**SIGNALS AND SYSTEMS ASSIGNMENT**

**Introduction**

A signal x[n] was given which consisted of the original temperature values that the sensor recorded from the surroundings. This x[n] was stored in the base unit but the stored signal consisted of noise and blurriness, which was supposed to be removed to get back a signal which could be closest to the original signal x[n]. A h[n] was given due to which the original signal got blurred. Question 1 asked to first denoise the signal then deblur it to get a predicted x[n]. Question 2 asked to first deblur the signal then denoise it to again get a predicted x[n].On comparing the results of the two Questions we can get the best way to purify y[n] .

In Figure-a 1 we can see that a noisy and blurry signal y[n] is plotted and in Figure-b1 the original x[n] is plotted.

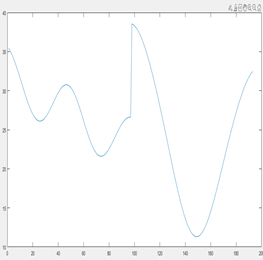
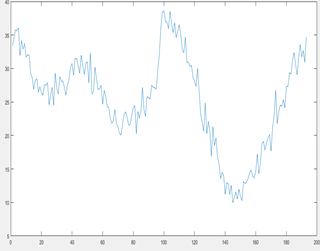
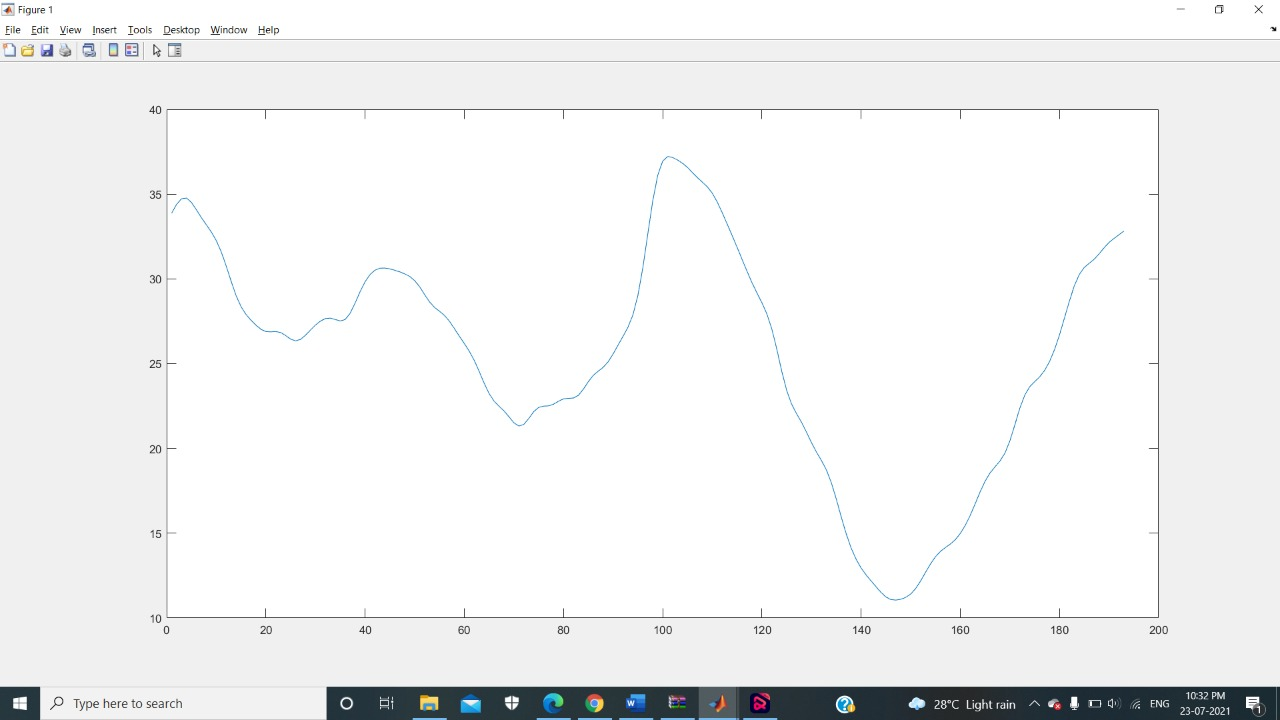


Figure-a 1 Figure b

**Denoised then Deblurred -x1[n]**

For the first question the noise is to be removed first and then the blurriness. Now to remove the noise we basically averaged the signal y[n] and iterated to get a smoother signal in every loop. For averaging we took a total of three values. For every value of y[n] we took the average of values from one index more to one index less than the sample.This process is based on the logic that the mean of noise over the entire interval of the signal is negligible.

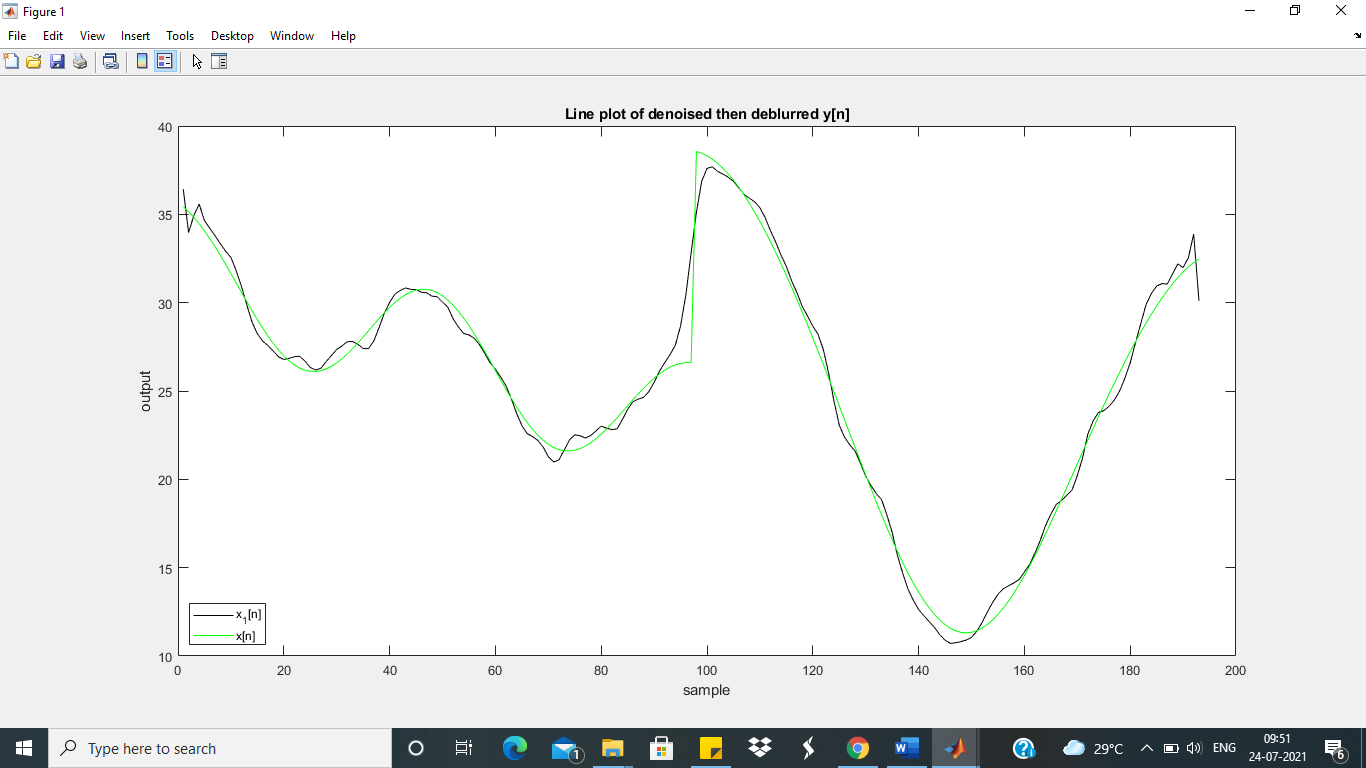
For every iteration noise was minimized in such a way that MSE (Mean squared error) can be minimized.The signal obtained after denoising is named as ‘y\_denoised’(Figure C-1).

 Figure C-1

Given the h[n],y[n]=h[n] X x[n] (where X stands for the convolution symbol).Since it is difficult to compute convolution of signals with raw data and no mathematical formula, we may use the Convolution Property of the Fourier Transform which states that Fourier transform of Convolution in Time domain is multiplication of fourier transform of each signal in frequency domain.So, Y(e^jw)=H(e^jw)\*X(e^jw).

Given the h[n]=1/16\*[1 4 6 4 1], it can be converted into dirac delta function as 1/16\*(δ[n+2]+δ[n+1]+δ[n]+δ[n-1]+δ[n-2]). Now for dirac delta function fourier transform is 1 and using the shifting property in time domain we get H(jw) on solving finally as (1/8)\*(4\*cos((k-1)\*wo)+cos(2\*(k-1)\*wo)+3), where k is variable which runs in a loop from 0 to 193 and wo=2\*pi/193.Now to calculate the Discrete Fourier Transform of the y[n] the conventional formula is implemented using two loops both ranging from 1:193 with wo=2\*pi/193. Let Y(jw)=Y and H(jw)=H.Now since some values of H might be zero and 1/H can reach very large values.To avoid that we perform thresholding in which values of H<0.3 will be replaced by 0.3.This thresholding coefficient is chosen by hit and trial method.

Now since Y=H\*(Xj) we can calculate the Xj by dividing Y by H where Xj will be our predicted Xjw values.By one to one division of each element of Yj by elements of Hj.So finally the predicted output Xj has size of array as 193x1.Now to get the output in time domain from frequency domain we need to perform Inverse Discrete Fourier Transform on Xj to get x\_denoise\_deblurr[n].The inverse discrete fourier transform is implemented using the conventional formula by using two loops of range 1:193 giving a complex x[n]. Since the complex part of x[n] is negligible we can plot the real values of x[n] to get the required output.

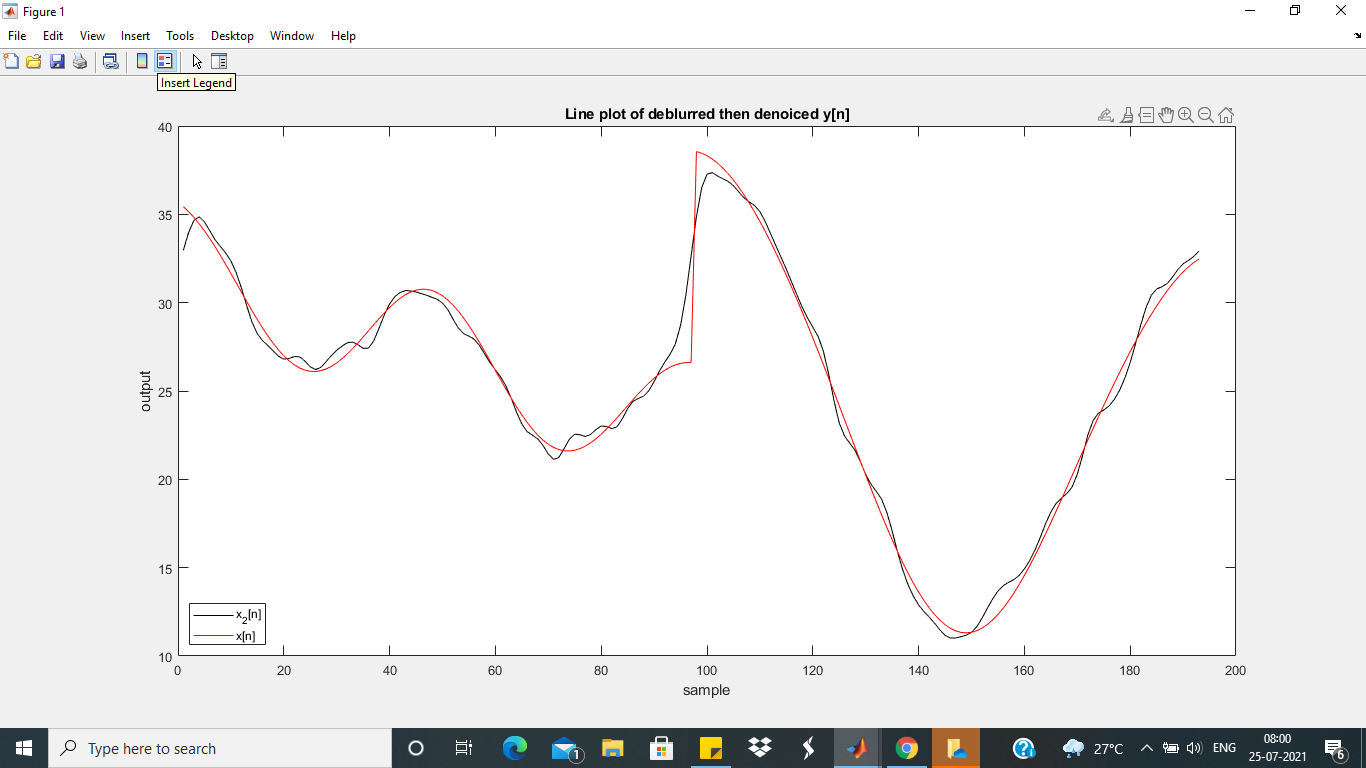


*Figure-d-1-black plot represents x1[n] and green represents x original.*

**Deblurred then Denoised-x2[n]**

For the second question the blurriness is removed first and then the noise, now to remove blurriness we used the multiplication property of the fourier transform in frequency domain just like in x1[n] after finding the DTFT of Yj and after using dirac delta function and shifting property in time domain to get impulse response H(jw) and then again convert it in time domain by finding inverse fourier transform of Yj/H i.e., Xj and by doing that we found deblurred signal .But the deblurred signal x\_deblurred[n] still contains noise which is yet to be removed.After that to find denoised signal we averaged the signal x\_deblurred[n] like we did in earlier case of y[n], Hence we removed the noise in such a manner that MSE (Mean squared error) is minimum

After these operations we have got a deblurred denoised signal.



*Figure-e-1-black plot represents x2[n] and green represents x original.*

**Conclusion**

Mean squared error(MSE) of x1[n] is 0.684 whereas that of x2[n] is 0.7080.Since MSE of x1[n] is less than MSE of x2[n] we may conclude that x1[n] is better and closer representation of original signal than x2[n].

A possible explanation for this conclusion could be that while calculating x2[n] we are actually deblurring/sharpening the signal containing noise due to which the noise is also sharpened and applying the algorithm of denoising on sharpened/amplified noise is not as effective as it is on the initial signal with unamplified noise(I.e y[n]).

**Citations:**

* [**https://in.mathworks.com/**](https://in.mathworks.com/)
* Course Instructor’s video lectures

**Contribution**

* **Devyani Gorkar(B20ME027):**Denoising and Deblurring code and theoretical planning, DTFT implementation,inverse DTFT implementation code, thresholding,report writing:
* **Mruganshi Gohel(B20CS014):**Theoretical planning of deblurring, DTFT implementation, report writing,commenting in the code