Signals and Systems

Programming assignment

Names	Harshit Mathur & Anurag Bhat
Roll no.	B20CI017 & B20MT007
Language Used	Python

Theory,Observations and Final Conclusions.

AIM –

To process input which has undergone blurring ,additive noise and obtain 2 recovered signals –
1. First remove noise and then sharpen (de-blur). Let the resulting signal be x1[n].
2. First sharpen (de-blur) and then remove noise. Let the resulting signal be x2[n].
Then, compare x1[n] and x2[n] with x[n].

THEORETICAL EXPLANATION –

First lets look at the different sub-methods and their use in the assignment.

1. Averaging derived from convolution – mathematical way of combining 2 signals to form a 3rd signal . It's physical interpretation is weighted average of samples (or average when all weights are 1) .It has been used in de-noising .

2. Fourier Transform : Mathematical Expression to convert a given signal from time domain to frequency domain. It has been used in de-blurring

3. Inverse Fourier Transform : Mathematical Expression to convert a given signal from frequency domain to time domain. It has been used in de-blurring.

We have been given x[n] which is distorted and in-turn changed to y[n].We wrote separated codes for de-blurring and de-noising then performed both tasks .

1.First de-noised then de-blurred gives x1[n]

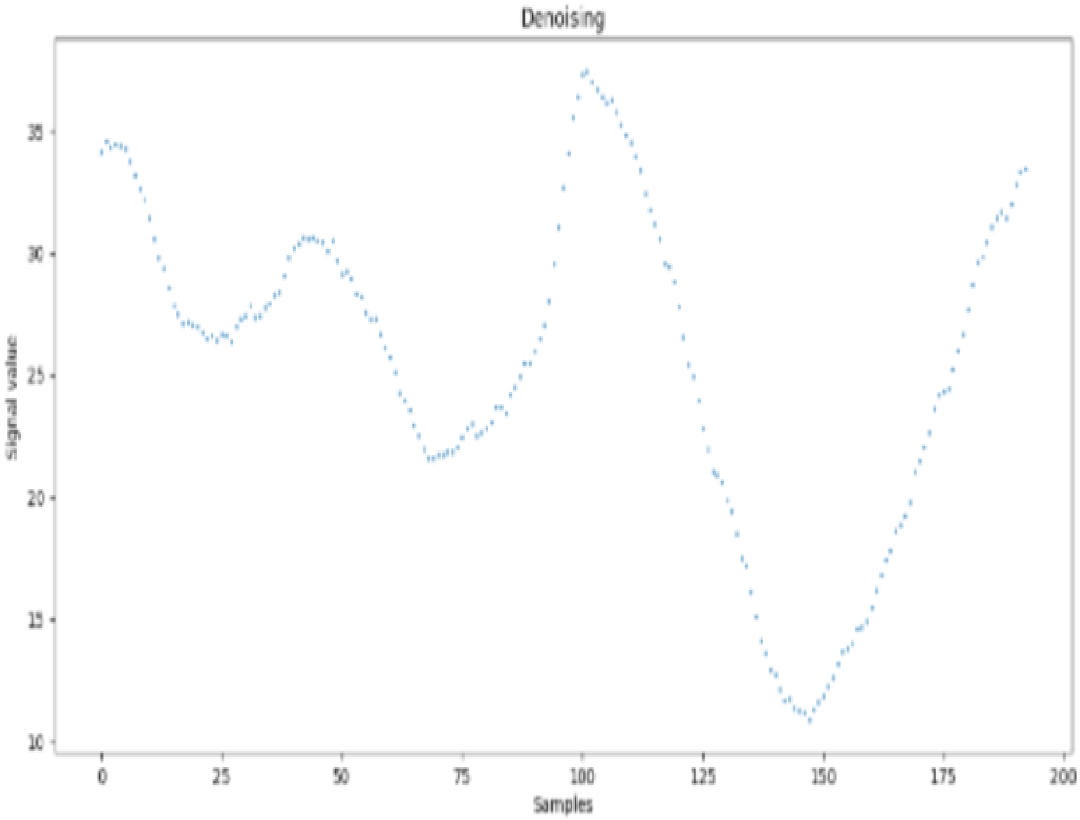
2.First de-blurred then de-noised gives x2[n]

For **de-noising** we know that noise signals are random high frequency values so we have to make a low pass filter.

We have used averaging of neighboring values technique .We derived this logic from de-noising of 2D images were a kernel of k squares each having value 1/k is convolved with the image (basically replacing a pixel with average of its neighbors).So here we used averaging with neighbors on left and right side for 1D signal .

It is also known that weighted averaging is manually the most common method to prepare a low pass filter .It is also a less effective method but minimal difference after plotting graphs motivated us to use this .

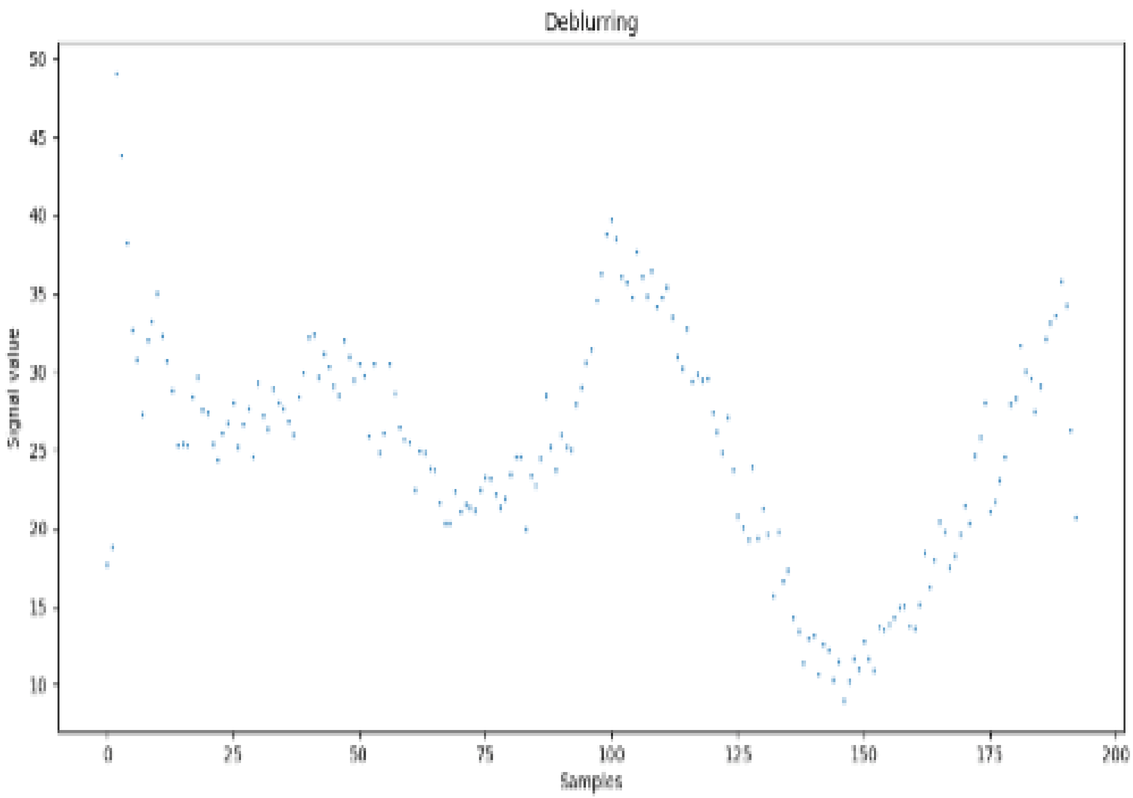
De-noising y[n] plot



For **de-blurring,** we know that Ib=convolution(Ic,h) where Ib=Blurred_signal, Ic= Clear_signal and h= Gaussian filter which caused blurring. To get the original signal that is Ic we have to first take the fourier transform F(Ib)=F(Ic)*F(h) [as convolution in time domain results in multiplication in frequency domain] therefore F(Ic)=F(Ib)/F(h).

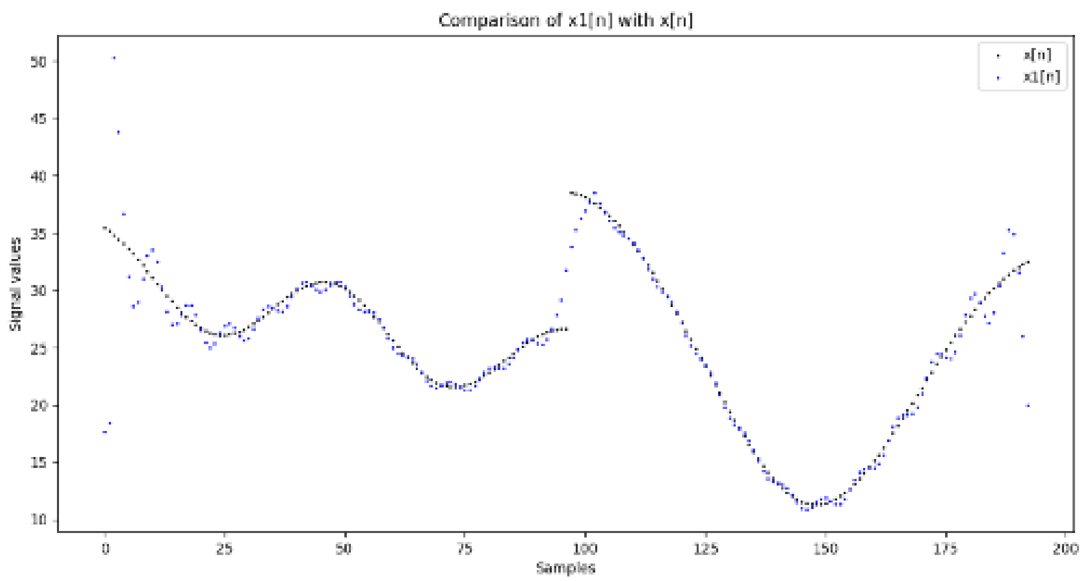
DTFT is periodic so we will only run it over a time period from 0 to 2 pi. The DTFT of Y[n] and h[n] will have that many terms as samples taken from 0 to 2pi.After dividing their fourier transforms we can easily divide them and further InvF(F(Ic))=Ic, that is by taking the inverse Fourier of that we can get back the original signal .This is the theoretical explanation of De-noising and De-blurring.

De-blurring of y[n] plot

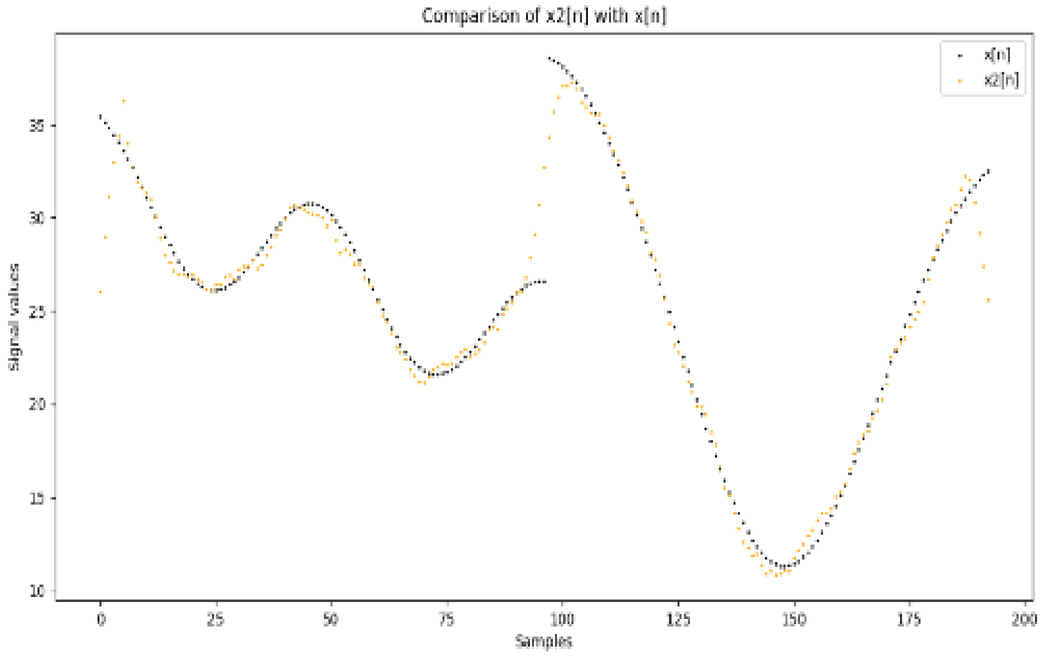


Screen Shots Of Plots For Observation-

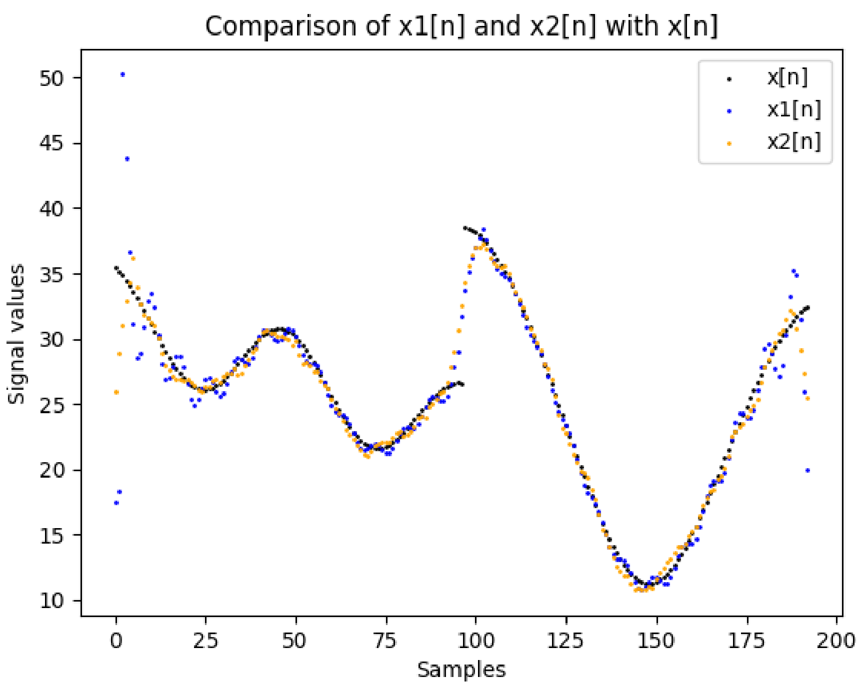
1. x1[n] and x[n] vs n



2. x2[n] and x[n] vs n



3. x1[n],x2[n] and x[n] vs n



OBSERVATIONS BASED ON PLOTS-

Based on the graphs we can observe that x2[n] is closer to x[n] than x1[n] and values are much more refined which shows that process to achieve x2[n] signal is a better technique to create an inverse system. It gives us a hint that first signal would have undergone blurring then noise would have been added to it .
Some more observations are that in both signals x1 and x2 edge distortion is observed to some degree (x2 still gives a better approximation) .
Another observation is that in range of samples 95 to 100 x[n] originally had a jump discontinuity but in x1[n] and x2[n] signals the graph values appear to be continuous .This loss of accuracy is due to our averaging method .It is actually a consequence of edge distortion itself because an edge in signals is defined as a point where abrupt change at any sample takes place .

THEORETICAL EXPLANATION OF OBSERVATION-

We have understood that blurring happens as a result of convolution with a filter and noise is just a additive high frequency signal .We have used a low pass filter to eliminate noise but then convolution with a low pass filter blurs the image further so actually de-noising and de-blurring are counter acting each other .It is clear that x1 and x2 would not be same .**x2[n] is better than x1[n]** because of better correlation with x [n] in this process .

By plots we can see that de-blurring does not contribute to preciseness (as seen by difference in continuous samples and spread out values) as much as de-noising does .So in 2nd procedure when we have de-blurred the signal first some amount of unnecessary noise gets added which is tackled and removed when we de-noise the signal after that .This does not happen in the explicitly in the 1st procedure .

References –

1. Stack-exchange on how to read data.csv file in python using pandas.
https://datascience.stackexchange.com/questions/14021/importing-csv-data-in-python/14022

2. Stack-overflow on plotting graphs in python using mathplotlib.
https://stackoverflow.com/questions/8575062/how-to-show-matplotlib-plots-in-python

3. Lecture 27 ,28 of Prof. Rajendra Nagar(IIT Jodhpur)
https://drive.google.com/file/d/1h0fVt2kzXlZdMg-ZRnbeKUr2JOPN2pA9/view
https://drive.google.com/file/d/1rSKuqRNej70Ess11c0D4wYqPtj-h51Cj/view

4. Lecture on DTFT implementation of Prof .Manish Narwaria(IIT Jodhpur)
https://drive.google.com/file/d/1wl2ef3-nZQBb4-bHz0MyB23LLwc8tneR/view?usp=drive_web&authuser=0

5. DTFT used for deblurring stack-exchange discussion
https://dsp.stackexchange.com/questions/76333/deblurring-1d-data-using-direct-inverse-filtering

6. Alan V. Oppenheim, Alan S. Willsky, with S. Hamid-Signals and Systems-Prentice Hall (1996).pd

Contributions-

Anurag Bhat B20MT007

1.Collecting information on denoising and deblurring to find out a logic for the programming part of the assignment .

2.Rigorous brainstorming with partner to figure out possibly shortest and most efficient method for this assignment .

3.Surfing on online sources like 'stack exchange' and 'geeksforgeeks' to tackle issues we had faced for the first time on python.

4.Contributed in coding of graphs .

5.Helped in rectification of errors in the code .

6. Made theory-observation file explaining the physical interpretation of our code, comparing graphs and eventually deriving final conclusions .

Harshit Mathur B20CI017

1. Learned multiple new libraries/packages of python and used them for the first time to implement plotting of graphs ,reading external files and implementing mathematics part of the code.

2. Contributed in coding the main functions of Fourier transform and it's inverse .

3. Helped in improving space and time complexity of the code

4. Made read.me file explaining the procedure and gave theoretical reasoning for the results observed in this

assignment .

5. Commenting the code for the viewer .

THANK YOU