/3 rd second there should be 14 samples for the duration. This is done by modifying the equations to

$$\mathrm{y(n)} = 3\cos(\frac{30\pi}{Fs}n) + 1.5\cos(\frac{36\pi}{Fs}n) + 0.75\cos(\frac{60\pi}{Fs}n)$$

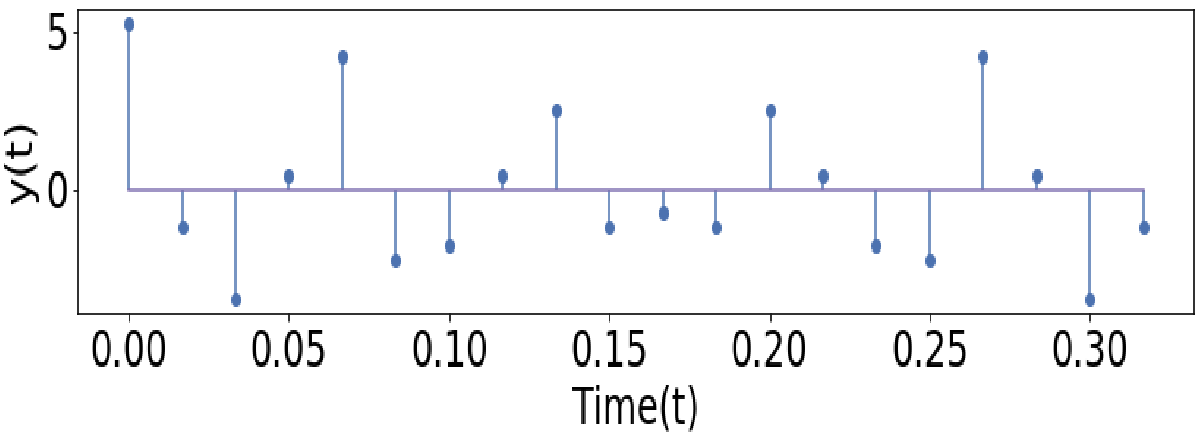


The question next asks to plot the graph for Nyquist Rate as sampling frequency.

$$\text{Nyquist Frequency} = 2*(\text{Maximum Frequency})$$

In the above summing equation maximum frequency is of the component $0.75\cos(60\pi t)$ where f=30Hz, therefore f_{max}=30Hz.

$$\mathrm{Fs} = 2*\mathrm{F_{max}}$$
$$\mathrm{Fs} = 60\mathrm{Hz}$$

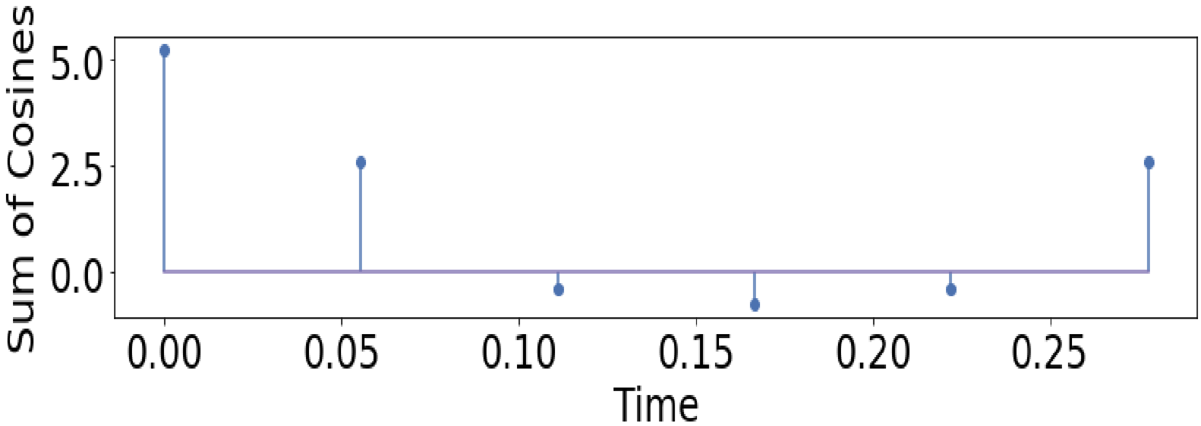


As we can observe from the above two figures, the second figure is more accurate and close to the actual figure. This is because sampling at frequency above Nyquist rate gives appropriate modelling of graph. The first plot is at a sampling rate lower than Nyquist rate therefore the plot is not accurate to the point.

Using the below two equations we find that 6α=18Hz can be aliased to 3α=9Hz when Fs= 18hz or 36Hz or 54Hz.

We'll be plotting for Fs = 18Hz

$$\frac{F}{Fs} = f$$
$$f = \frac{k}{N}$$



As we can observe from the above graph the signal when tried to regenerate from this is actually an aliased one and we miss a lot of the components and signal generated is a false one.

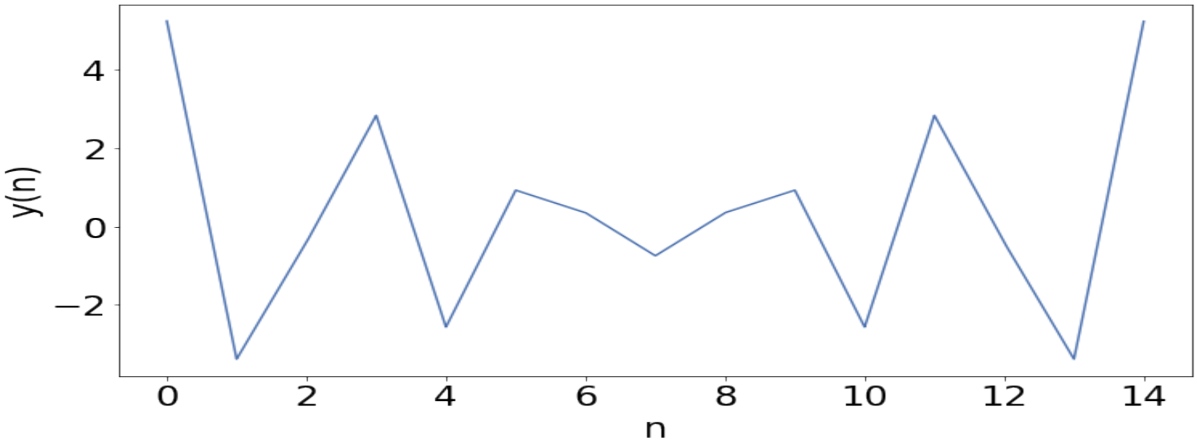
(iv)

Interpolation is the process of reconstructing a signal from the given samples.

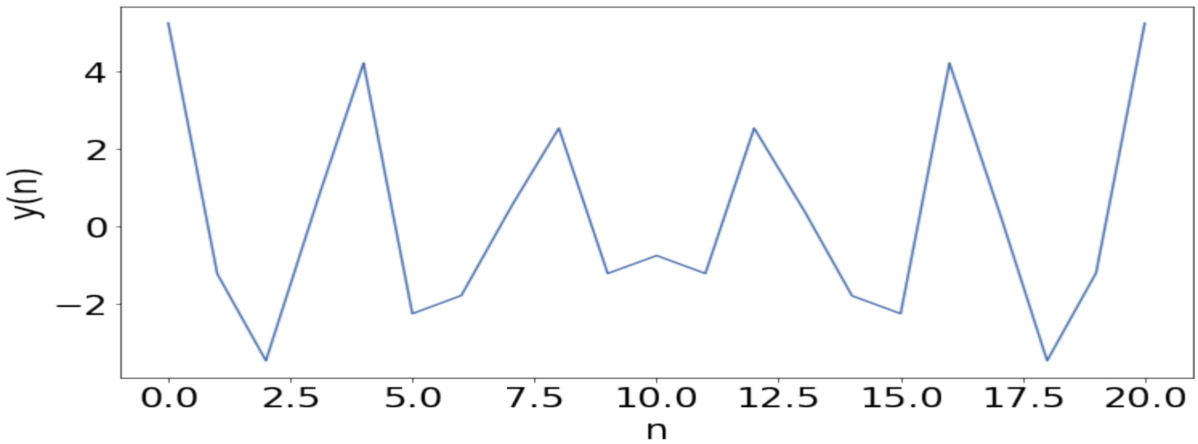
The question here asks to perform linear interpolation on the signals with different sampling frequencies.

Here I have performed interpolation using the command interpld in scipy.interpolate. Linear interpolation can manually be performed using equation of line between two successive samples/points in a signal. On this line we can introduce new points as per our requirement.

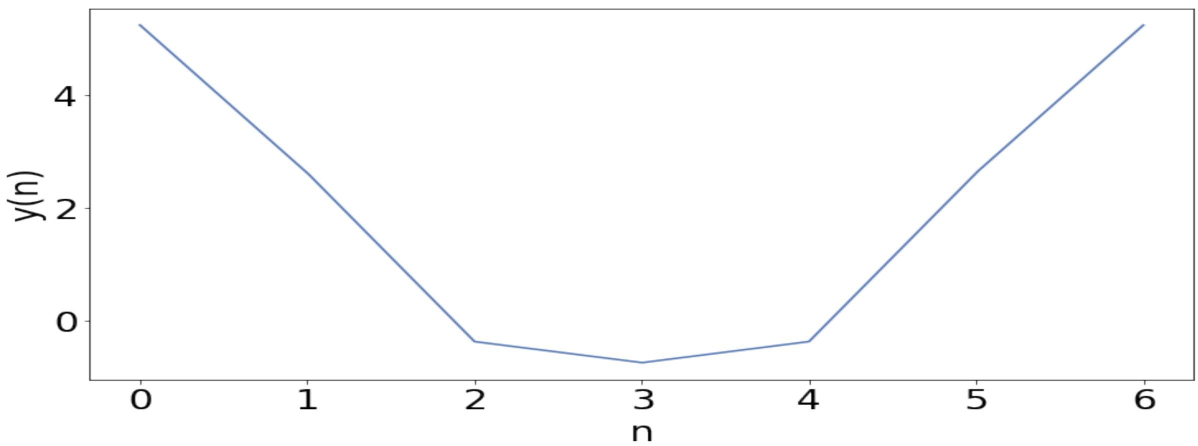
This is fundamentally called linear interpolation.



The above signal is interpolated for Fs= 42 samples/sec



The above signal is interpolated for Fs= 60 samples/sec



The above signal is interpolated for Fs= 18 samples/sec

From the 3 sampling rates for interpolation of the same signal we can clearly observe the change in the way the signals are reproduced.

In the First interpolation we can observe that we lost the central peak at sample no. 7. Also the peaks have become different causing loss of data.

Second interpolation is for sampling frequency equal to Nyquist rate. We can clearly see that almost all the peaks and valleys are at proper position as the original signal. This clearly shows that anything above Nyquist Rate should improve the regeneration of signal.This is because it takes 2 samples every oscillation. This ensures the peaks and changes are not gone unobserved.

Third interpolation is at a sampling rate much lesser than Nyquist rate. It is clearly observed that there is abundant loss of information and is undesirable in general. The plot produced is completely wrong in this interpolation.

From the above discussion and observation it can be inferred that for different sampling rates there shall be different plots produced. Also linear interpolation is a bit fuzzy to compare the graphs with original, but when we use the kind= cubic then we see that sampling at anything above Nyquist shall produce a plot very much close to the actual plot.

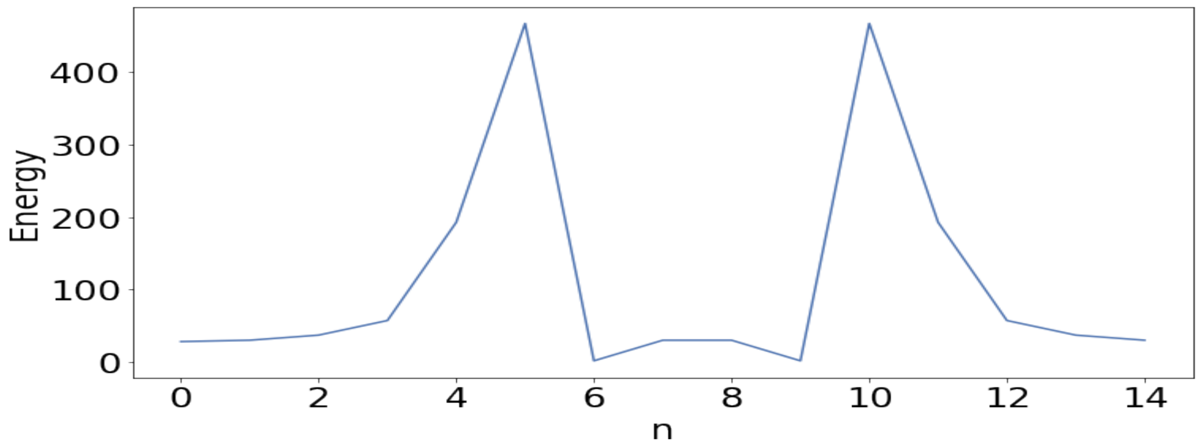
(v)

$$\mathrm{X}(\omega) = |F(\omega)|^2$$

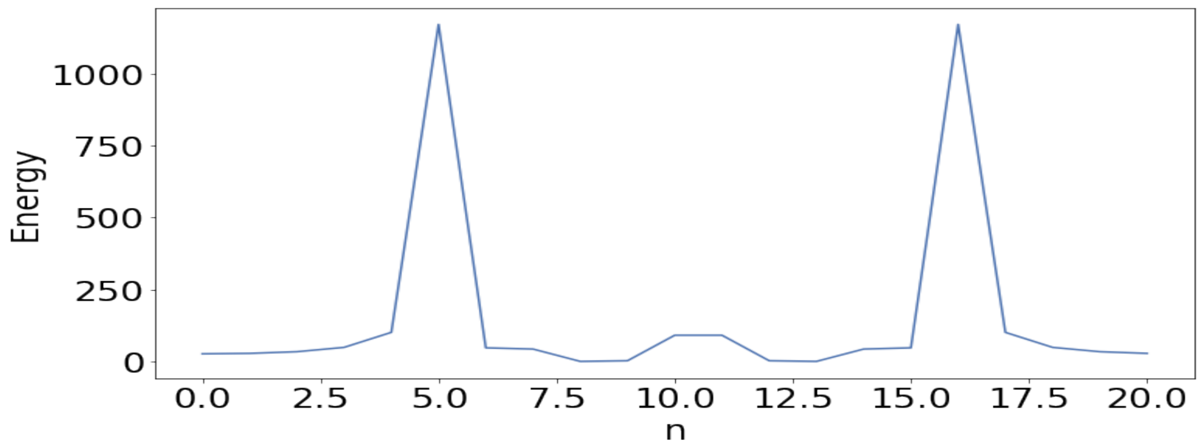
Here, $\mathrm{X}(\omega)$ refers to energy of the signal of an fft.

$|F(\omega)|^2$ refers to the amplitude of the transform obtained of a time domain signal. Since this transform contains complex values we need to take absolute values and the amplitude of the transform.

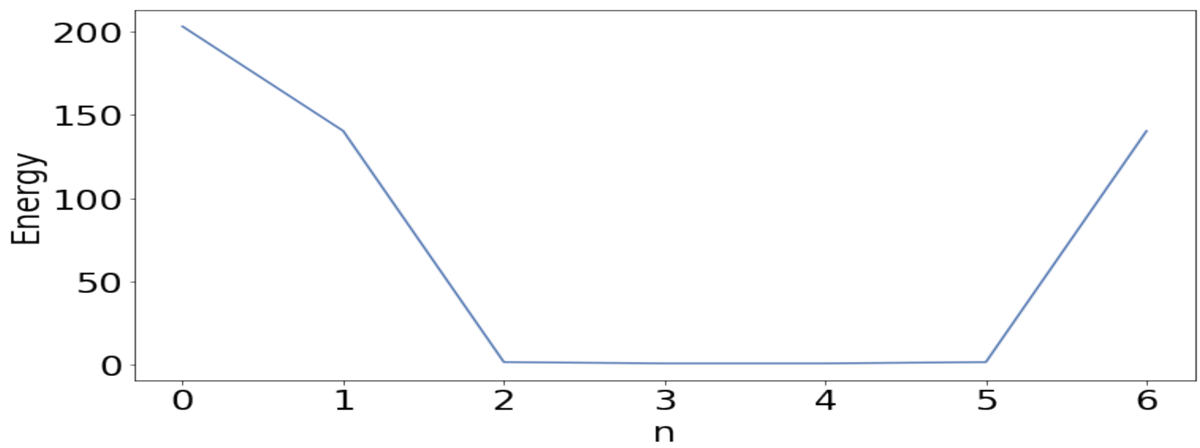
Energy for a signal is given by the above formula.



Fs = 42 samples/sec



Fs = 60 samples/sec



Fs = 18 samples/sec

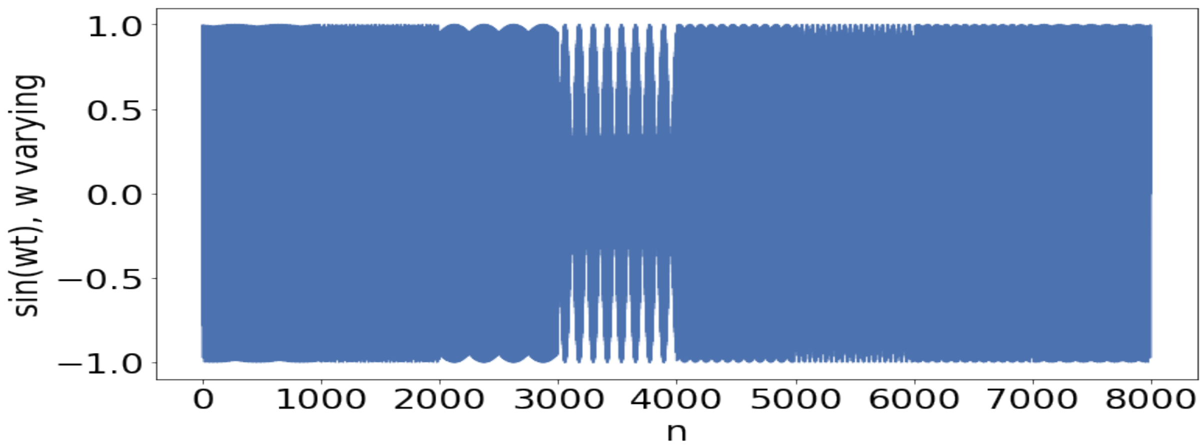
2 Problem 2

In this problem we need to play the tones corresponding to the famous and basic "Do re mi fa so la ti do ". On doing some research on these tones and components it is noted that these tones correspond to the following frequencies *261.63, 293.66, 329.63, 349.23, 392.00, 440.00, 493.88, 523.25* respectively.

As given in the hint these tones are basically sinusoidal functions:

$$\text{y(t)} = \sin(2\pi f t)$$

On appending these signals one after the other we get the desired tone of Do re mi.



Appended Signal for Do re mi fa so la ti do

On trying different sampling rates for the given appended signal we observe that on increasing the undersampling factor there is always some loss of data which is observed and heard in the voice as noise. There are no more changes that can be seen in this.

3 Problem 3

The problem asks us to analyze the given Track for different sampling rates.

I have performed various sampling rates for the Track003.wav . The file is read and the given sampling frequency turns out to be 48000Hz. For resampling/downsampling there are two methods which can be used:

(i) **resample fucntion** directly from the library. Arguments is the original signal array and the factor by which the signal needs to be resampled.

(ii) **Manually select the samples** from original array of samples according to the sampling frequency. Eg: For 1/3 rd sampling rate pick every third element of the original sample and save this array to a .wav file.

I have scaled/ downsampled the frequency by :

(a) **1/2** : There is no significant or identifiable changes observed while playing the original audio. Although some minor noise introduction is suspected in the background low frequency voice.

(b)**1/3** : There is some change in the background voice noticable. The sound moves toward being a little unpleasant than the original uninterrupted song.

(c)**1/6** : There is introduction of clear noise in one of the background sounds. There seems to be insignificant or no change in the principle song.

(d) **1/10**: More profound noise can be heard in this resampled signal.