Signal and systems

Programming Assignment

Team	Roll No.
Hemant Bansal	B20CS020
Harsh Sharma	B20CS017

Problem Statement :

Given a distorted discrete signal y[n], implement the following two approaches to recover the original signal x[n].

1. First remove noise and then sharpen (deblur). Let the resulting signal be x1[n].

2. First sharpen (deblur) and then remove noise. Let the resulting signal be x2[n].

Then, compare x1[n] and x2[n] with x[n] .

Python and its dependencies :

Libraries	**Used to**
pandas	Extract the data from the csv file and perform various operations.
matplotlib	Plotting the graphs
numpy	Finding the difference between the approaches.
cmath	Getting constants like pi and complex exponents.

Approach-

1. Firstly, we read the csv file with the use of pandas.
2. Then we called the function of “denoising” where we replace the array y by the average of its surrounding 7 elements as shown in our .py file.
3. Then we called the function of “deblurring” where we first made the array of DTFT of the y[n] signal. Then we made the DTFT array of h[n] signals also. Then divide the elements of dtft_y[n] by dtft_h[n].
4. Then after making the inverse dtft of this signal. Finally we reached our plot.
5. Plotted the grapes of both approaches and compared them.
6. For mathematical comparison we found out the mean squared error of both the arrays with the original array. (for clarification, see the code)

Implementation :

1. Denoising -

   The concept for removing noise is averaging the signal with the neighbouring elements. We have taken 7 nearby elements.

2. Deblurring -

   DTFT - discrete fourier transform of the signal

   Inverse DTFT

It is given that original signal is getting convoluted with impulse response h[n]

(1/16[1 4 6 4 1]) & h[0] = 6/16;

So y[n] = x[n] * h[n]   (Convolution)

Now according to the convolution property if Y[k],X[k],H[k] respectively represent the fourier transform of y[n],x[n],h[n] at $\Omega = 2\pi k/N$.

Then Y[k] = X[k]H[k]

=> X[k] = Y[k]/H[k] (inverse) ...eq.(1)

H[k] and Y[k] can be calculated with the formula,

$$X(e^{j\Omega}) = \sum_{n=-\infty}^{\infty} x[n]e^{-j\Omega n}, \text{ Substituting } \Omega = \frac{2\pi k}{N}$$

$$X(e^{j\frac{2\pi k}{N}}) = \sum_{n=-\infty}^{\infty} x[n]e^{-j\frac{2\pi kn}{N}}$$

For H[k], n $\in$[-2,2] and for Y[k] n $\in$[0,192], n $\in$Z

Theoretical Explanation:

1. Fourier transform : Mathematical Expression to convert a given signal from time domain to frequency domain.

2. Inverse Fourier Transform : Mathematical Expression to convert a given signal from frequency domain to time domain.

3. Denoising : A signal processing method that extracts signal from the mixture of signal and noise.

Given x[n] as the original signal which is being distorted and converted to signal y[n]. It is given that h[n] causes the blurring effect in the signal. We did two things as mentioned in the question, first denoised the signal and then de-blurred which gave x1[n] secondly deblurred the signal and then denoised it which gave x2[n].

For denoising the signal, we replace the element of the array with the average of its surrounding 7 elements. To get the original signal that is x we have to first take the fourier transform F(y)=F(x)*F(h) [as convolution in time domain results in multiplication in frequency domain] therefore F(x)=F(y)/F(h).

Now in the given question the length of y is 193 and that of h is 5 hence we can't compare and divide those two signals. So we need to sample both of the signals to a certain length and then divide their fourier transforms. By taking the inverse Fourier of that we can get back the original signal.

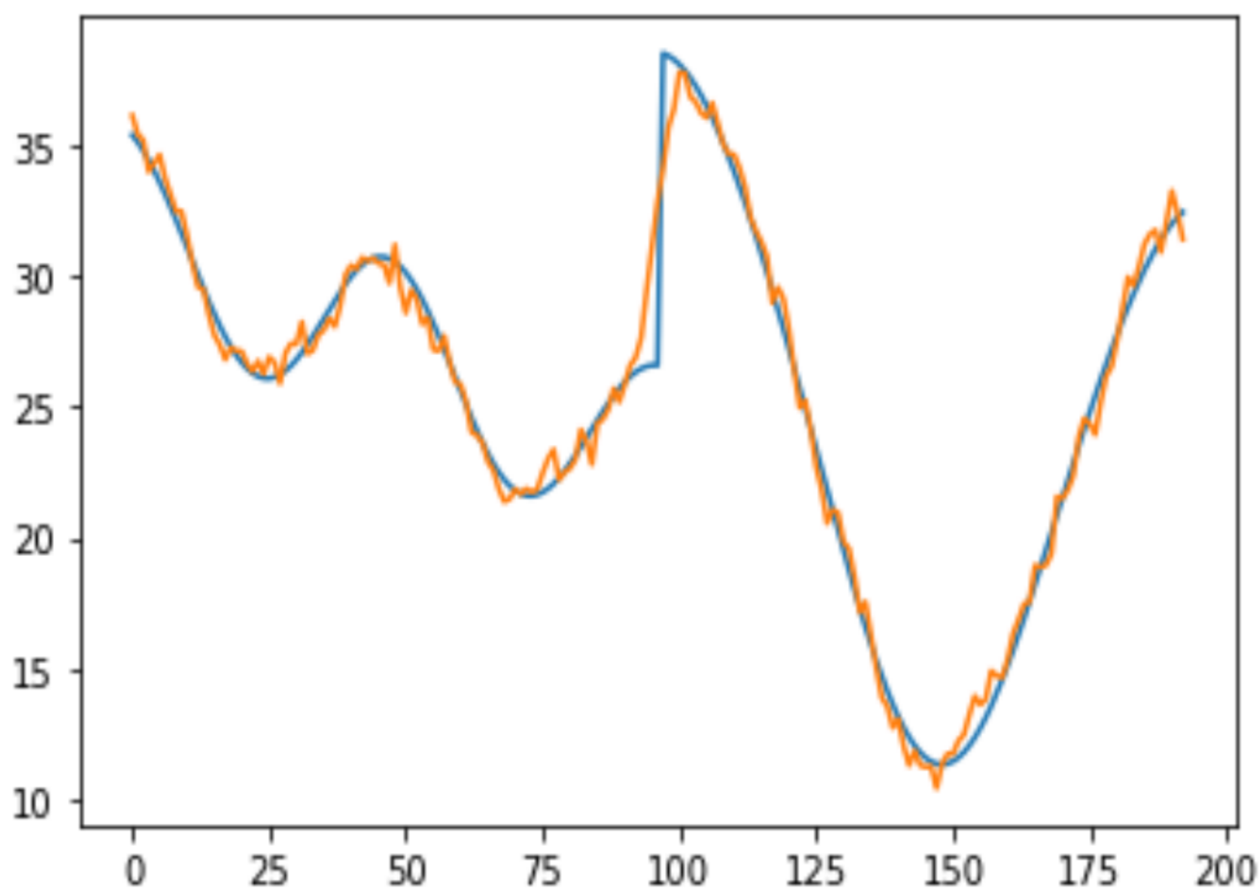
For the first approach, first we did denoise and then deblurred the signal.

For the second approach, first we did deblur and then denoised the signal.

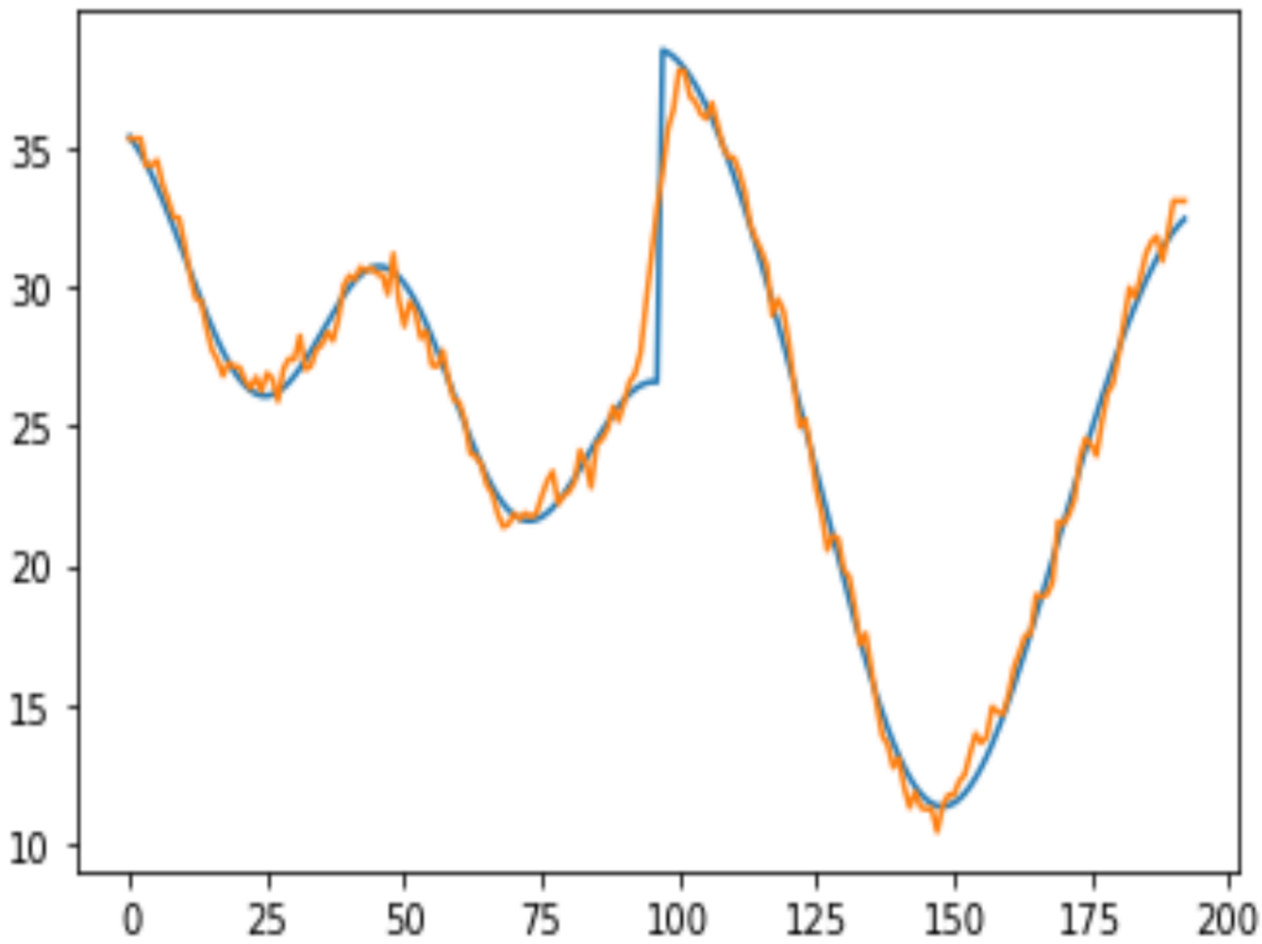
This is the theoretical logic behind the code.

Screenshots of the plots :

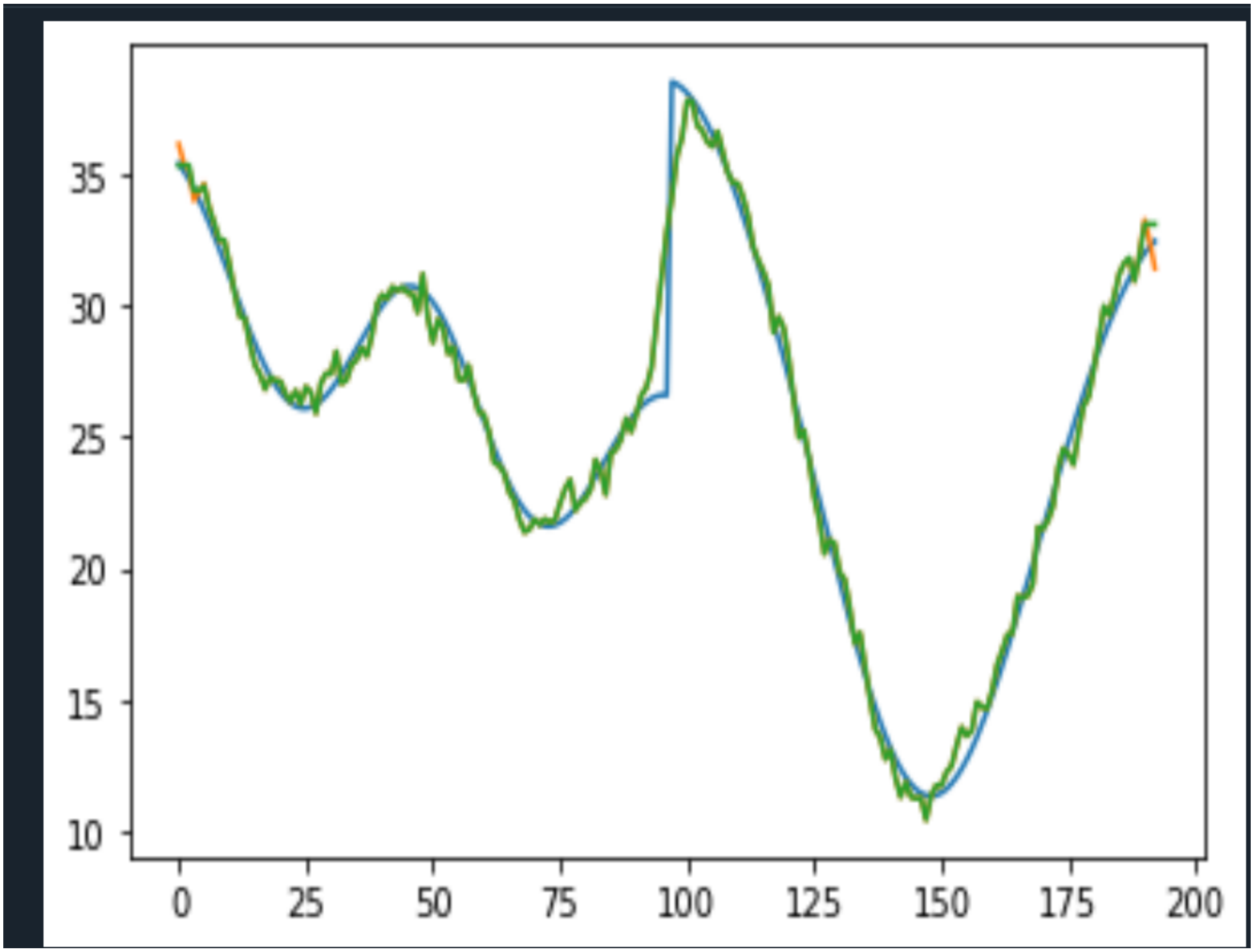
1) x1[n] and x[n] vs n :



2) X2[n] and x[n] vs n :



3) X1[n], x2[n] and x[n] vs n :



Conclusions :

From the two graphs and after the calculation of the mean squared error with respect to the main signal x[n], calculation of correlation between the signal and the x[n].

MSE between x1 and x = 0.880621447374584

MSE between x2 and x = 0.87492873241291

We can observe from the values that MSE of x1 is slightly higher than that of MSE of x2 and Correlation of x1[n] is slightly higher than that of x2[n].

So, the second approach is more efficient than the first approach because the MSE of x2 is less than x1. Thus, x2 has slightly less error than x1.

So, we can conclude that the original signal was first blurred and then noise was added to it.

Contribution :

Hemant Bansal	***Harsh Sharma***
1. Commenting and modifying the code.	*1. Main approach to the coding part.*
2. Made the report and readme file.	*2. Coded the deblurring and denoising part.*
3. Drew errors in code and mathematical conclusions.	*3. Drew graphical conclusions.*

THANKS :)