REPORT

Aim

We need to recover the original signal x[n] from the distorted signal y[n] by

- First removing noise and then deblurring to get x1[n].
- Then deblurring and then removing noise to get x2[n].

Then comparing x1[n] and x2[n] with x[n].

Approach

We have used Python for solving this assignment and the following are the libraries that we used in our code.

- Pandas - for extracting and reading data from the csv file.
- Numpy - mathematical tool in python code.
- Matplotlib - for plotting the signals in the graph.
- csv - data is stored here.

Explanation

First of all let's understand the meaning of few terms,

- Fourier transform - It is a mathematical function that converts a signal from the time domain to the frequency domain.
- Inverse Fourier transform - It is a mathematical function that is used to convert a signal from the frequency domain to the time domain.
- Convolution - Convolution is a mathematical way of combining two signals to form a third signal.
- Sampling - It is the reduction of a continuous-time signal to a discrete-time signal.

Given x[n] is the original signal which is distorted and is convoluted to y[n].It is done with the LTI system h[n].

The signals get blurred and noisy due to convolution with h[n]. So to remove these both we can proceed through two paths.
1) Firstly denoise the signal and then deblurr it .
2)First deblurr and then denoise the given signal.

For denoising the signal we have use discrete time Fourier transformation of it using for loop with the array of length 193. If it's frequency is less than we have appended and more than that it goes for denoising.

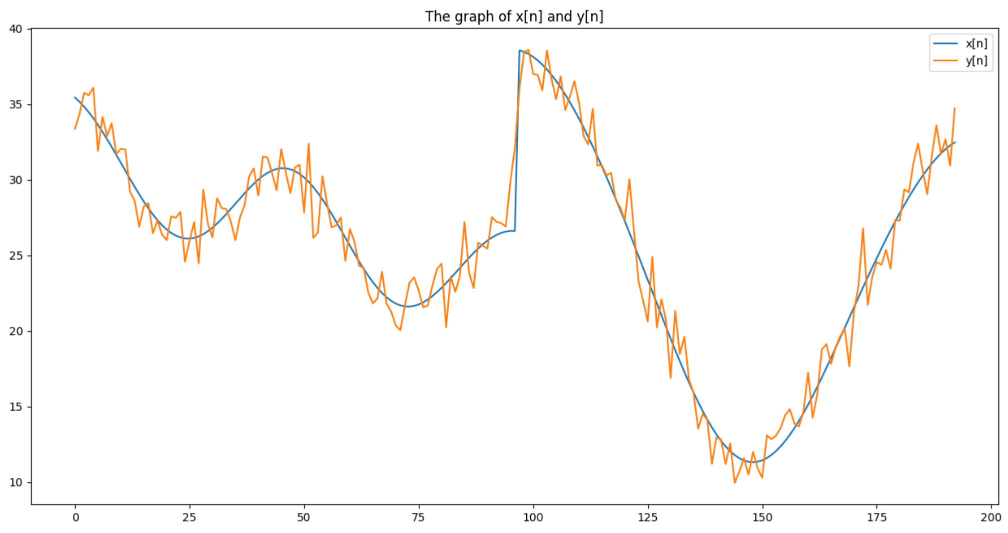
For deblurring we have taken both the signals x[n] and y[n] and then using DTFT of it we have conducted a condition statement for checking the value of impulse is whether less than 0.01 or not . In these Inverse DTFT is also needed which can easily done with the help of for loop.
Length of signal = 193
Impulse response= 5

Hence we can compare them easily just by dividing both the signals . We need to take it for certain length and then apply Fourier Transformation.
Then with the help of all these functions , we can easily plot the signal using matplotlib library.

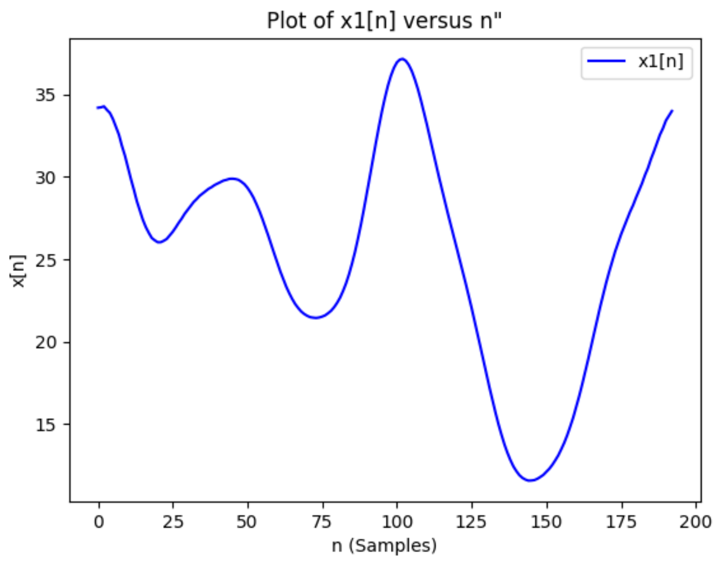
Result

We got the following plots of signals.

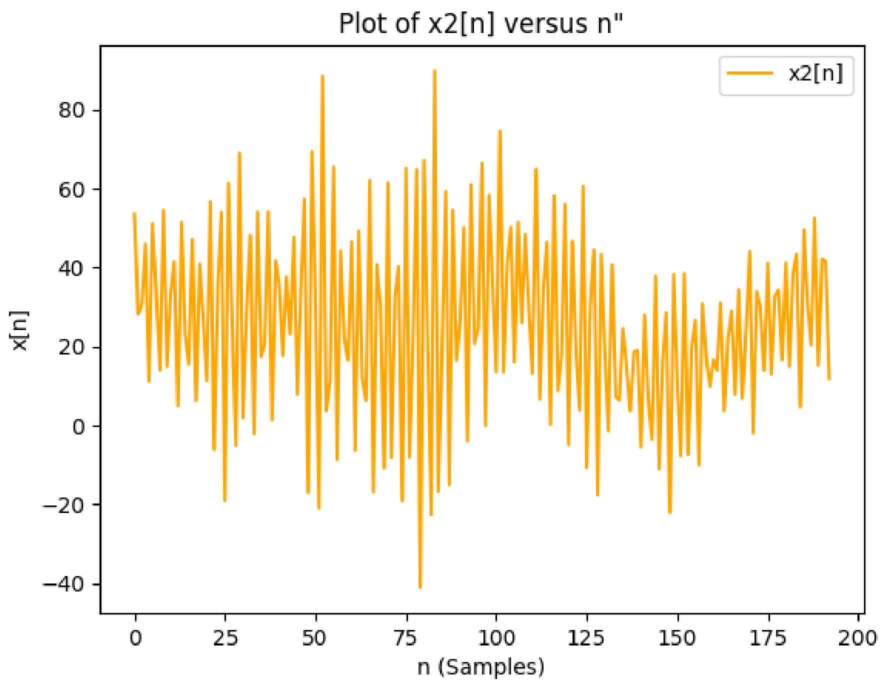
- x[n] and y[n] vs n



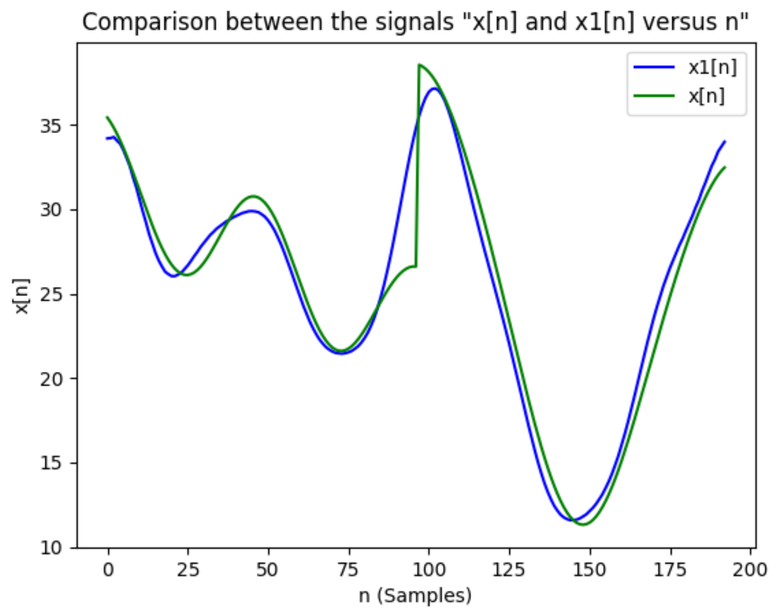
- x1[n] vs n



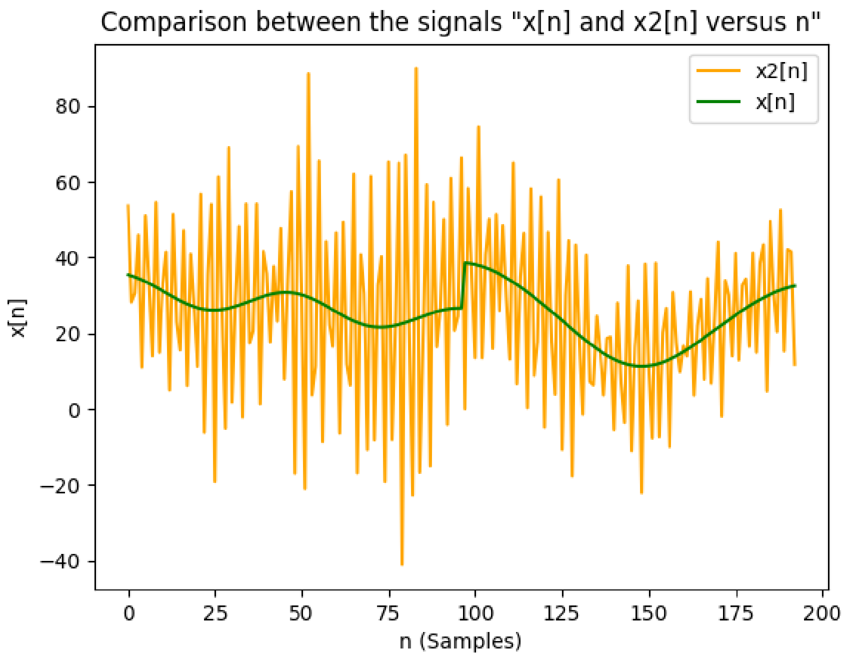
- X2[n] vs n



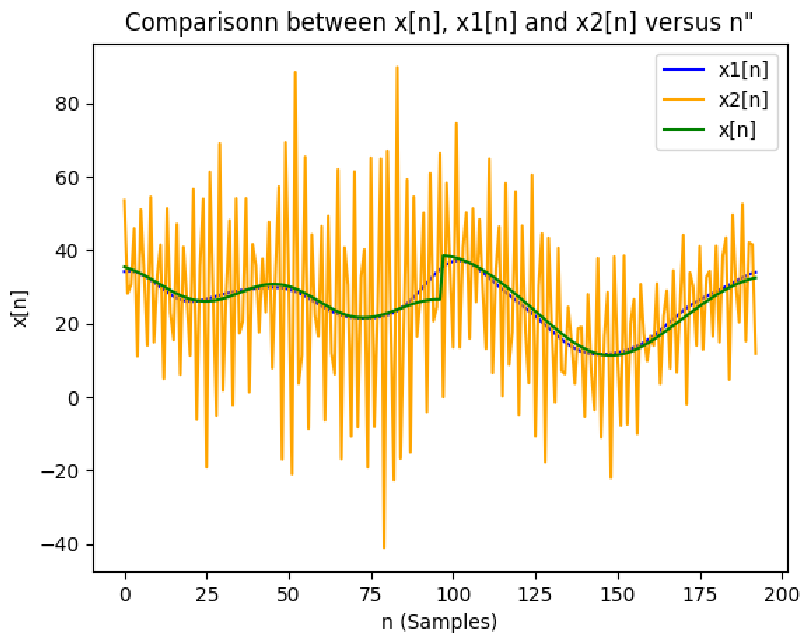
- Plot of x[n],x1[n] vs n



- Plot of x[n[,x2[n] vs n



- Plot of x[n],x1[n] and x2[n] vs n



Conclusion

We calculated the standard deviation and root mean square with respect to the input signal x[n].

STD of x1 : 0.11019406937620371
STD of x2 :1.6772470437556608
RMSE of x1 :21.267455389607317
RMSE of x2: 323.7086794448425

By observing the above values we see that STD of x2 is more than that of x1 and RMSE of x2 is also more than that of x1. By observing the plots we see that the plot of x2 is much more deviating .
So we conclude that in the original signal first noise was added and then it was blurred .