01

24 JULY, 2021

SIGNALS AND SYSTEMS

EEL2010

REPORT
PROGRAMMING ASSIGNMENT

Ayush Abrol (B20BB004)

Abhishek Rajora (B20CS002)

02

INTRODUCTION

When a Signal is transmitted, it usually results in addition of various other factors to the signal such as unwanted noise and blurring of the signal and sharpness of the signal reduces. Therefore, in the problem given, the data which is picked up by the temperature sensor is not accurate thus creating a discrepancy at user's end. We need to remove all the such discrepancies and come up with an approximation of the original signal.

OBJECTIVE

- First remove noise and then sharpen (deblur). Let the resulting signal be x1[n].
- First sharpen (deblur) and then remove noise. Let the resulting signal be x2[n].
- Compare x1[n] and x2[n] with x[n].

PROPOSED SOLUTION

This problem can be solved by deblurring the signal and denoising the signal using active systems created by different algorithms.
Thus we can acquire the approximate value of signal.

These algorithms have been coded in Python 3.9.5
The code files of the problem are attached herewith, along with this report in the zipped folder and have 6 files as follows:

- **main.py** (Includes all the functions DTFT, inverse DTFT, Denoising, Deblurring)
- **x1[n].py** (Includes the plotting of x1[n] and its comparison with x[n])
- **x2[n]** (Includes the plotting of x2[n] and its comparison with x[n])
- **comparison.py** (Includes the plotting of x1[n] and x2[n] and their comparison with x[n])
- **DTFT_of_y[n].py** (Includes the plot of Discrete Time Fourier Transform of y[n])
- **DTFT_of_h[n].py** (Includes the plot of Discrete Time Fourier Transform of h[n])

THEORITICAL EXPLANATIONS

1.Denoising: Main strategy for removing the noise that is for denoising is to use a moving average filter on y[n]. Noise is a high frequency signal which is added to the i/p. By passing a noisy signal through a moving average filter (window length =3), the opposite spikes cancel out. This leaves behind a cleaner signal

2.Deblurring: We can model this process of deblurring as $y[n] = x[n] * h[n]$. Given y[n] and h[n], we can recover the clean signal x[n] by using the property of DTFT for convolution.
We can proceed in the following way:

--> x[n]*h[n]=y[n]
--> X(C).H(C)=Y(C) {H(C)=DTFT(h[n])}
--> X(C)=Y(C)/H(C)
--> x[n]=inverse_DTFT(X(C))

Thus we retrieve the deblurred signal x[n].

PROCEDURAL EXPLANATIONS

- Denoising is simply done by taking the average of every 3 consecutive term and due to averaging, the sharp points (due to noise) will get reduced .
- We did denoising 8 times to get a more smoother and accurate curve with less sharp edges.
- Deblurring is done by firstly dividing (DTFT of y[n]) by DTFT of h[n] and then taking the inverse DTFT of X(C).
- We do division because:
  x[n] * h[n] = blurred_x[n]
- Hence to deblur we must come up with an inverse system :
  h[n] * h_inv[n] = delta[n]
  - Applying Convolution Theorem:
    H(e^jw).H_inv(e^jw)=1
  - To de-blur y[n]:
    y[n] * h_inv[n] = deblurred_y[n]
  - Applying Convolution theorem:
    Y(e^jw)*H_inv(e^jw)=deblurred_Y(e^jw)
    = Y(e^jw)/H(e^jw)

The problem which will occur while dividing the DTFT of Y[n] by DTFT of h[n] is that in a range of omega the value of h[n] tends to 0. Due to which it will approach infinity which is undesirable, hence we have created a filter which will tend to make the value constant when the DTFT of h[n] is less than or equal to 0.4.

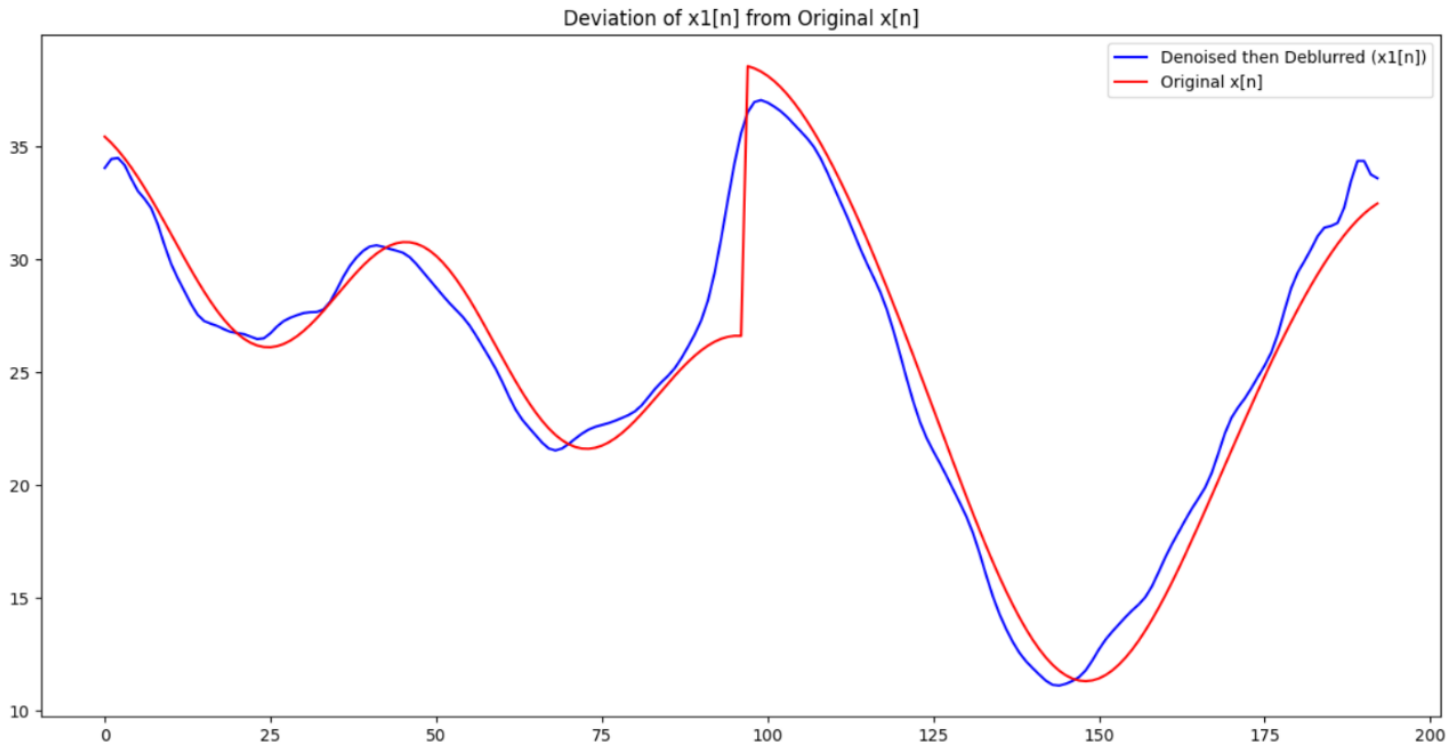
RESULTS AND OBSERVATIONS

These results were obtained after running the code to get the required signals.

PLOT 1

Denoising then deblurring the signal.

x1[n] (Blue) and x[n] (Red)

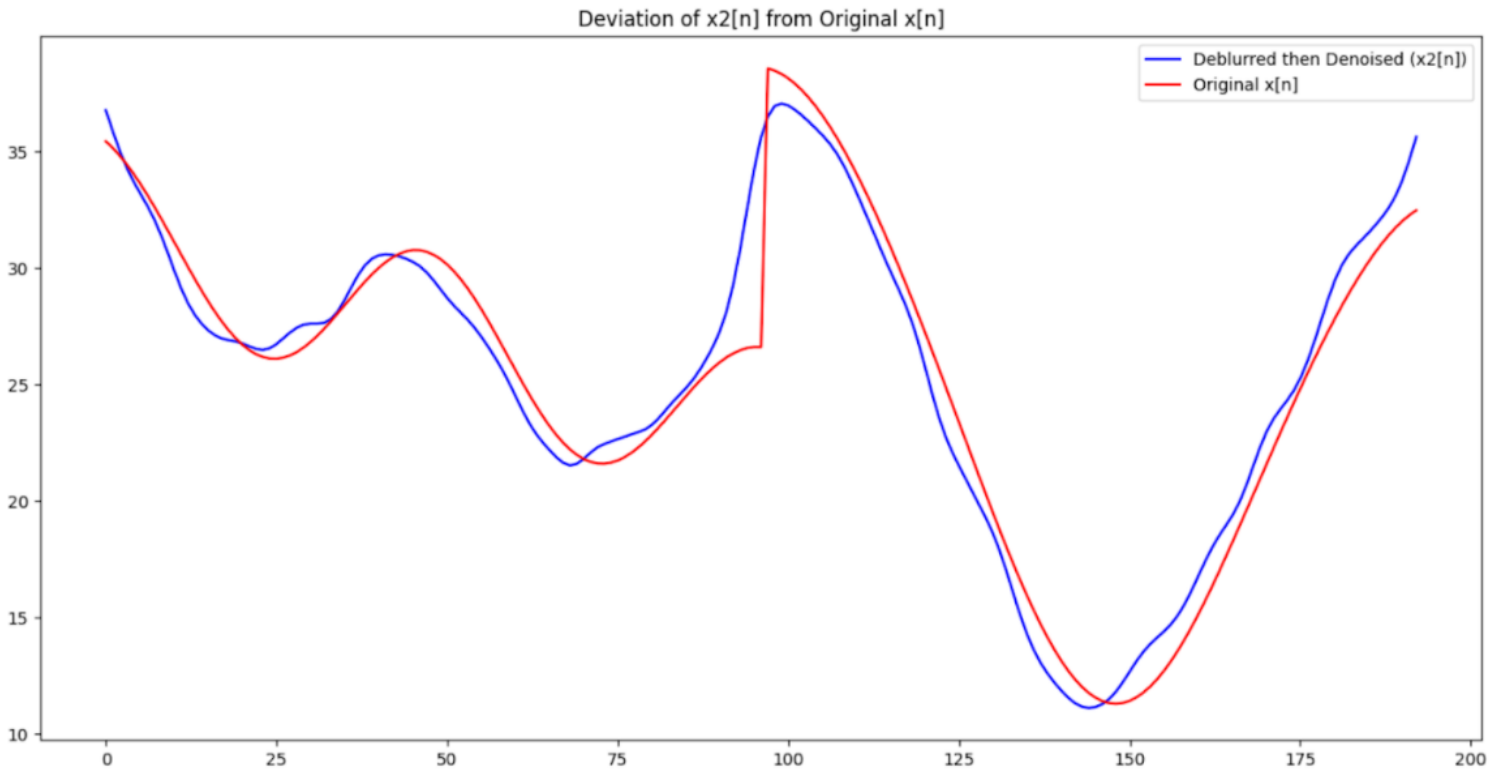


Note: This plot of x1[n] is firstly denoised 8 times using recursive code and then deblurred using DTFT and inverse DTFT.

PLOT 2

Deblurring then denoising the signal.

x2[n](Blue) and x[n](Red)

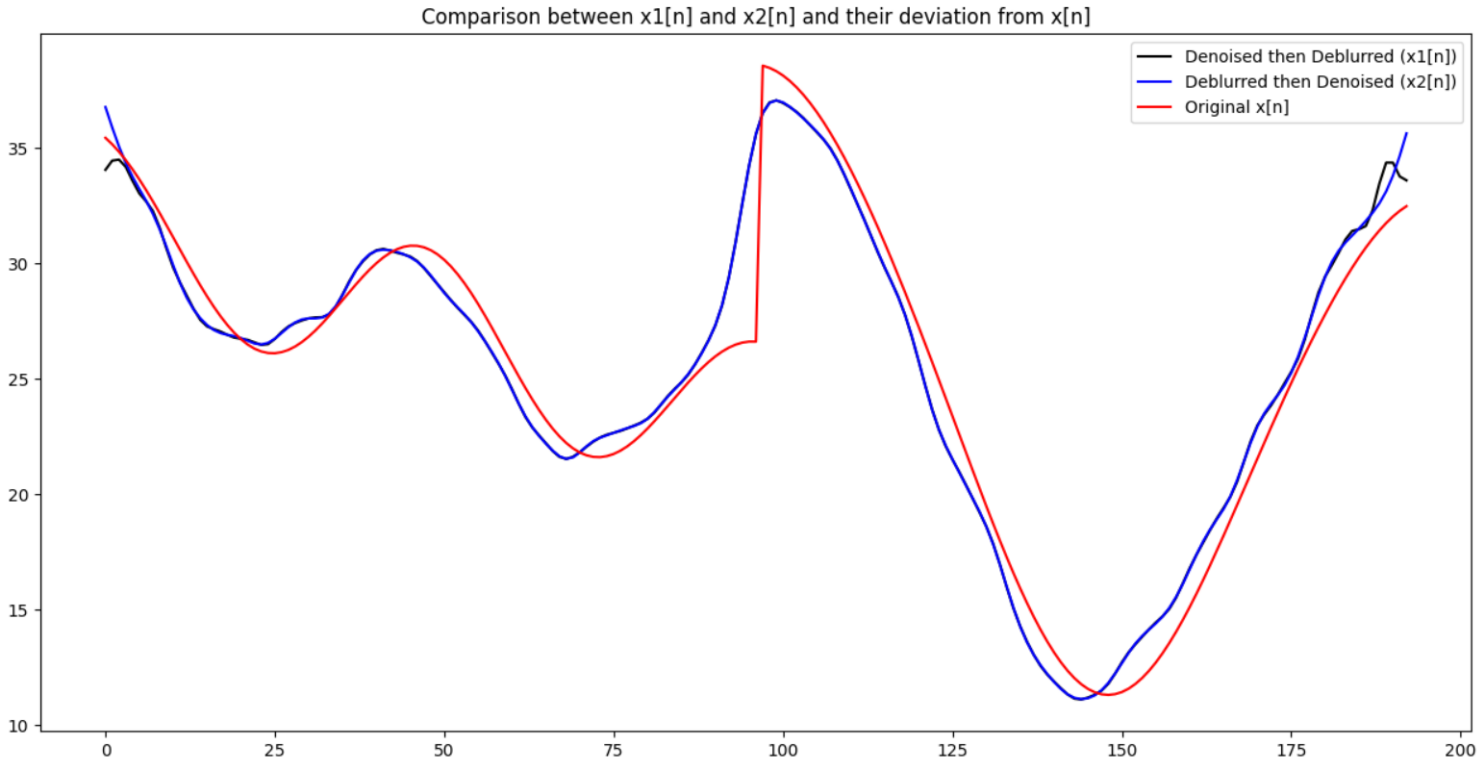


Note: This plot of x1[n] is firstly deblurred using DTFT and inverse DTFT and then denoised 8 times using recursive code.

PLOT 3

Comparison between x1[n] and x2[n].

x1[n](Black), x2[n](Blue) and x[n](Red)

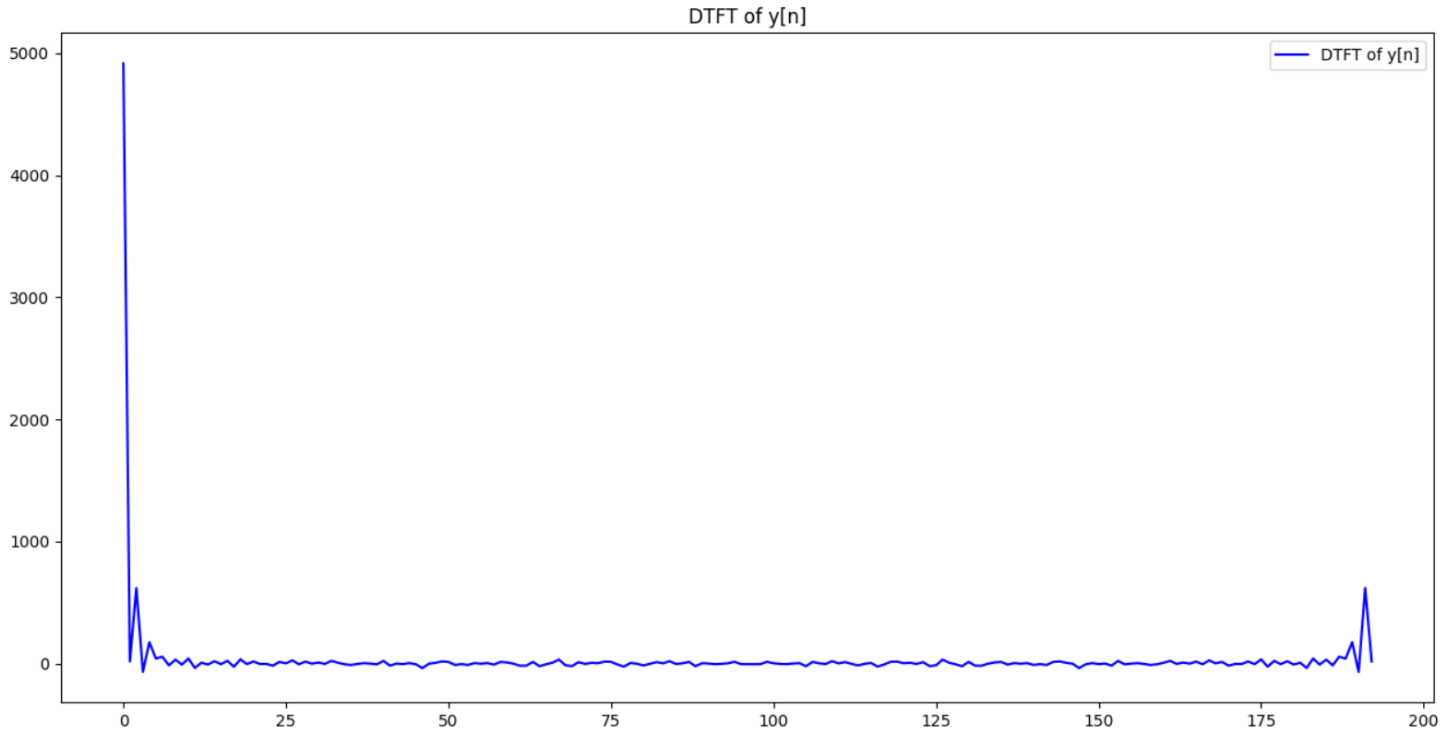


Comparison:

When x1[n] and x2[n] signals are compared, it is found that x2[n] signal is smoother and closer to x[n] than x1[n] but on a very minute level. This is because in case of x1[n], Denoising Filter was applied before the Deblurring Filter, thus the slightly blurred noises that may have been present in the y[n] were only partially removed in this case.

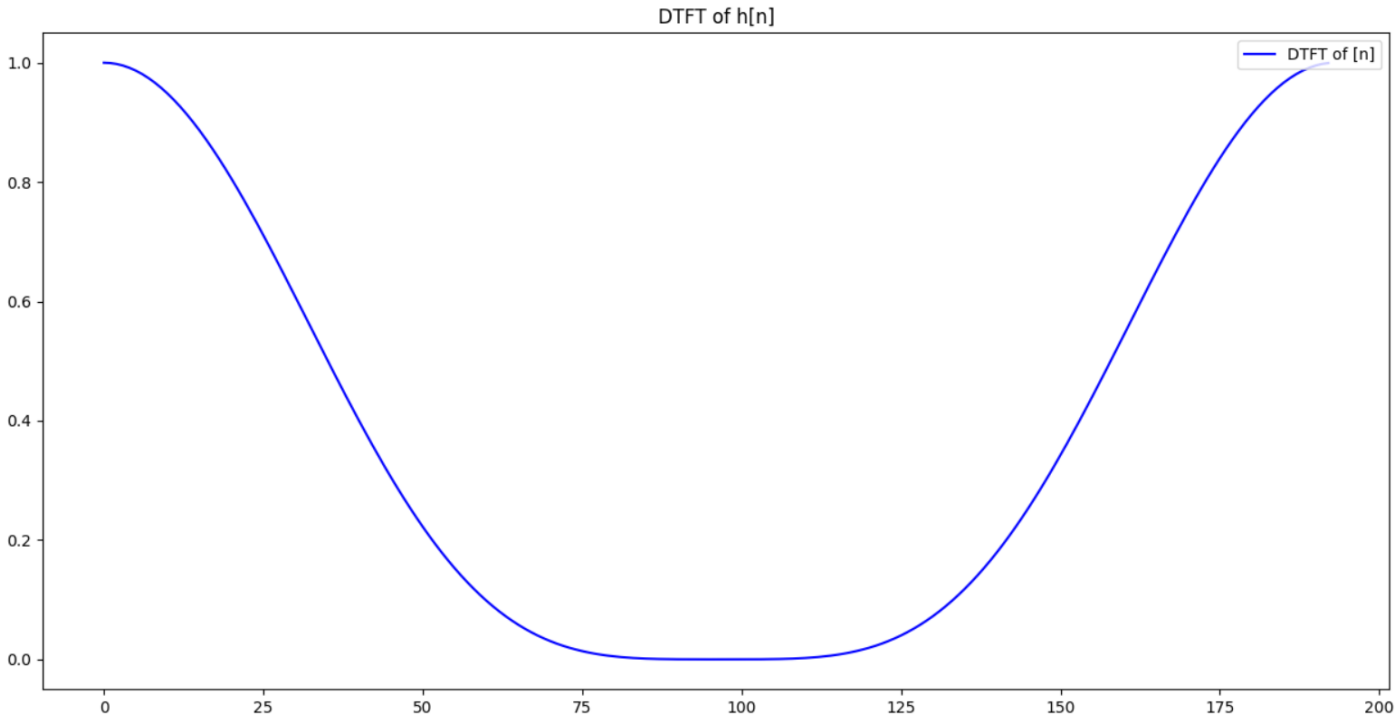
PLOT 4

DTFT of y[n].



PLOT 5

DTFT of h[n].



CONCLUSIONS:

Through this Programming Assignment, we deep dived into deeper concepts of how Signals and Systems is implemented in the real world and we realized how to denoise and deblur signals using inverse deblurring filter and a moving average filter.

CONTRIBUTIONS:

AYUSH ABROL:

- DENOISING USING RECURSIVE ASPECTS
- DEBLURRING PROCEDURAL CODE WITH FILTER FOR EXPLODING VALUES.
- HANDLED THE PLOTTING OF COMPARISON BETWEEN X1[N] AND X2[N].

ABHISHEK RAJORA:

- HANDLED THE DTFT AND INVERSE DTFT OF DEBLURRING.
- HANDLED THE PLOTTING OF X1[N], X2[N].
- HANDLED THE PLOTTING OF DTFT OF Y[N] AND H[N].