Signals and Systems

PROGRAMMING ASSIGNMENT REPORT

Team Members:

1) Abhishek Jilowa (B20CS001)

2) Sujal Gupta (B20MT045)

Problem Statement:

One of the many applications of Internet of Things (IoT) consists of continuous monitoring of temperature in an area. To that end, several temperature sensors are installed at different locations. These sensors measure and store the recorded value of temperature over time. However, due to limitations of hardware, the sensor memory needs to be cleared periodically and this is done by transmitting the stored values to a base unit. Assume that x[n] denotes the samples of the true value of temperature recorded by a sensor. However, it is found that the received signal y[n] at the base unit suffers from blur distortions and noise (additive). Hence, the signal y[n] needs to be first processed so that we can recover x[n] from it. Assume that blur happens via a system characterized by an impulse response h[n] = 1/16 (1 4 6 4 1)(assume that the center value of 6/16 corresponds to n = 0). Then, implement the following two approaches to recover the original signal x[n] from distorted signal y[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].

Now, compare x1[n] and x2[n] with x[n]. What conclusions can you draw from your observations?

Also,explain your observations from a theoretical perspective if possible.

Solution:

To solve this problem, it is necessary to deblur the signal and then denoise the signal, using active filters created by suitable algorithms. Thus, the original Data then can be recovered to a very large extent. The coding required to create the algorithms have been done in the Python Language. It has been done using the VScode v1.58 editor.

Libraries included:

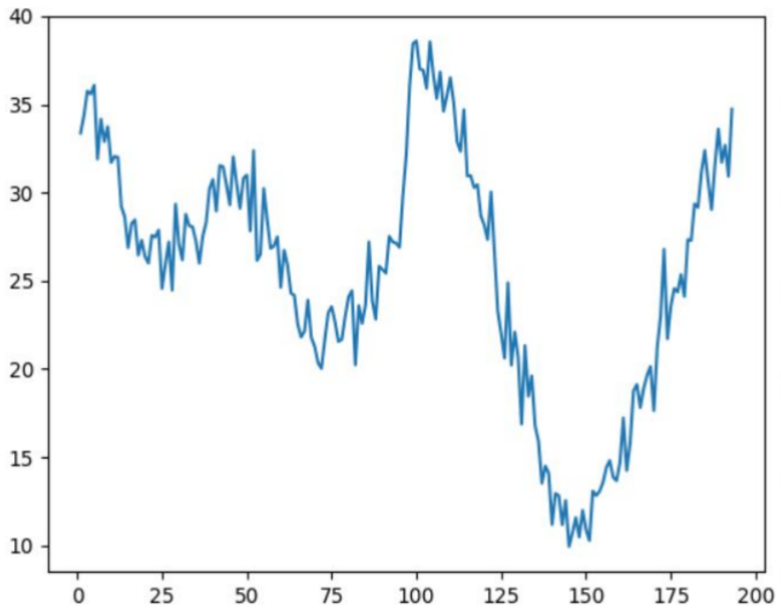
numpy

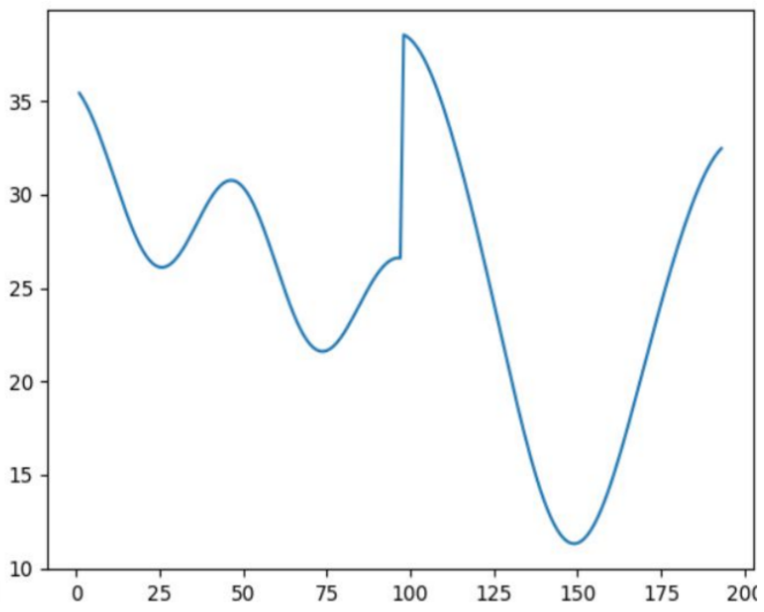
math

matplotlib.pyplot

pandas

ORIGINAL X[N] AND Y[N]





APPROACH:

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

Denoising :

We have used local mean signal denoising or averaging to remove noise from the signal. Since neighboring signal values generally (except at ends) have similar values. Therefore, we have replaced a signal value with the average of its neighbors.

We have used the convolution property for deblurring. If x[n], h[n], and y[n] are the input, impulse response, and the output, respectively of an LTI system, so that

$$y[n] = h[n] * x[n]$$

then

$$\boxed{Y(e^{j\omega}) = X(e^{j\omega})H(e^{j\omega}),}$$

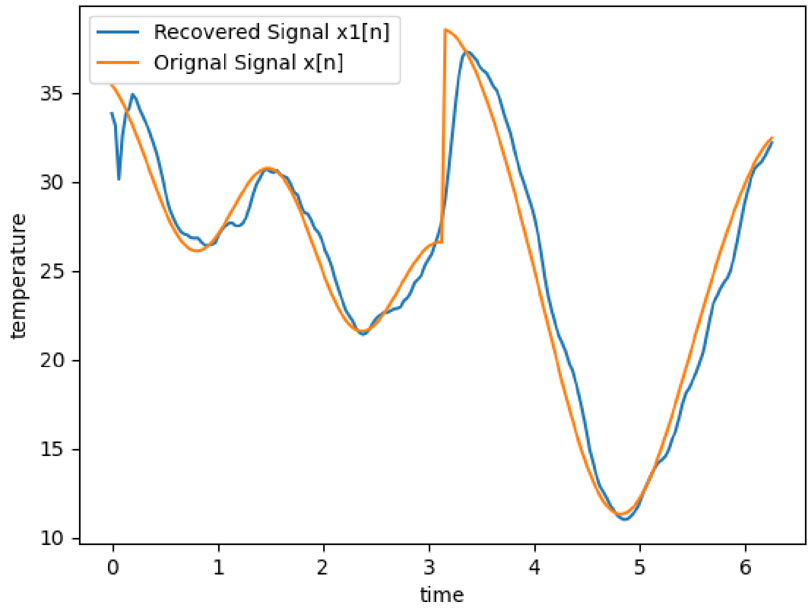
Then we can get x1[n] by taking the inverse Discrete-time Fourier transform of X(e^jw).

For finding DTFT and Inverse DTFT we have used the following expressions :

1) The DTFT y[k] of length N of the length - N sequence x[n] is defined as :

$$y[k] = \sum_{n=0}^{N-1} e^{-2\pi j \frac{kn}{N}} x[n]$$

2) The inverse transform is defined as follows :



$$x[n] = \frac{1}{N}\sum_{k=0}^{N-1} e^{2\pi j \frac{kn}{N}}\, y[k]$$

3)Denoising is done twice to improve the final signal and taking an average of 5 elements at a time.

2)First deblur then denoise. Let the resulting signal be x2[n]

First, sharpen (deblur) and then denoise

Deblurring :

We have used the convolution property for deblurring. If x[n], h[n], and y[n] are the input, impulse response, and the output, respectively of an LTI system, so that

$$y[n] = h[n] * x[n]$$

then

$$\boxed{Y(e^{j\omega}) = X(e^{j\omega})H(e^{j\omega}),}$$

For finding DTFT and Inverse DTFT we have used the following expressions :

1) The DTFT y[k] of length N of the length - N sequence x[n] is defined as :

$$y[k] = \sum_{n=0}^{N-1} e^{-2\pi j \frac{kn}{N}} x[n]$$

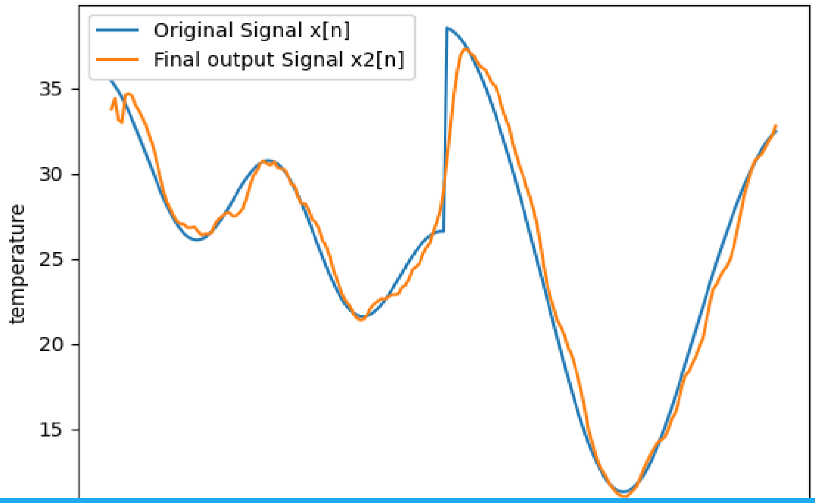
2) The inverse transform is defined as follows :

$$x[n] = \frac{1}{N} \sum_{k=0}^{N-1} e^{2\pi j \frac{kn}{N}} y[k]$$

3)Denoising is done twice to improve the final result with averaging of 5 elements at a time.

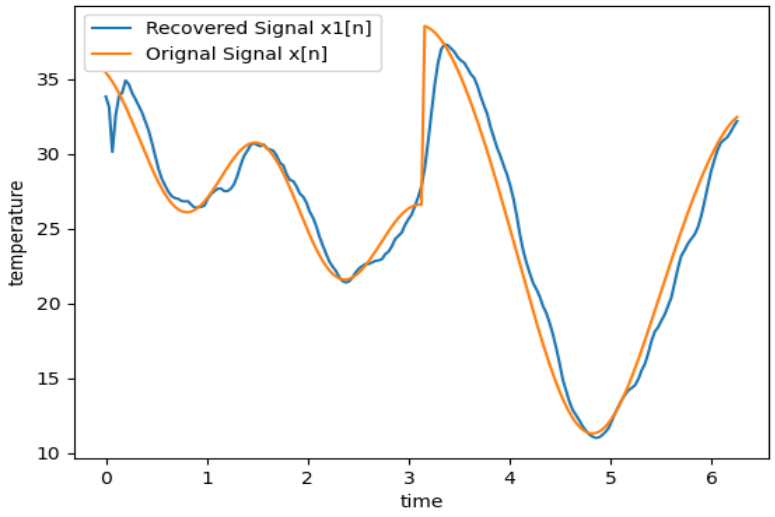
Denoising :

We have used local mean signal denoising or averaging to remove noise from the deblurred signal. Since neighboring signal values generally (except at ends) have similar values. Therefore, we have replaced a signal value with the average of its neighbors.

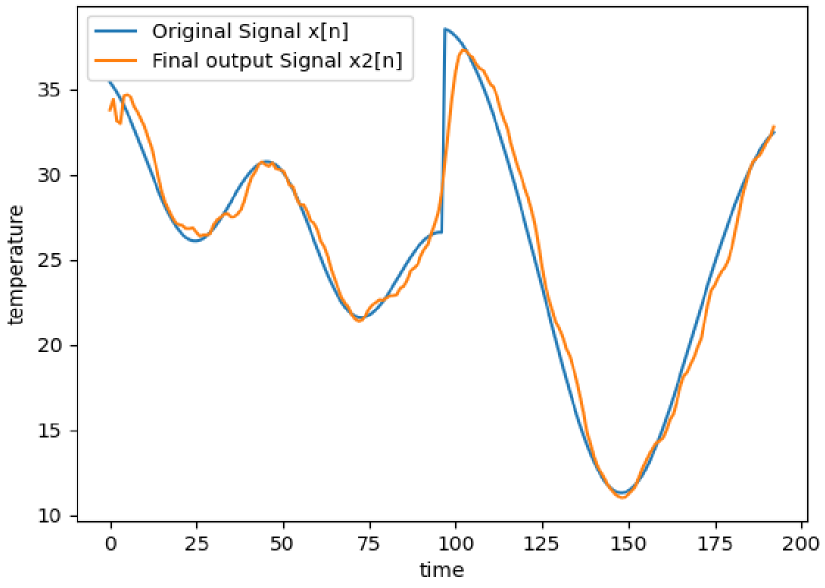


Result:

X1[n]:



X2[n]:



Conclusion:

Error in x1[n]=3.06 and Error in x2[n]=1.594

Since complete denoising didn't take place which is why we can see fluctuations in both X1[n] and X2[n]. And as a result we did not get the same signal X[n]. But we can see that X2[n] is better as compared to X1[n] because it has somewhat less fluctuations and less error and is more closer to our original signal, i.e., X[n].

We finally conclude that approach 1(Denoising first then Deblurring) is better than approach 2(Deblurring first then Denoising) in recovering the original signal.

Contribution of each team member :

Abhishek jilowa:

1) Wrote the code for denoising in both X1(n) and X2(n) and the approach part in the report.Made plots using matplot for the denoising part.

2)helped in making the report file.

Sujal Gupta:

1) **Wrote the code for Deblurring in both x1n and x2n part and file and used libraries like pandas matplot and numpy during plotting.**
2) **Made the report file and Readme.txt**