Indian Institute of Technology Jodhpur
EEL2010: Signals and Systems
Programming Assignment

Group Members:
Aaditya Baranwal (B20EE001)
Haardik Ravat (B20EE021)

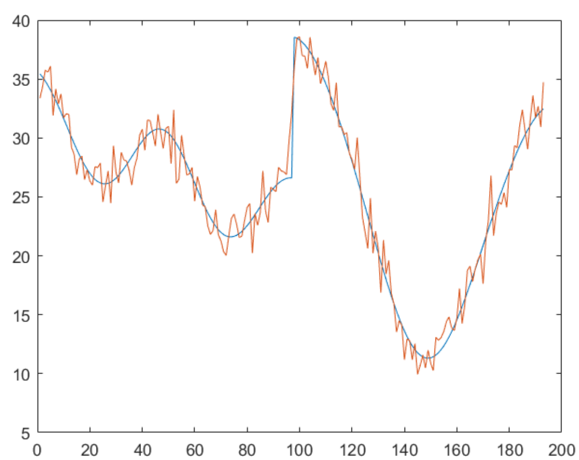
Question:

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 $x_1[n]$.
2. First sharpen (deblur) and then remove noise. Let the resulting signal be $x_2[n]$.

Now, compare $x_1[n]$ and $x_2[n]$ with $x[n]$. What conclusions can you draw from your observations? Also, explain your observations from a theoretical perspective if possible.

Here is a plot of the original and the distorted signal:



Theory:

- **Denoising the signal** :

The concept for removing noise is averaging the signal with the neighbouring elements.
The solution includes the concept that the mean of the noise over the entire interval tends to 0.
This needs to be done in the following way, separately for edge elements.

Examples:
y[2] will be replaced by (y[1] + y[2] + y[3])/3

y[0] will be replaced by (y[1] + y[0] + y[1])/3

(For edge elements we can take mirror values for absent values)

This will perform the task of De-noising the distorted signal.

- **Deblurring the signal**:

The original signal is getting convoluted with impulse response h[n] (i.e $\frac{1}{16}$[1 4 6 4 1]) & h[0] = $\frac{6}{16}$)

So,

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

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 $\Omega = \frac{2\pi k}{N}$.

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

Multiplicative property:

$$\frac{1}{2\pi}\int_{2\pi} X(e^{j\theta})Y(e^{j(\omega-\theta)})d\theta$$

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

$$X(e^{j\Omega}) = \sum_{n=-\infty}^{\infty} x[n]e^{-j\Omega n},$$

Substituting $\Omega = \frac{2\pi k}{N}$ in the above equation we get,

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

where h[n] ≠0, n ∈[-2,2] and y[n]≠0, n ∈[0,192]; n ∈Z

By Discrete Time Fourier Transform,

$$x[n] = \frac{1}{2\pi}\int_{2\pi} X(e^{j\Omega})e^{j\Omega n}d\Omega$$

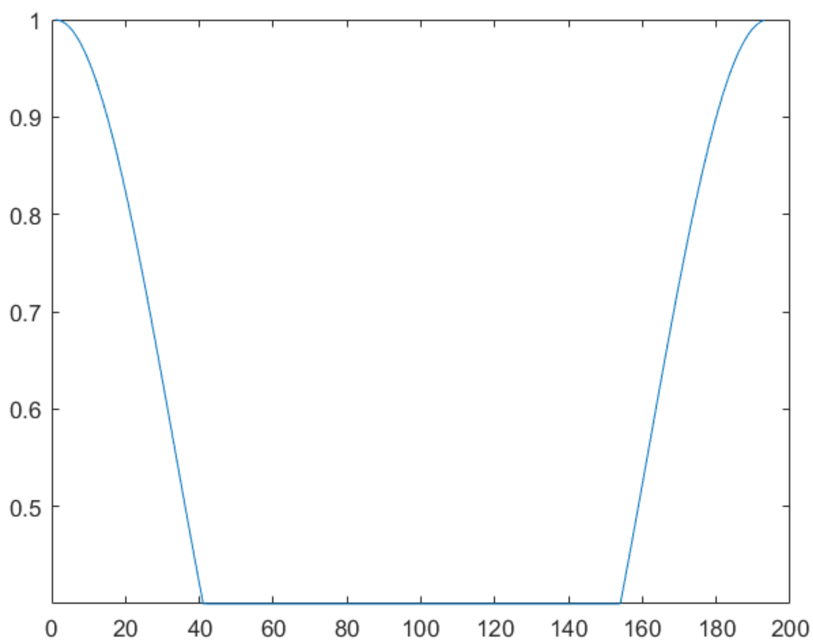
Substituting $\Omega = \frac{2\pi k}{N}$ in the above equation we get,

$$x[n] = \frac{1}{N}\int_{N} X(e^{j\frac{2\pi k}{N}})e^{j\frac{2\pi k}{N}n}dk \;\; -> \;\; x[n] = \frac{1}{N}\sum_{k=0}^{N} X(e^{j\frac{2\pi k}{N}})e^{j\frac{2\pi k}{N}n}$$

(Integration was performed as a Reimann sum but mathematical manipulation was done to optimise the code as integrated values were not needed. Hence, the number of terms in Reimann sum were not increased more than 193 as it was to be sampled in 193 parts anyway)

Frequencies will range from $\frac{2\pi(0)}{193}$ to $\frac{2\pi(192)}{193}$.

The factor 1/ H[k] which will be multiplied to every term needs to be low to act like a low pass filter and doesn't contribute to further distortion of the signal. The signal here is filtered using the value of 0.4 which gives us the optimum output.

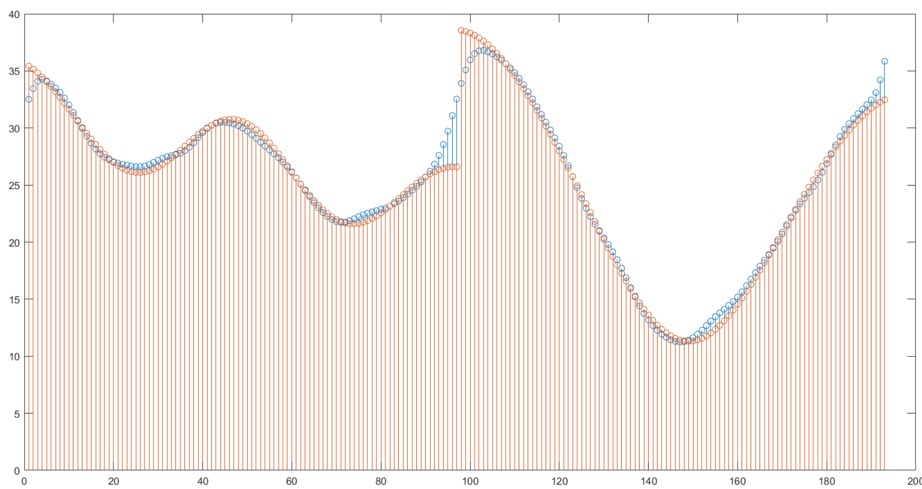


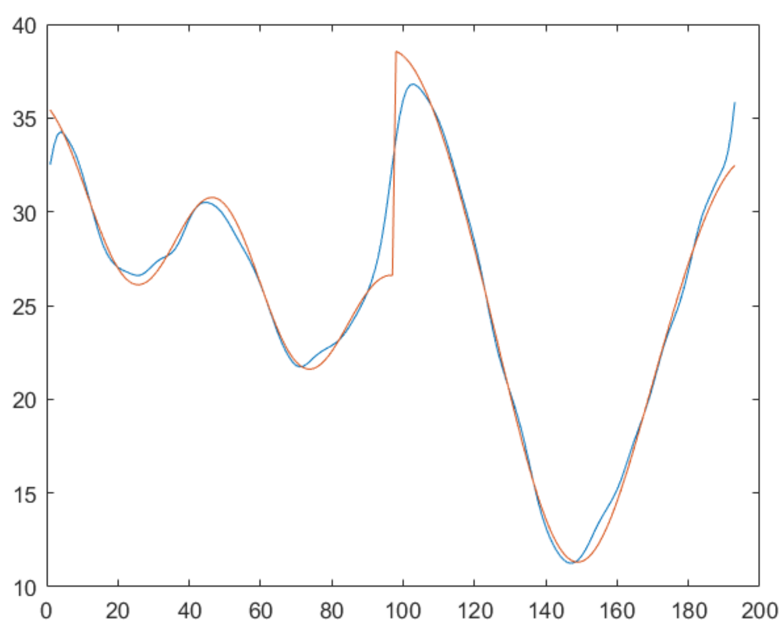
Using the above filter and taking the inverse of the output, the signal is De-blurred.

Note that, De-noising and De-blurring do not go hand in hand. Removing the one and introducing the other respectively when they are performed.
Thus to manipulate the signal to its original state, the number of times each is performed needs to be taken care of while de-noising and de-blurring the signal.

Results:

1. **X1[n]**
   The following output was obtained from executing denoise function before deblurring the data:

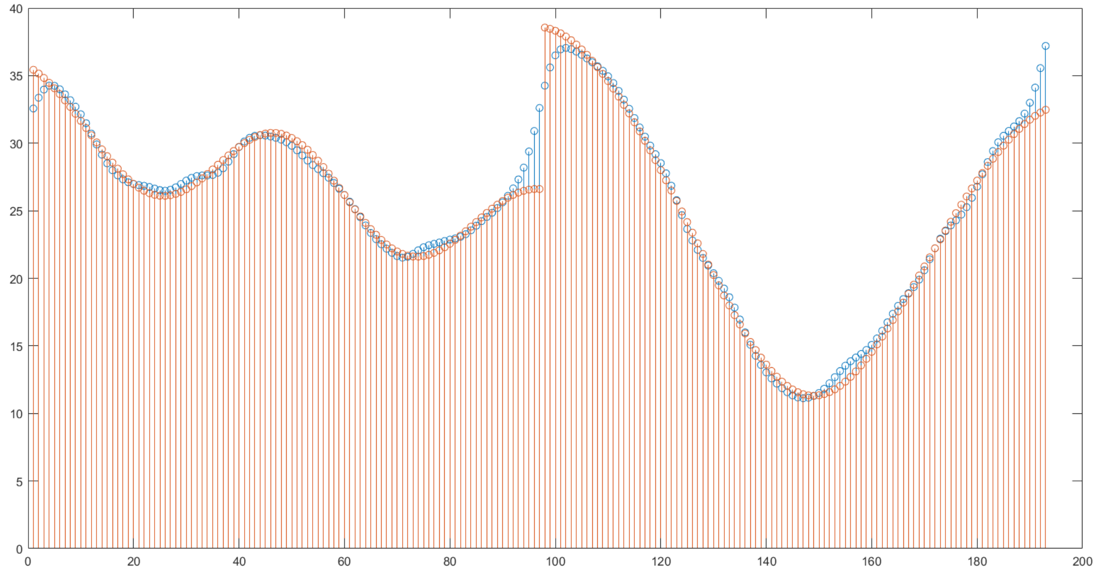


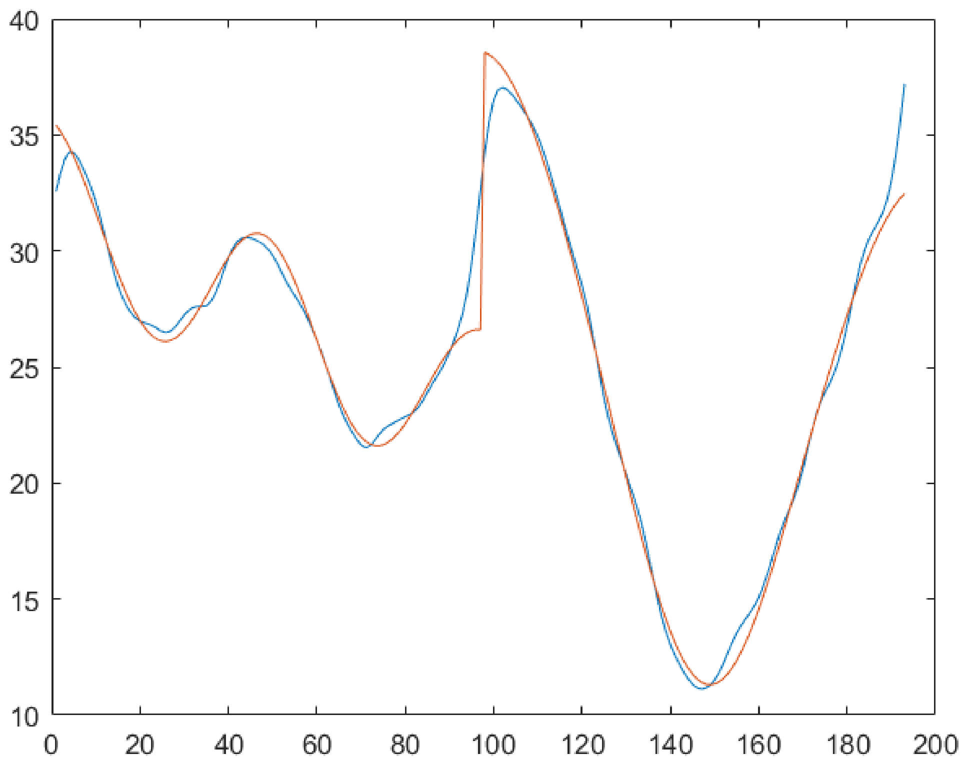


The mean squared error of the above graph comes out to be 0.8974.

2. **X2[n]:**

The following output was obtained from deblurring function before denoising the data:





The mean squared error of the above graph comes out to be 0.9620.

Conclusion:

The method used to recover the signal x1[n] is more efficient than the method applied for x2[n].
This can be seen in the mean square error of the recovered signals with the original signal. Where:
M.S.E of x1[n]= 0.8974 , whereas
M.S.E of x2[n]= 0.9620

A plausible theoretical explanation to this can be inferred from the function which distorted the signal,which is:

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

As can be seen in the above equation, blurring is imparted to the original signal, then noise is added to it.
To rectify the same and following the reverse order, we need to denoise the signal first followed by sharpening, as it is done in x1[n].
The second method which gives rise to the signal x2[n], is done by first sharpening the signal, which affects the noise present in it as well. Hence hindering the recovery of x[n].

Therefore, x1[n] is closer to x[n] rather than x2[n], which can be seen in the mean square error of both the plots.

Contributions:

Aaditya Baranwal(B20EE001):

- [ ] DTFT implementation
- [ ] De-blurring
- [ ] Commenting
- [ ] Filtering and Inverse DTFT implementation
- [ ] Debugging
- [ ] ReadMe file
- [ ] Theoretical Planning

Haardik Ravat(B20EE021):

- [ ] De-noising
- [ ] DTFT implementation
- [ ] De-blurring
- [ ] Commenting
- [ ] Debugging
- [ ] Theoretical Planning
- [ ] Report Writing

Citations:

Matworks: https://in.mathworks.com/
EEL2010: https://sites.google.com/iitj.ac.in/eel2010/