EEL2010 Signals and Systems
Programming Assignment

Group Members	**Palaskar Adarsh Mahesh (B20CI026)** **Sakshi Todi (B20CI038)**
Title	**Recovering original signal from a given recorded signal**

Aim:

To recover original signal by removing additive noise and deblurring the given recorded signal using two different methods and comparing the results with the original signal

Given:

Two discrete time signals x[n] and y[n] such that

x[n] represents the original signal that contains true values

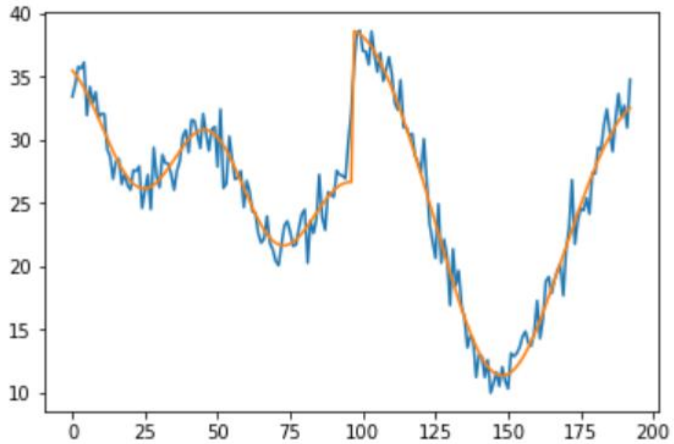
y[n] represents the transmitted signal which is convolved with an LTI system having an impulse response h[n] where h[n] is given by

h[n] = 1/16 [1 4 6 4 1]

Method 1

First removing noise and then sharpening (deblurring) the signal

How different are the given signals?

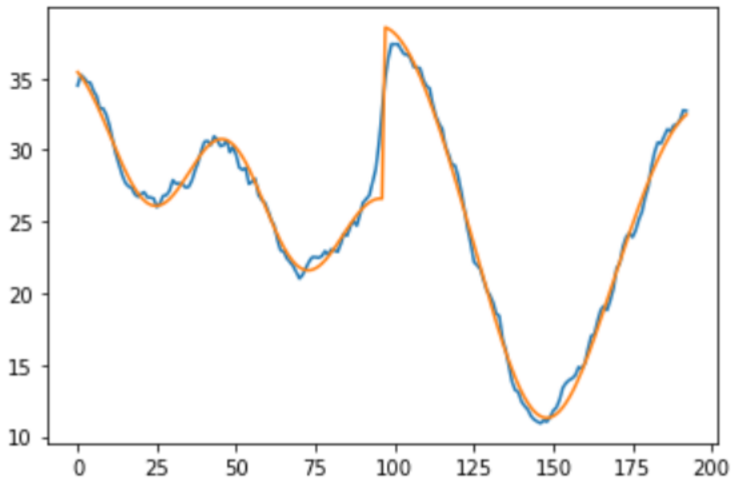


Step 1: Removing Noise

In the above graph orange line represents the original signal x[n] and blue line represents the transmitted signal y[n]. the random fluctuations in y[n] is due to additive noise. We can remove it by using moving averages.

Here we take the average of every 5 values and append it to a new array y_denoised to remove the noise. This leads to the loss of four samples which can be added manually using averages of nearby signal values.

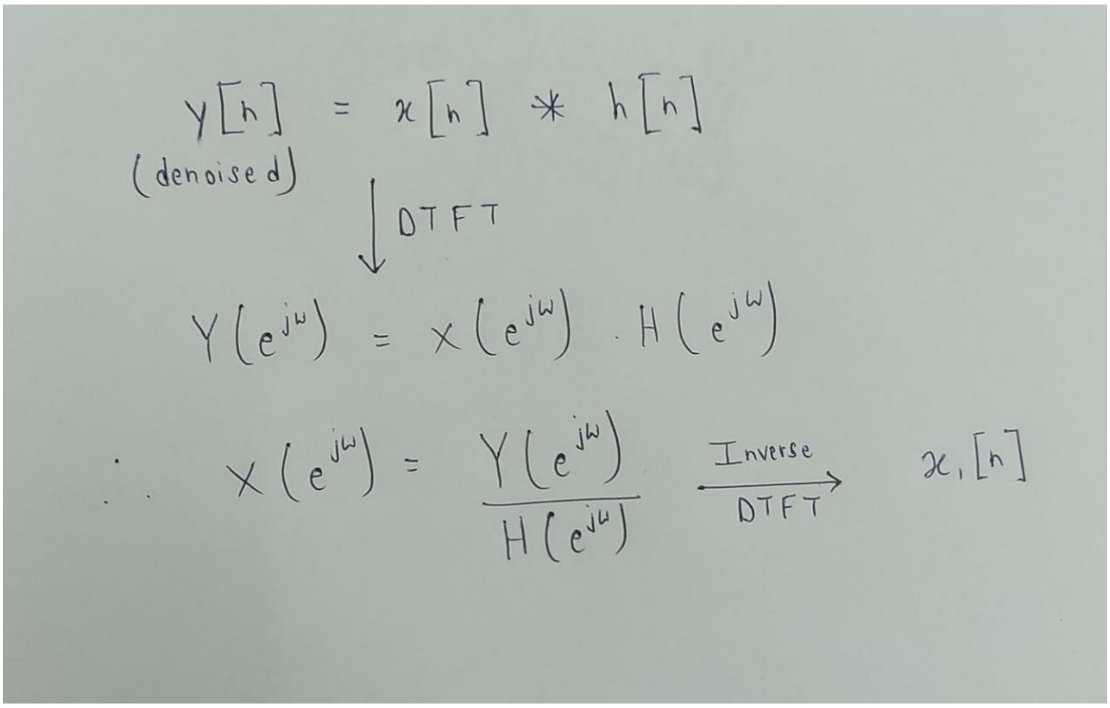
The de-noised signal is represented as follows:



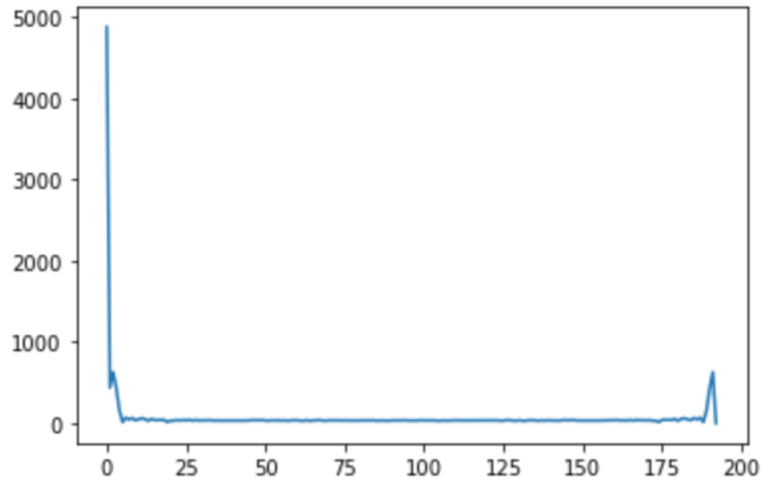
We can thus see that the random spikes have clearly reduced and the signal is approaching its true value.

Step 2:

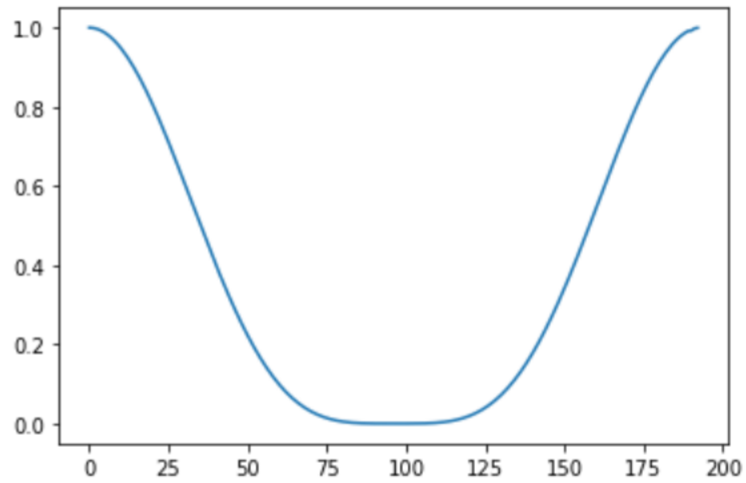
Using Discrete time Fourier transform to convert the signal to frequency domain, we can deconvolute the signal. This can be represented in a flowchart as follows:



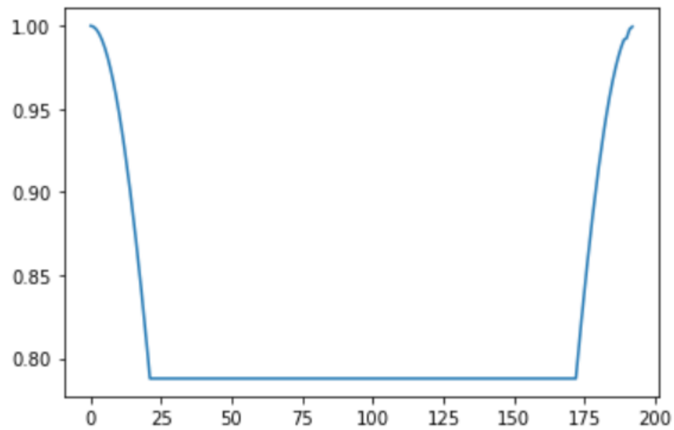
Therefore, taking DTFT of x[n] we get the following graph:



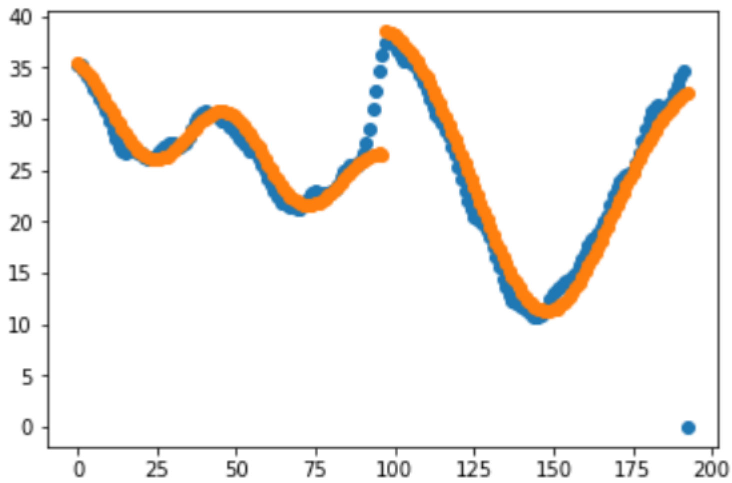
Taking DTFT of h[n] we get:



In this plot we can observe that the graph approaches zero at certain values. This indicates that we will get infinite values at these points since H(e^jw) is in the denominator. To deal with this issue, we can clip the graph at a certain value A, and all values below A can be assumed to be equal to A. The corresponding clipped graph is plotted below:



We can now take the inverse Fourier transform of X(e^jw) to obtain X1[n]



Here orange graph represents original signal and blue graph represents the obtained X1[n]

Comparison with x[n]:

To compare our accuracy, we can find out the mean absolute error. The lower is our mean error, the higher will be our accuracy.

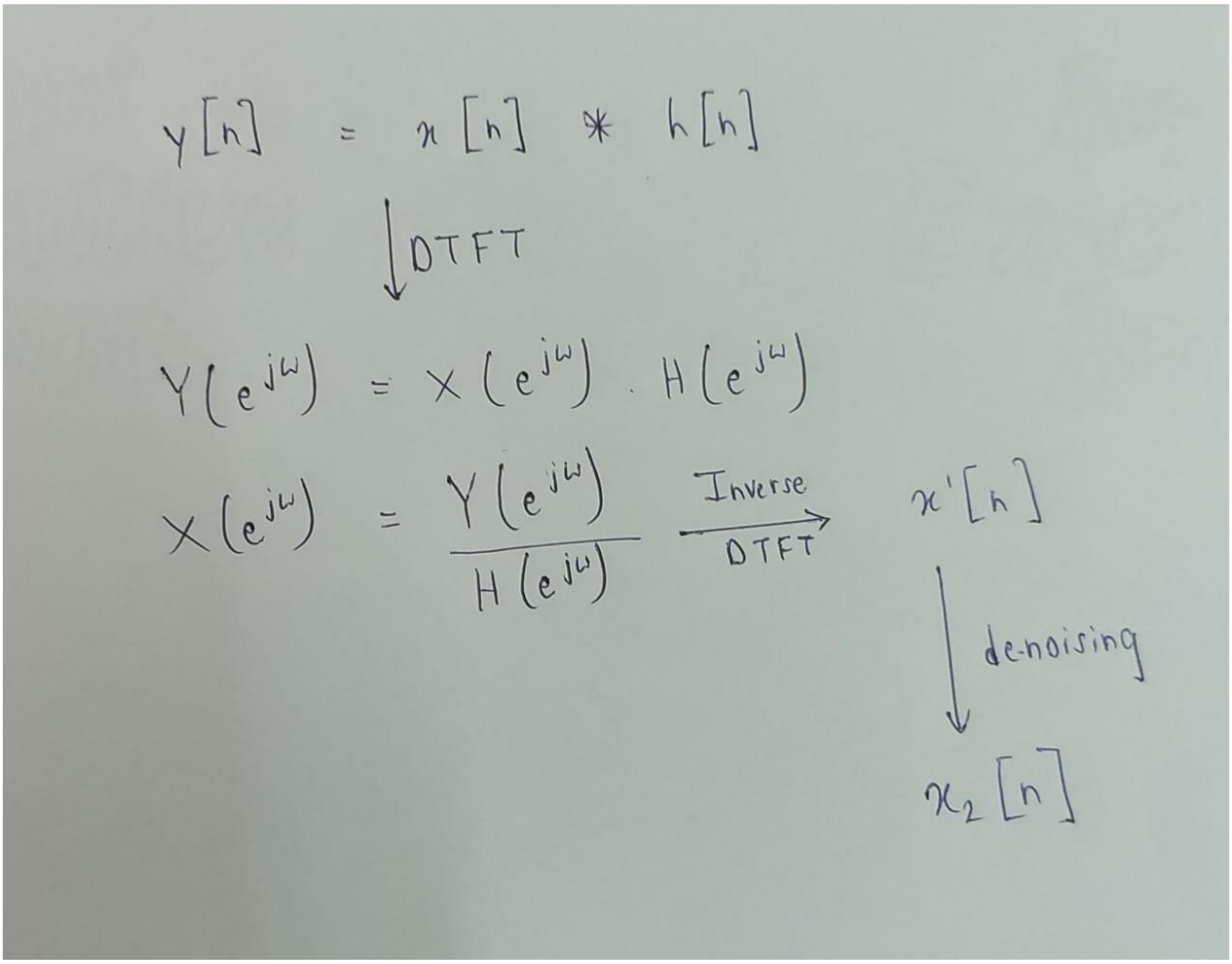
Mean absolute Error = 1/N * { $\sum_{0}^{N-1} \mathrm{abs}(\mathrm{x[i]}\ +\ \mathrm{X1[i]})$ }

Mean absolute Error for method 1 = 50.57979074862414

Method 2

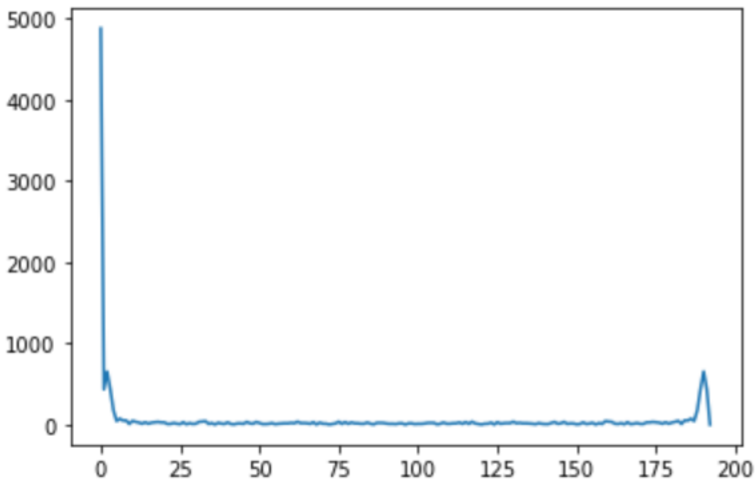
First sharpening (deblurring) the signal and then removing noise

Flow Diagram:

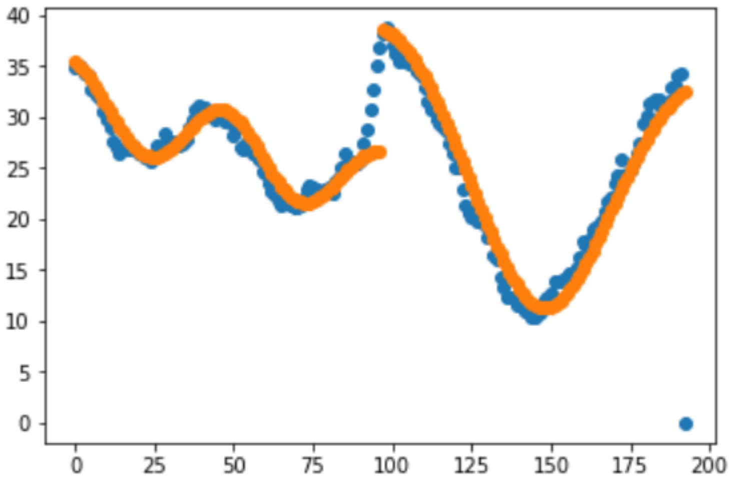


Step 1:

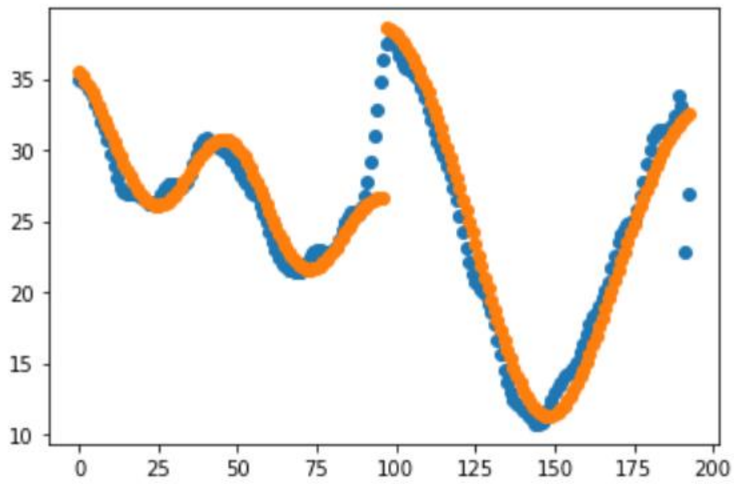
Taking discrete time Fourier transform of given x[n] we get the following graph:



The dtft for h[n] and its analysis would be similar to method 1. We would then take the inverse Fourier transform to obtain the following graph:



Step 2: Using moving averages for removing noise from the signal, we get the final plot for X2[n]



Here, orange graph represents original x[n] and blue graph represents obtained X2[n]

Comparison with x[n]:

To compare our accuracy, we can find out the mean absolute error. The lower is our mean error, the higher will be our accuracy.

Mean absolute Error = 1/N * { $\sum_{0}^{N-1} \mathrm{abs}(\mathrm{x[i]} + \mathrm{X2[i]})$ }

Mean absolute Error for method 2 = 50.622467196875064

Conclusion:

By observing the mean absolute errors of both plots, we see that error in method 1 was less than error in method 2, although not by a significant margin. But in practical applications, the number of values could be higher and we could use a different measure to find errors like the mean squared error method that will only elevate the difference in errors between these methods.

Thus, we can conclude that method 1 is a better alternative to recover original signal from the transmitted signal.