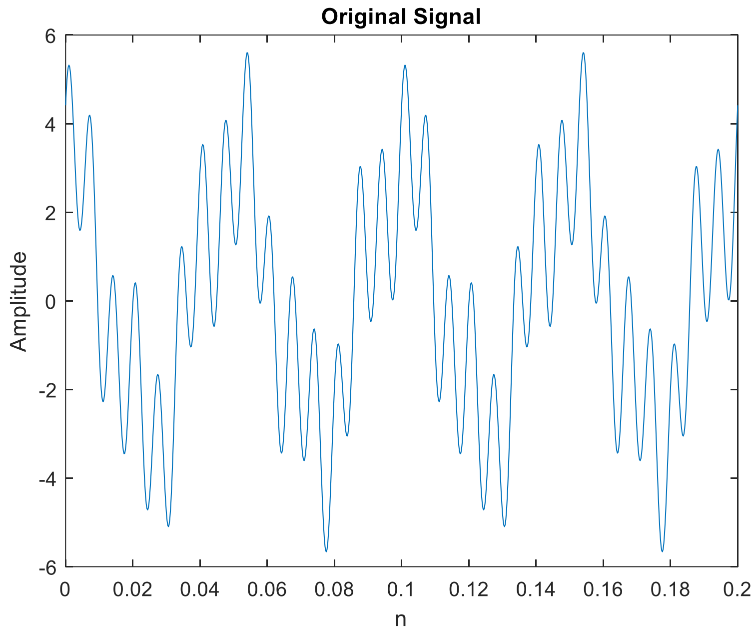
COMM 2 LAB 1 REPORT

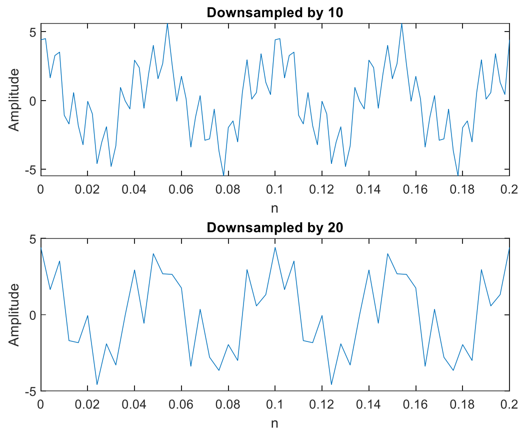
UTKU ACAR / 250206062

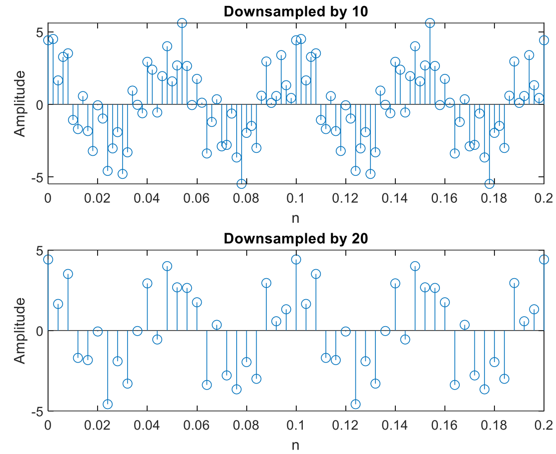
1.1.a



The graph for original signal has been constructed with anonymous function (with @ operator) and value vectors such as Amplitude, sample number (n), frequency and phase. Since every value of vectors are different so we should expect that irregular looking waves just like above. (In magnitude we can see Amplitude as 1+2+3=6 and signal have phase differences pi/2, pi/4 or frequencies like 60, 20, and 150) End time is 0.2 seconds

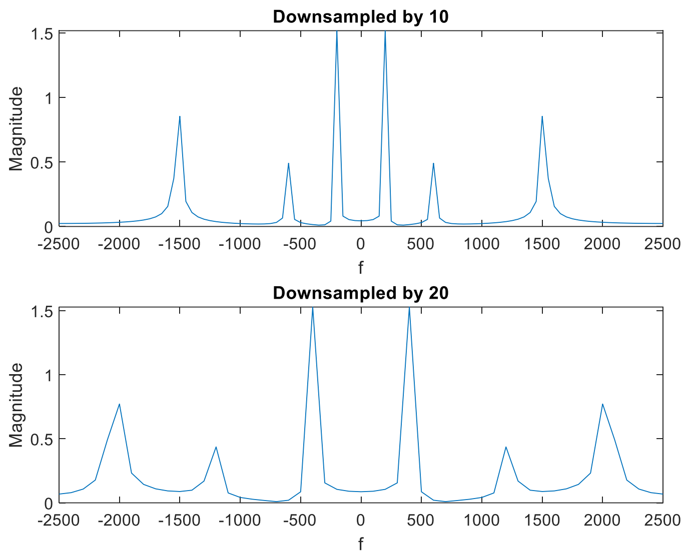
1.1.c





I have done a mistake in this with using plot() function instead of stem() function but this mistake does not affect my understanding. We can see that when we increase down-sampling rate we lose the accuracy of signal values because we are getting far away from original signal both graphical and numerical which is logical because we are getting lesser number of sample than original. Also, we can see that we got lesser number of peaks at lower graph than upper graph. My mistake gave me an idea such that what would be if this signal is continuous time signal. The corrected stem(.) version can be seen right side of the graph above.

1.1.d

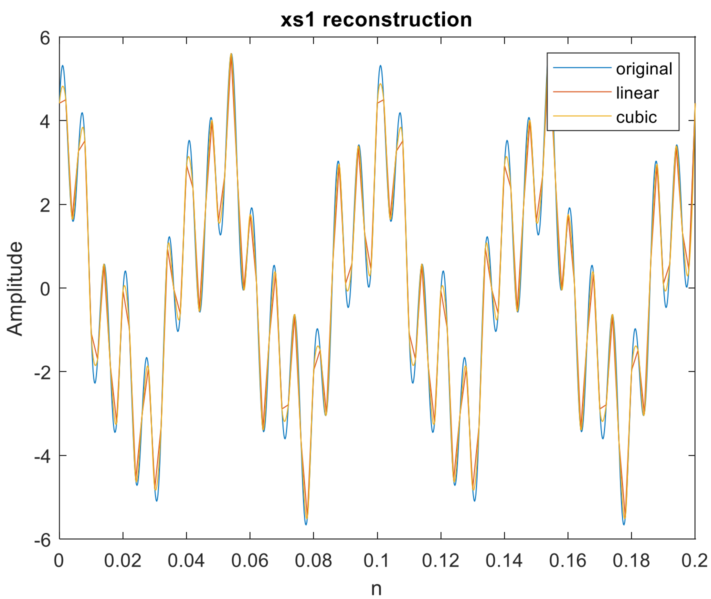


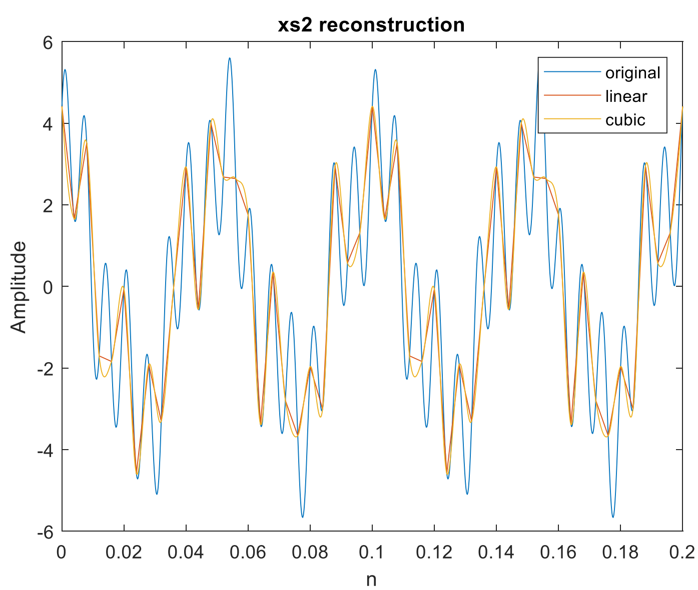
We can see that the peaks of the frequencies are different from our original signal because in the original signal it composed of 20 Hz, 60 Hz and 150 Hz signals while the signal down-sampled by 10 having peaks at 200 Hz, 600 Hz and 1500 Hz and respective negative frequencies and the signal down-sampled by 20 has peaks at 400 Hz, 1200 Hz, 2000 Hz and same points in left sided part of x axis as negatives of points. I also detected that the frequency peaks are open out a little bit, so I think that it is because of while we are getting less number point the frequency of the sampled signal varies unlike the original signal. Magnitudes are also different from original signal due to the variation process stated before.

Report Question 1 Answer: Sampling frequency and length of the signal affects the frequency resolution mainly. Formula of frequency resolution is “Fs/N” Which leads us when we increase the length of the signal or decreasing sampling frequency, we get better frequency resolution. For example, 10 Hz is better resolution than 20 Hz since we cannot detect true location of 10 Hz component of the signal by using a device with 20 Hz frequency resolution.

Report Question 2 Answer: The reason of why we saw Third frequency component (150 Hz) at 100 Hz is down-sampled signal by 20 has frequency resolution of 100 Hz (5000/50) formula has been given above. So, in xs2 we can see only multiples of 100 Hz. We cannot see 150 Hz at its own location instead we see this component at 100 Hz which is lesser multiple of 100. So, I think we should see 180 Hz component (if exists) at **100 Hz** location just like 150 Hz (because rounding smaller multiple of 100 Hz). So, the answer is Yes, we can predict the observed location of 180 Hz signal in 100 Hz resolution system and it should be at 100 Hz but it’s not logical prediction in my opinion because we will lose much data about original signal with this prediction. Regardless 180 Hz is much near value to next multiple of 100 Hz (200 Hz).

1.1.e





Report Question 3 Answer:

Reconstruction process of down-sampled by 10 signal is much like the original signal than down-sampled by 20. Reason of difference is length of the sampled signals. If we have more samples of the signal, we can interpolate signal (predict values with existing values) better. Also, we can see that cubic method is smoother than linear method of interpolation this is because cubic needs 4 data points minimum while linear requires 2 data points minimum. So, if we use more datapoints, our result will be more consistent (smoother in a way) than using less datapoints. So best result would be less down-sampled with cubic interpolation.

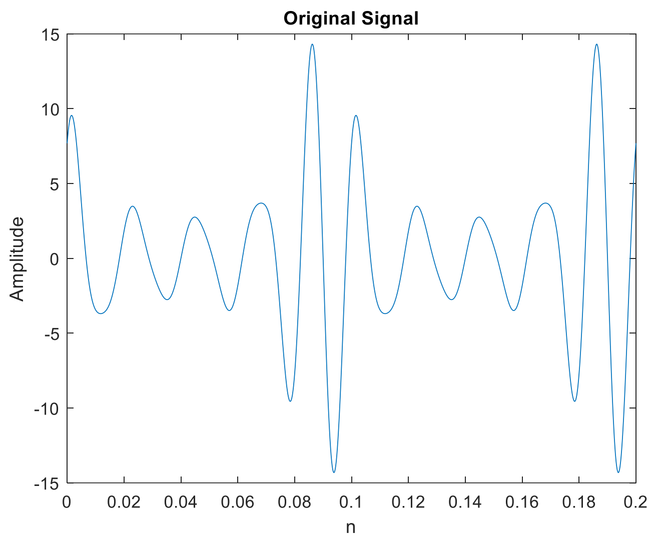
Report Question 4 Answer:

We need at least two x values and corresponding y values (x1, x2, y1, y2) and desired x point or desired y point (generally it would be x point since we calculate the corresponding value between two x points (x1=1->y1=5, x2=2->y2=10 what would be the y3 value for x= 1.5))

We can calculate the example above with this formula: $y = (y1) + (x - x1) * \frac{y2-y1}{x2-x1}$

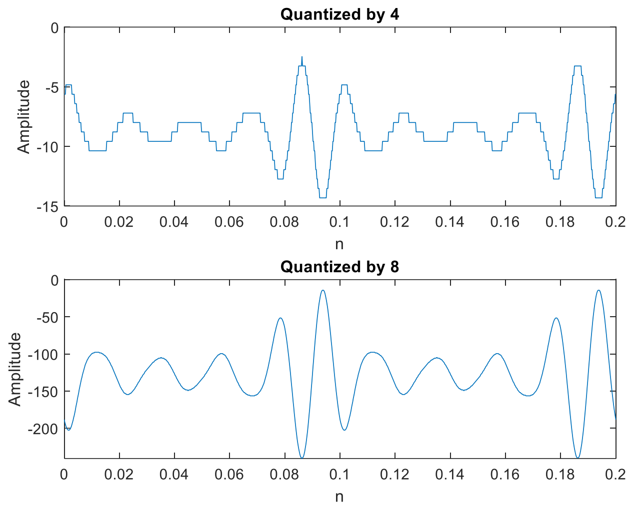
y=5+(1.5-1)*(10-5)/(1) --> 5+0.5*5= 7.5 would be desired y value and so on we can find any y value if requirements met.

1.2.a



Similar to the 1.1.a I obtained desired signal by using anonimous function and value arrays. Then I summed up all the signals which has different values of corresponding vectors due to summation symbol. It has peak amplitudes at 14.23 which is roughly sum of 5+4+3+2 which is sum of first 4 amplitudes. Reason behind why it didn’t reach to a peak point at 15 is phase difference which is another parameter of the signal’s function.

1.2.c



As I expected the signal quantized by 4 bits are more serrated than the signal quantized by 8 bits because there is more corresponding value(level) to specific time in 8 bits than 4 bits. Because 4 bits can only represent 16(2^4) different amplitude(voltage) levels while 8 bits has 256 different amplitude levels. So, 8 bits quantization result is much smoother than 4 bits. But the amplitudes are different from original signal which I did not understand at laboratory hours. Like 8-bit quantized are inversed of the original signal or has 90-degree phase difference.

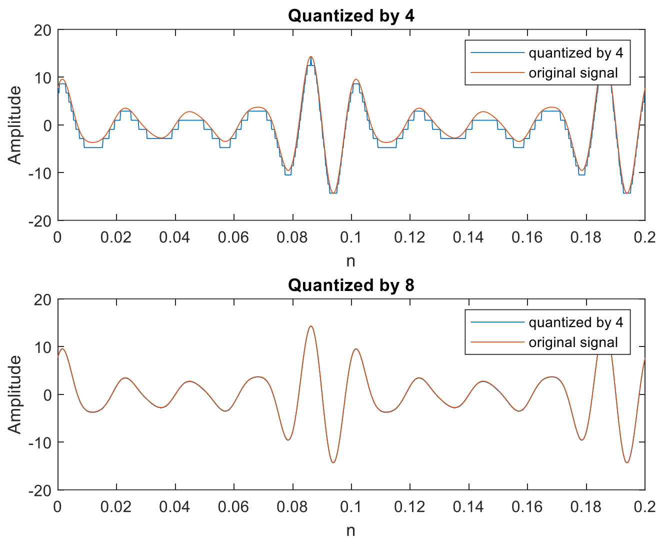
1.2.d

Report Question 5 Answer: Since we have low voltage level numbers for 4 bits than 8 bits Mean Square Error of 4-bit quantized signal should be higher than 8-bit quantized signal (insufficient voltage levels to represent). This also means that Signal to Quantization Noise Ratio of 4-bit quantized signal should be lower than 8-bit quantization since we have same original signal, the numerator part would be same for both quantized signals 4 and 8 while the error of the 4-bit signal becoming much higher which also cause lower SQNR than 8-bit signal.

But whatever I have done both in lab hours and after several hours I did not find why my result is completely inverse of what I just explained. I did mistake by using log() instead of log10() but even I have used log10() MSE of 4 bit got lower than 8 bit or SQNR of 4 bit got higher than 8 bits which is completely opposite to my logic above. I tried sum(.)/len(.) method instead of mean(.) (which is what I have used in labwork) but it gives exactly same result. Next day I have found mistake which I have explained further pages. The lab results are given below:

mse4	86.9791
mse8	1.8361e+04
sqnr4	0.3165
sqnr4dB	-11.5033
sqnr8	0.0015
sqnr8dB	-65.0267

Corrected graph and results for 1.2.c and 1.2.d has been given below:



mse4	1.2137
mse8	0.0041
sqnr4	22.6849
sqnr4dB	13.5574
sqnr8	6.6925e+03
sqnr8dB	38.2559

I have found my mistake which was in formula I need to add 1 more parenthesis to second part of the equation which was (2^N-1). Missing parenthesis is stated with yellow below.

The original equation `was that: floor(((x-a)/(b-a))*((2.^N)-1))*((b-a)/((2.^N)-1)) + a;`