Theoretical explanation

1. **Denoising : - Denoising is an important process to restore the original signals from the noisy ones. Weighted Average algorithm using kernel and reflection approach is a simple and effective denoising algorithm .**
   The principle of most denoising methods is quite simple: Replace the color of a pixel with an average of the nearby pixels .

   If X_i are i.i.d of standard deviation σ

   $$Var\left(\frac{X_1 + \cdots + X_m}{m}\right) = \frac{\sigma^2}{m}$$

   The average reduces the uncertainty by m.

   Using this principle we have done denoising . Can be seen in denoised_signal.m **.**
2. **DTFT :- The Discrete-Time Fourier transform of a discrete sequence of real or complex numbers *x*[*n*], for all integers *n*, is a Fourier series, which produces a periodic function of a frequency variable. When the frequency variable, ω, has normalized units of radians/sample, the periodicity is 2π.**

   $$X_{2\pi}(\omega) = \sum_{n=-\infty}^{\infty} x[n]\, e^{-i\omega n}.$$

   We have used this formula after some modifications in a more generalized way , which were required to run in MATLAB . can be seen in DTFT.m **function code .**
3. **IFT :- Inverse Fourier Transform. An operation that recovers the discrete data sequence from the DTFT function is called an inverse DTFT.**

   $$x[n] = \frac{1}{2\pi} \int_{2\pi} X_{2\pi}(\omega) \cdot e^{i\omega n} d\omega \quad \text{(integral over any interval of length } 2\pi\text{)}$$

   Using this formula we do IFT and can recover data from DTFT of signal .
   We have also used formula in code IFT.m **function in such a way that it could run in MATLAB .**
4. **Deblur :- Deblurring is the process of removing blurring effects from signal. Deblurring recovers a sharp signal *s* from a blurred signal *b*, where *s* is convolved with *k* (the blur kernel) to generate *b*. Mathematically, this can be represented as**

   b=s*k (where * represents convolution)

   Sharpening is done by taking DTFT of following formula
   And it becomes

   => B(i)=S(i) x K(i)
   => S(i)=B(i) / S(i)

   From this we obtain DTFT of sharp signal *s* that is S(i) , now we take IFT of S(i) to to get *s* sharp signal .
   This way we deblurred the signal .

Explanation of Q1 (First remove noise and then sharpen (deblur).)

1. As asked in implementation we first remove noise from then signal y ,by using the weighted average algorithm ,it result in formation of following signal

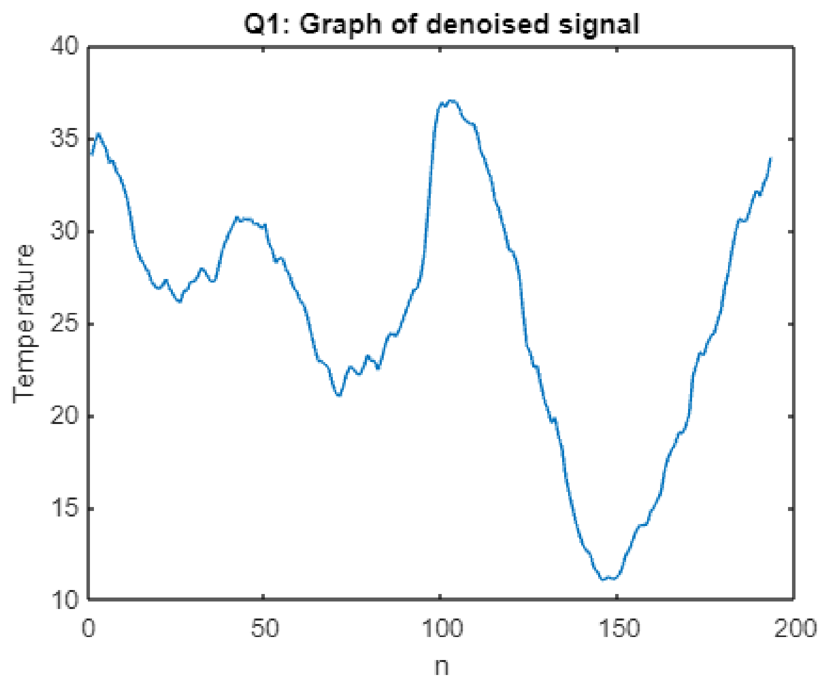

Fig.1

Then next we do deblurring by taking DTFT of signal(Fig.1) obtained from denoising , also we have to take DTFT of h[n] which results in H(i) .
Then we split ω in 0 to 2pi into 193 parts by linear spaced vector (linspace function), so that H(i) becomes an array of 1x193 . Also DTFT of denoised signal is equal to Y(i) that is also an array of 1x193 . Dividing element wise Y(i) by H(i) gives us deblurred signal in frequency domain , taking IFT of that signal will give us final signal $x_1[n]$.

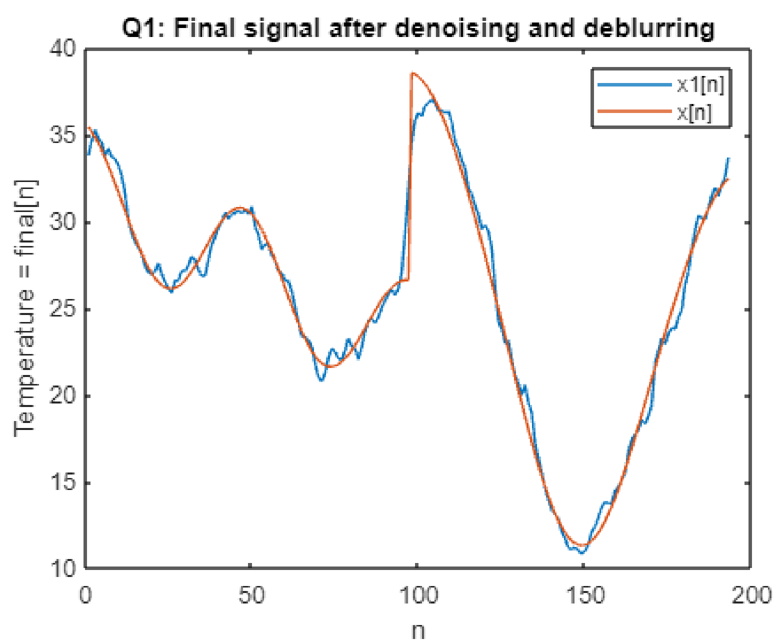

Fig. 2 Graph of $x_1[n]$

Explanation of Q2 (First sharpen (deblur) and then remove noise.)

1. **So here we do deblurring first as asked , now we take DTFT of signal y[n] and h[n] results in Y(i) and H(i) respectively .Also we will extend H(i) from 0 to 2pi into 193 parts using linearly spaced vector . Then we will divide Y(i) by H(i) ,this will give us deblurred signal in frequency domain , taking IFT of which will give us deblurred signal (Fig.3)**

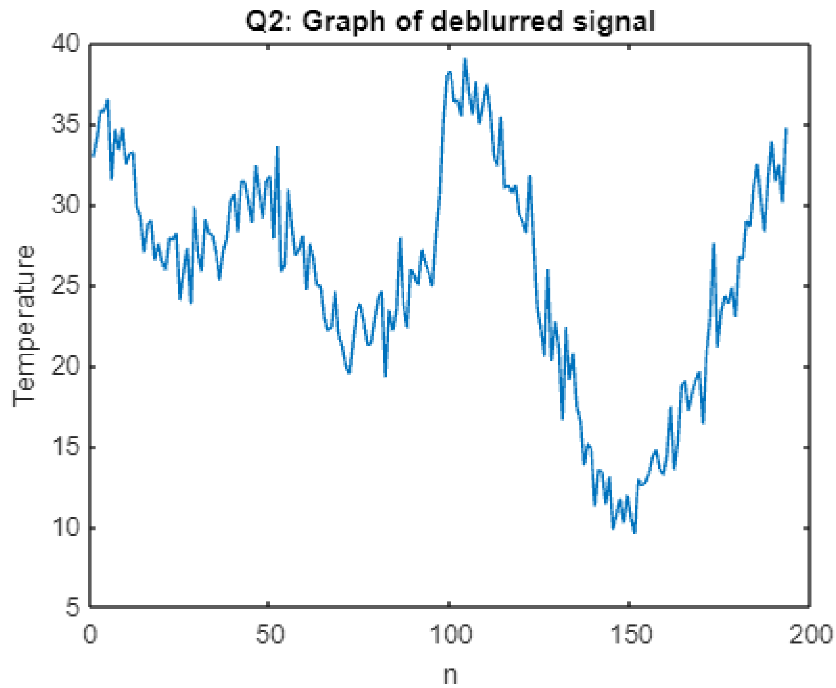

Fig.3 Deblurred signal

Now we will denoise the deblurred signal ,by using the weighted average algorithm ,it result in formation of final signal x_2[n] (Fig.4)

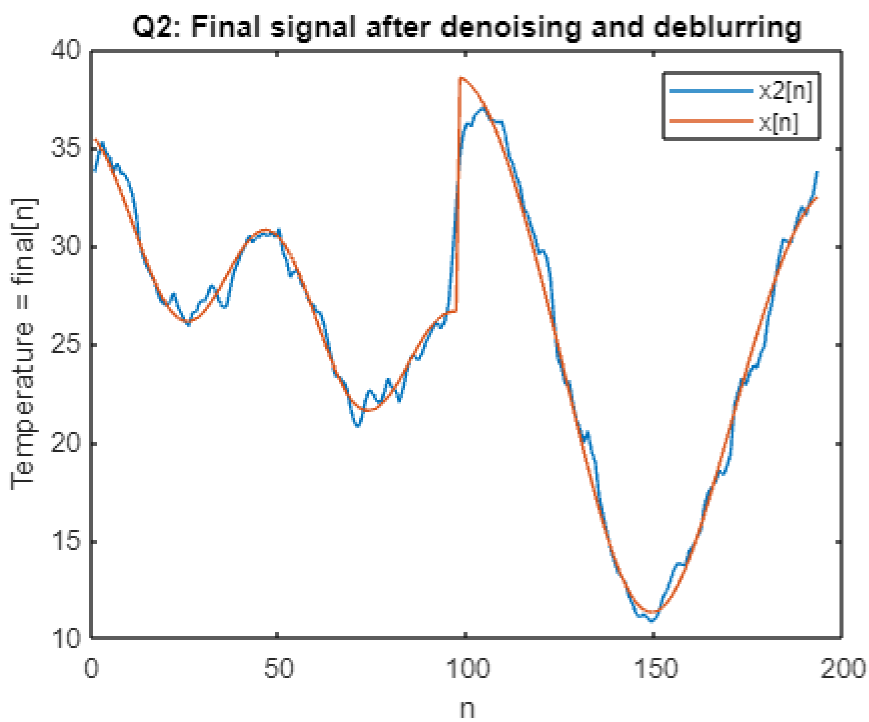

Fig .4 Denoised signal after deblurring

Final Conclusion:

1. **In $x_2[n]$ as we deblurred earlier and then denoised so the noise would be greater in $x_2[n]$ in comparison to $x_1[n]$.**

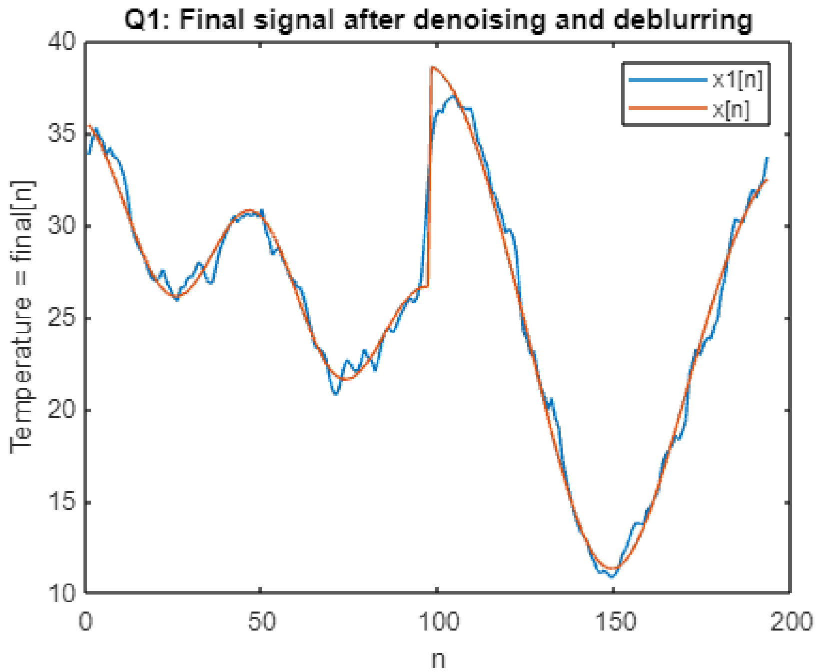

Fig.5 Comparison of $x_1[n]$ and $x[n]$

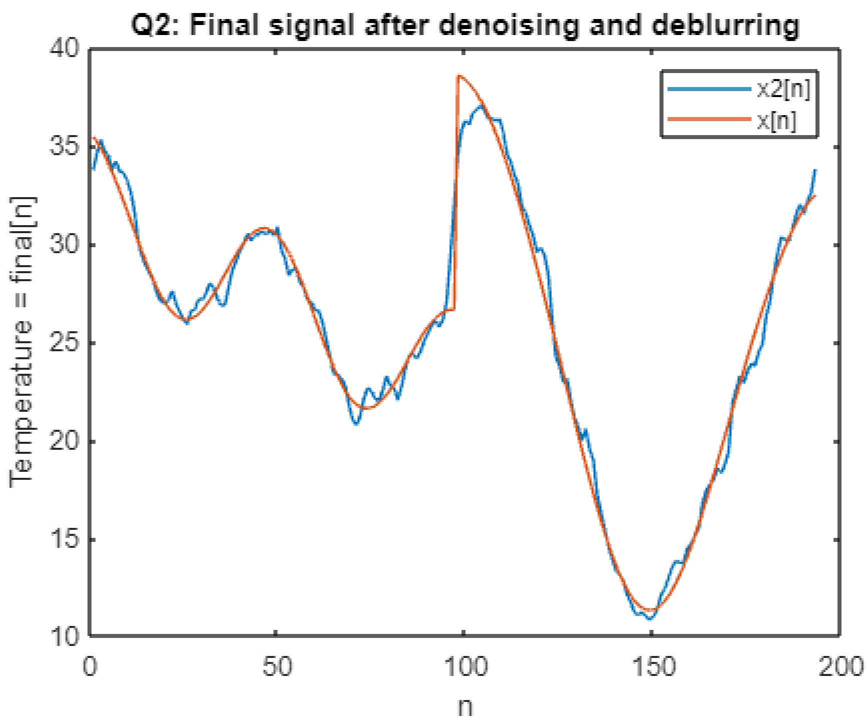

Fig .6 Comparison of $x_2[n]$ and $x[n]$

2. **We can deduce that in implementation 1 after deblurring , there is generation of some noise again which could be seen in Fig.5 .**

3. **Also we can see that in implementation 2 denoising after deblurring , there is some loss of sharpening which could be seen in Fig.6 .**

Important Observation we noticed:

1. **standard deviation of $x_1[n]$and x = 6.166269 and standard deviation of $x_2[n]$and x.= 6.166116**
2. **The standard deviation $x_1[n]$ and $x_2[n]$ are approximately equal .**
3. **x1 is more better than x2 as it is having less noise.**

Limitations:

1. **We have taken N=193 which is large enough but still not equal to infinity and so more error is introduced due to it.**
2. **In denoising , we are appliying kernel method of weighted everything , along with relection approach , but then also some error arises.**

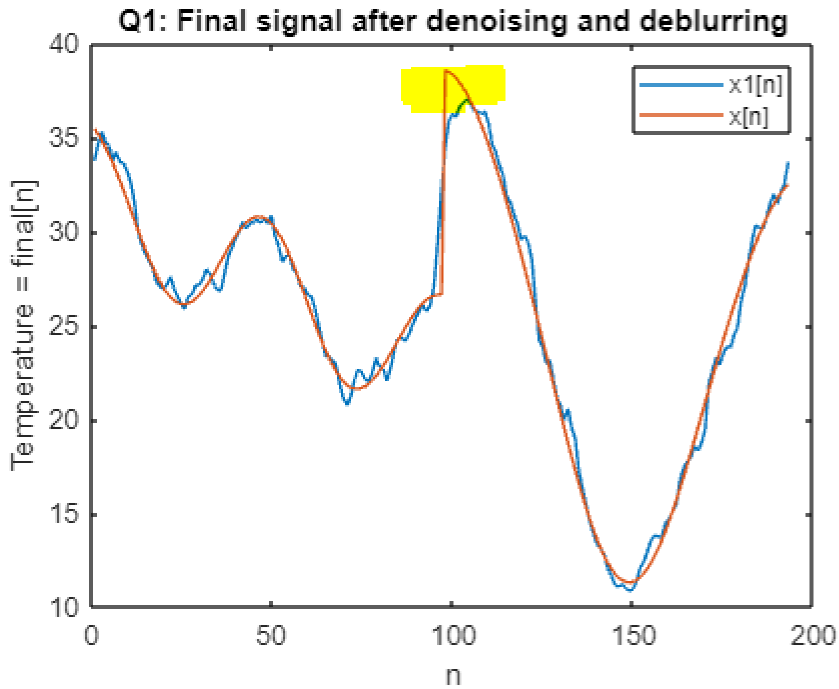

Fig . 7 (some error in highlighted part)