EC60064
Biomedical System Engineering and Automation

Experiment-1
Report

Prepared by

Sai Joshitha Annareddy
21EE62R10
MTech
Control systems engineering

Question 1:

Problem Statement:

The data file ecg2x60.dat contains ECG signal, sampled at 200Hz with a powerline interference noise of 60 Hz.

a. **Design a notch filter with two zeros to remove the artifact and implement it in Python.**

Theory:

Notch Filter Design:

A notch filter is a type of band-stop filter, which is a filter that attenuates frequencies within a specific range while passing all other frequencies unaltered. For a notch filter, this range of frequencies is very narrow.

In biomedical signal processing Notch filters are used to remove a single frequency ie., power line interference frequency which may be 50hz or 60 hz. This is the most commonly encountered periodic artifact in biomedical signals. So, in order to remove the specific frequency notch filter is designed in the following way.

The power line frequency (fo) = 60hz

Sampling frequency (fs) = 200 hz

Angle of zeros $= \frac{2\pi(fo)}{fs}$

$= \frac{2\pi(60)}{200}$

$= 108°$

The two zeros z1, z2 are

Cos (108) +j*sin (108), cos (108)-j*sin (108)

[0.32420137+0.94598809j 0.32420137-0.94598809j]

Results and Observations:

Notch filter is designed using python in Jupyter notebook and the below is the magnitude and phase response of Notch filter.

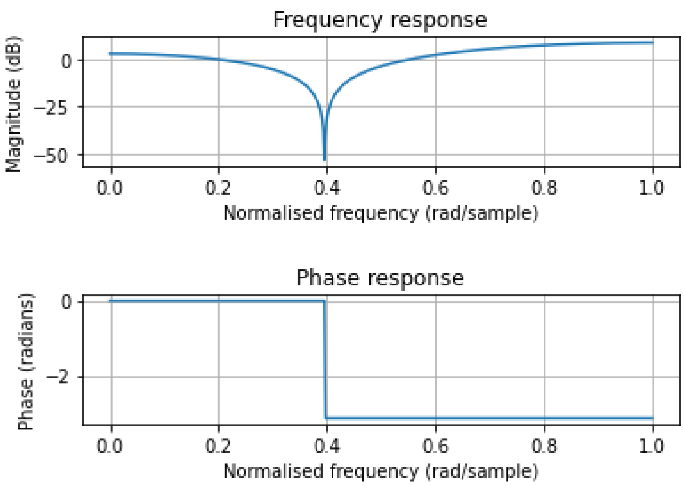

Figure 1a: Notch filter with only zeros

When the above designed notch filter is passed through given noisy ECG signal (ecg2x60.dat) to remove power line interference artifact.

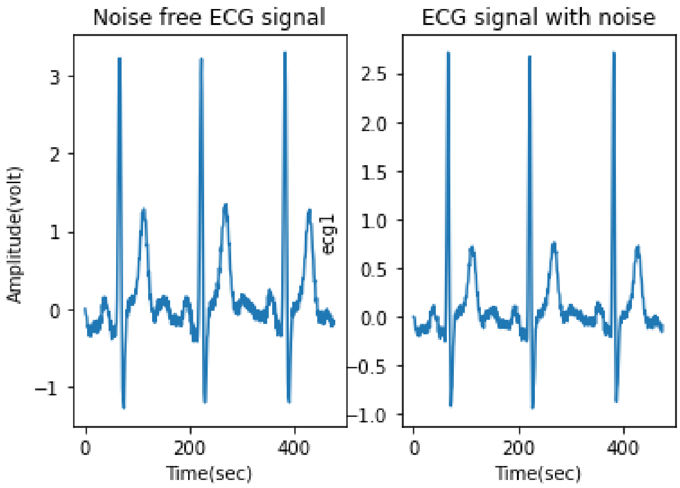

Figure 1 b: comparison of filtered signal with notch filter and noisy ECG signal

b. Add two poles at the same frequency as those of zeros, but with a radius 0.99. Study the effect of poles on the output of the filter as their radius is varied from 0.95 to 0.75.

Theory:

Including poles at the same frequency as those of zeros in design of notch filter with radius 0.99.

The two zeros z1, z2 are

[0.32420137+0.94598809j 0.32420137-0.94598809j]

The two poles p1, p2 are

r*[0.32420137+0.94598809j 0.32420137-0.94598809j]

where…

r = 0.99

Results and Observations:

Notch filter using poles and r=0.99 is designed using python in Jupyter notebook and the below is the magnitude and phase response of Notch filter.

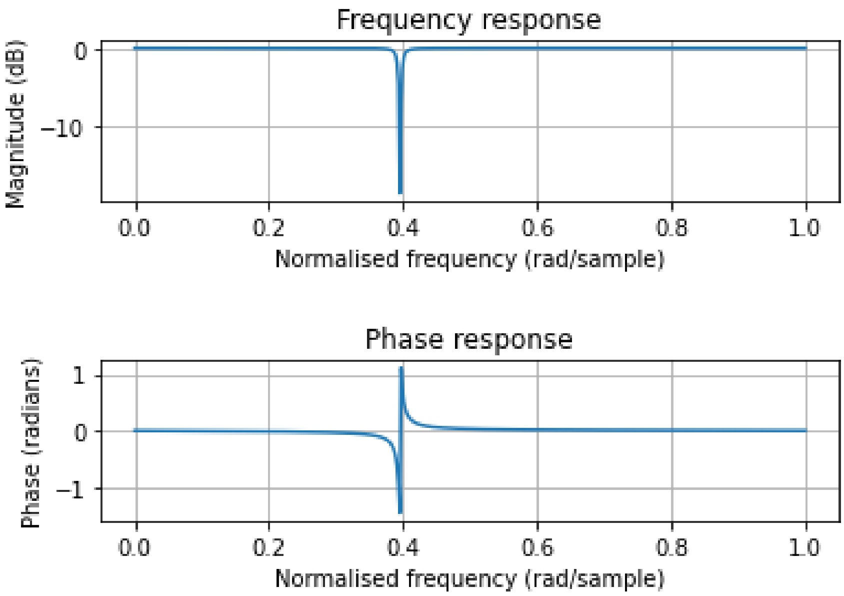

Figure 1c: Notch filter with zeros and poles and r=0.99

When the above designed notch filter with poles and r=0.99 is passed through given noisy ECG signal (ecg2x60.dat) to remove power line interference artifact.

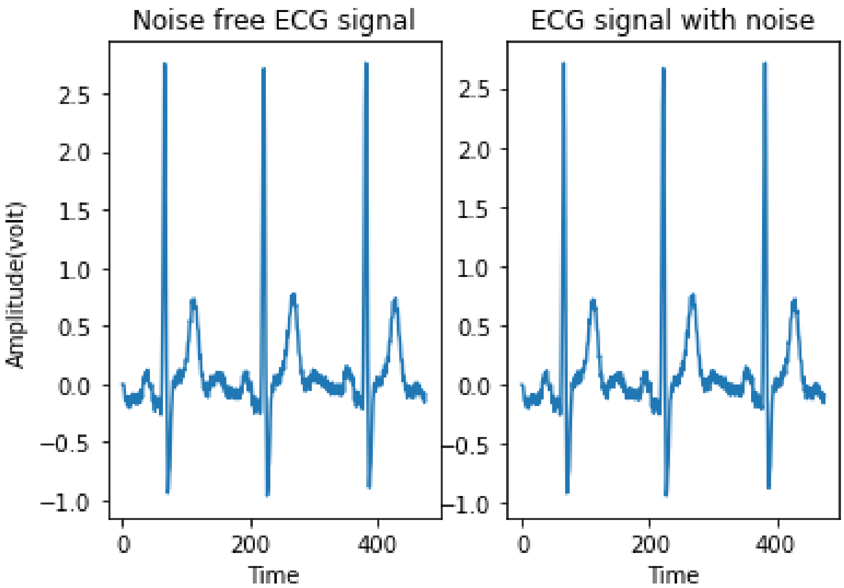

Figure 1d: comparison of filtered signal with notch filter (poles, r=0.99 and zeros) and noisy ECG signal

Now varying the radius r for poles and plotting the filter characteristics,

r is varied from 0.75 to 0.95

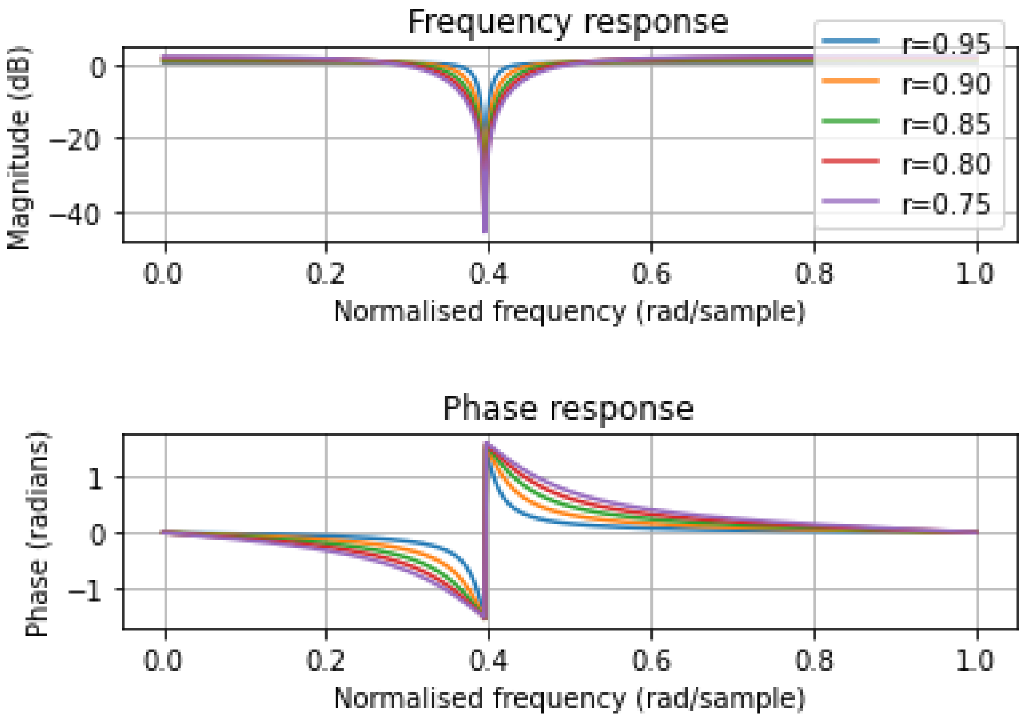

Figure 1e: Notch filter with various r values for poles

Conclusion and Discussion:

From the above results

The noise is removed if the notch filter with only zeros is designed and passed through noisy ECG signal. When poles are added at the same frequency as that of zeros and r=0.99 then noise is less removed compared to the filter with only zeros. (From Figure 1b and 1d)

The notch filter with only zeros is more accurate because we have linear phase whereas when poles are added phase is getting disturbed. (From Figure 1a and 1c)

We can also observe that as r increase the notch filter width decreases that is it becomes narrower. (From Figure 1e)

Question 2:

Problem Statement:

A noisy ECG signal (sampled at 1000Hz) is provided in the file ecg_hfn.dat. Develop a Python code to perform Synchronized averaging. Select a QRS complex from the noisy signal for use as the template and use a suitable threshold on cross-correlation function for beat detection.

a. **Comment on the effect on beat detection if the threshold is very low (0.3) or very high (0.95).**

Theory:

In order to obtain trigger points, a sample QRS complex (715 samples at a sampling rate of 1,000 Hz) was extracted from the first beat in the signal and used as a template. Template matching was performed using a normalized correlation coefficient defined as

$$\gamma_{xy}(k) = \frac{\sum_{n=0}^{N-1}[x(n)-\bar{x}][y(n-k)-\bar{y}]}{\sqrt{\sum_{n=0}^{N-1}[x(n)-\bar{x}]^2 \sum_{n=0}^{N-1}[y(n-k)-\bar{y}]^2}}$$

Where…

x is a template and y is ECG signal, N samples and k is the time index of the signal y at which the template is placed.

Results and observations:

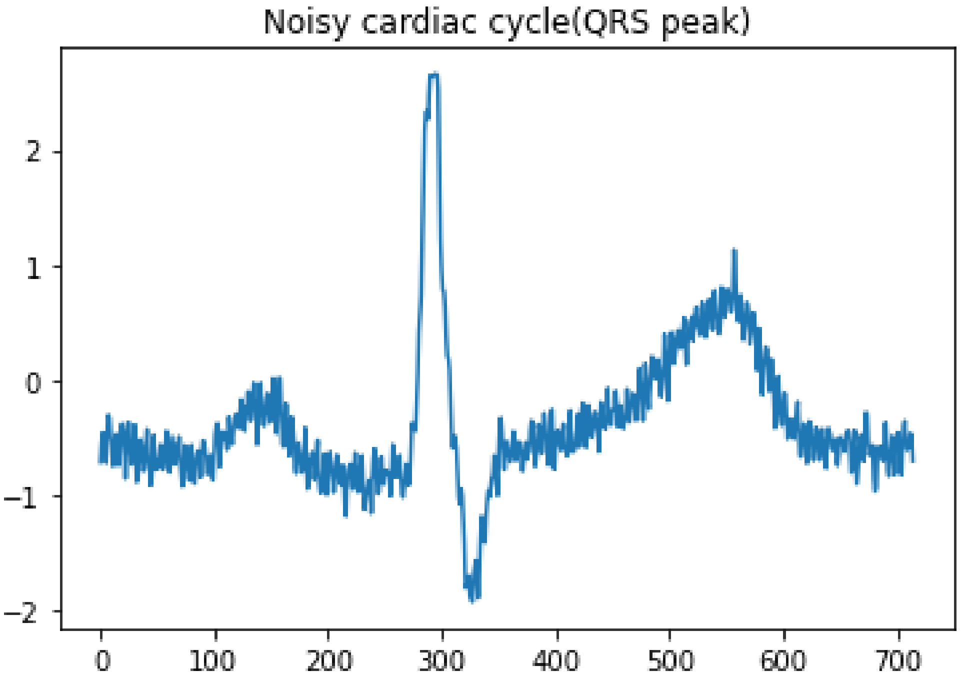

Figure 2a: Sample QRS complex

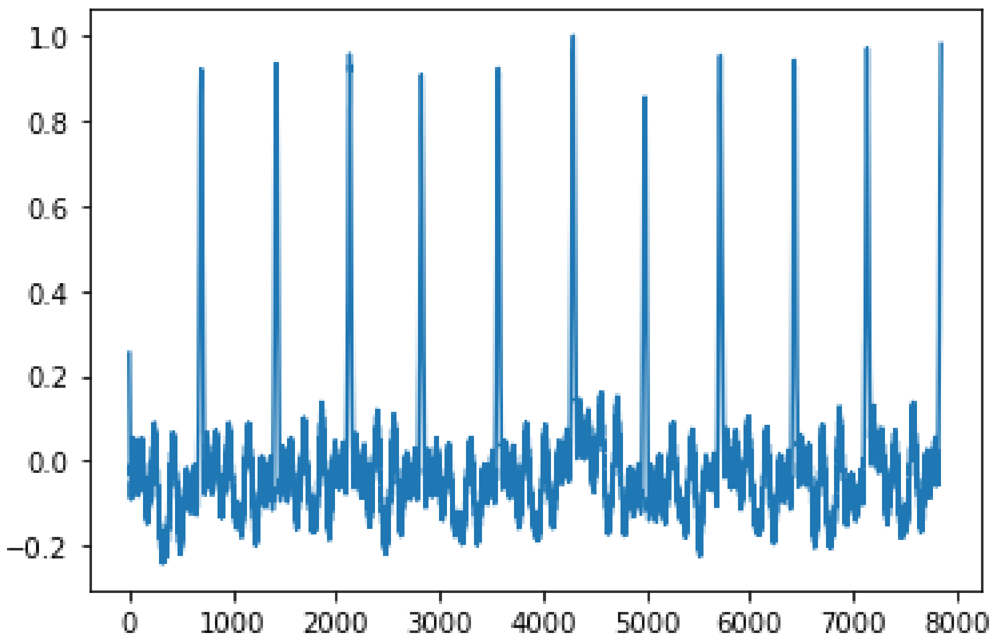

Figure 2b: Cross correlation of template with ECG noise signal

By choosing an appropriate threshold, it becomes possible to obtain a trigger point to extract the QRS complex locations in the ECG signal.

Conclusions and discussions:

If the threshold of 0.953 is applied, then 12 beats are detected and if threshold 0.3 is applied then 406 beats are detected which are undesirable.

b. Plot the result of an averaged QRS complex. Ensure that the averaged result covers one full cardiac cycle.

Theory

Synchronized averaging filter is time domain filtering approach. The signal consists of periodic parts that are separated sum up and then divided by total no of parts.
The averaging process will gradually eliminate the random noise.

An algorithmic description of synchronized averaging is as follows:
1. Obtain a number of realizations of the signal or event of interest.
2. Determine a reference point for each realization of the signal.
3. Extract parts of the signal corresponding to the events and add them to a buffer.
4. Divide the result in the buffer by the number of events added

Results and observations

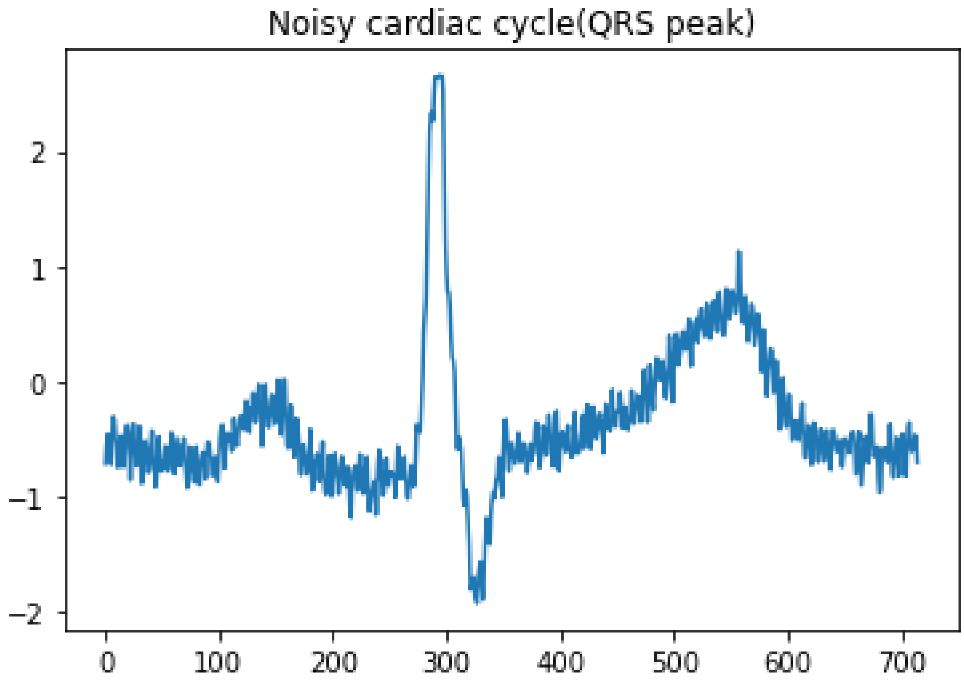

Figure 2c: Noisy cardiac cycle (QRS peak)

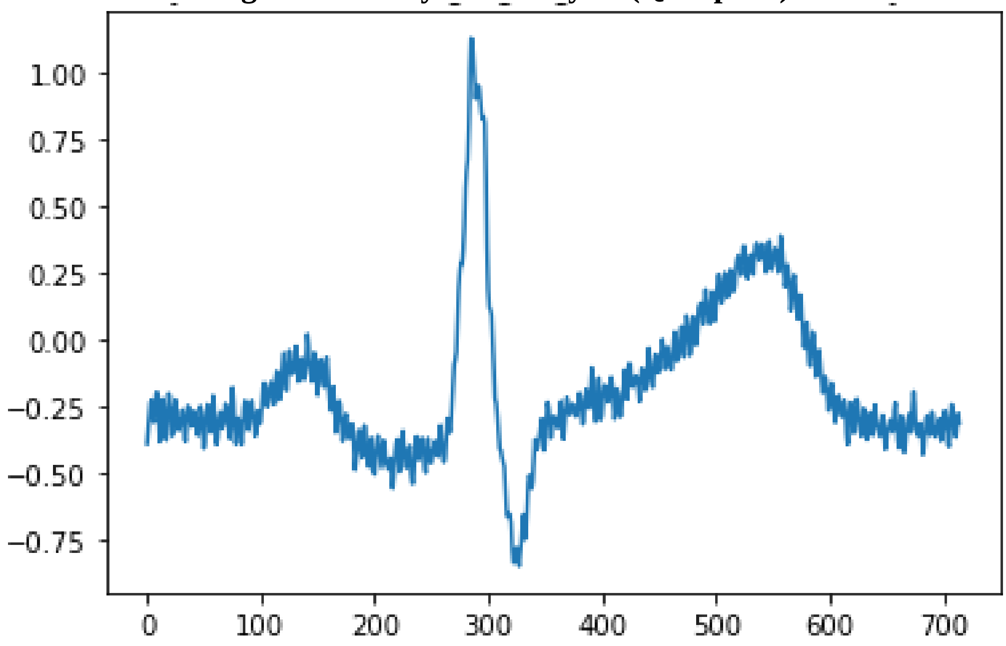

Figure 2d: Averaging 2 synchronous signals

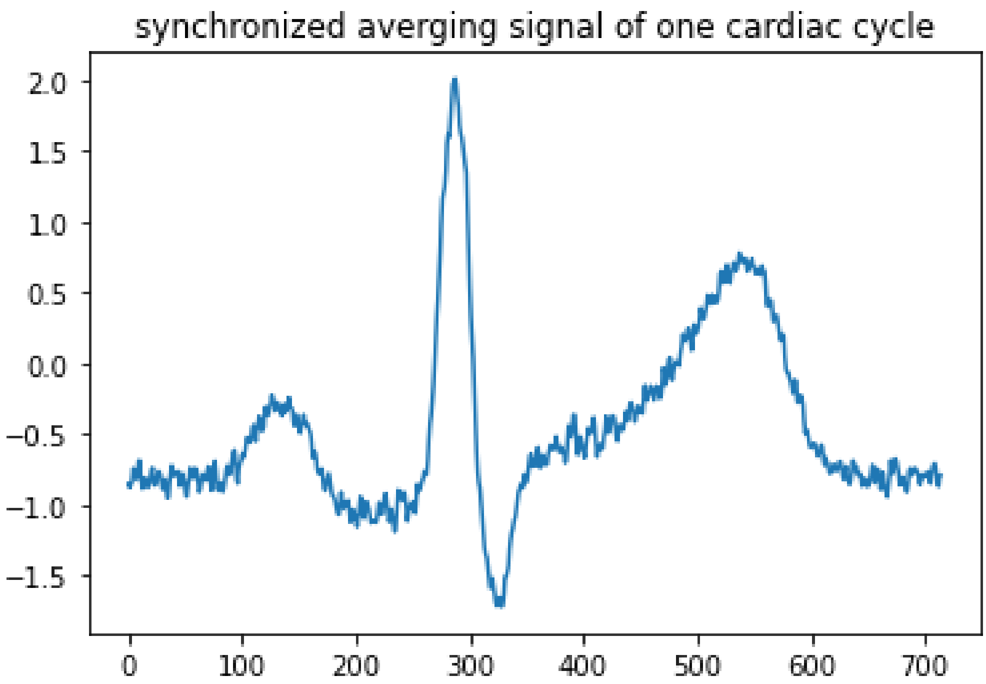

Figure 2e: Synchronized averaging signal of one cardiac cycle

Conclusions and discussions:

From the above results
we can observe that single QRS cycle is noisy(from figure 2c) and the noise got reduced when we average two signals in time domain(from figure 2d) so further averaging 12 parts of the signal gives synchronized averaging signals of one cardiac cycle(from figure 2e)

c. **Plot a sample ECG cycle each from noisy and original signals for comparison**

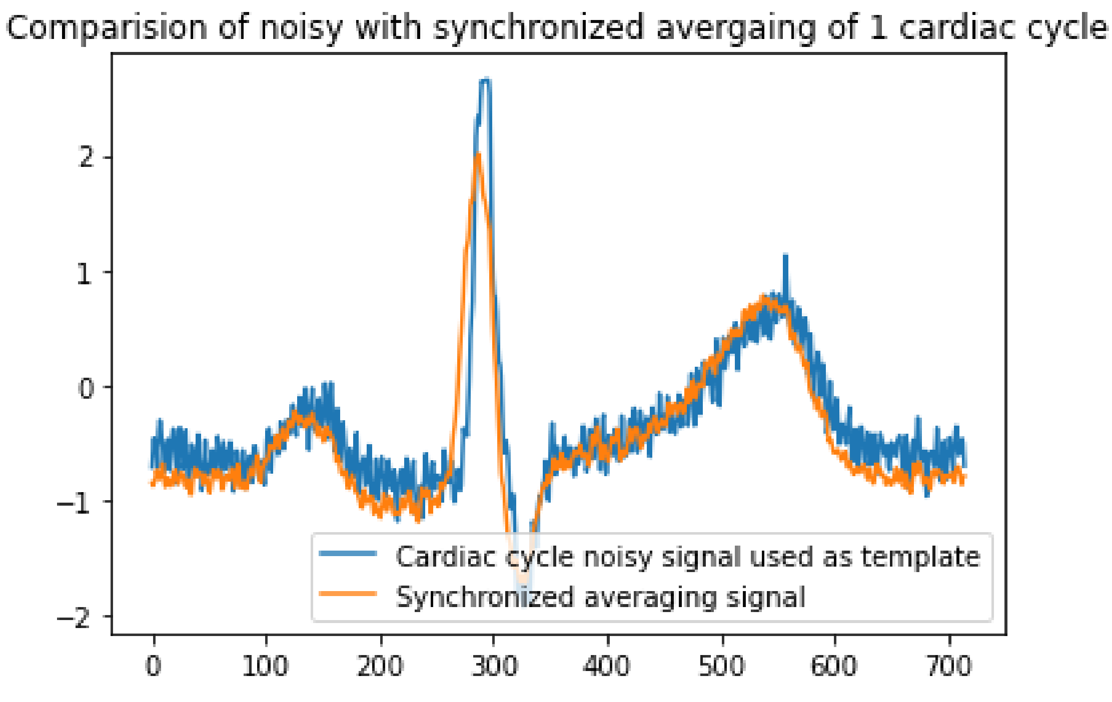

Figure 2f: Comparison of noisy ECG signal template with synchronized averaging signal of 1 cardiac cycle

Conclusions and discussions:

We can observe (from figure 2f) that synchronized averaging signal has less noise compared to signal which is used as template.

Question 3:

Problem Statement:

Filter the noisy ECG signal in ecg_hfn.dat. using four different Butterworth low pass filters (individually) realized through Python with the following characteristics:

a. **Compare the results using each of the four Butterworth filters (individually) with those obtained by synchronized averaging, and comment on the improvement or distortions of the outputs. Relate your discussion to specific characteristics observed in plots of the signals.**

Theory:

The Butterworth filter is a type of signal processing filter designed to have as flat frequency response as possible (no ripples) in the pass-band and zero roll off response in the stop-band. The low pass Butterworth filter is designed using signal. butter function in python with input order and normalized cut-off frequency
Normalised cut off frequency is given as

(cf/fs) *2

Where…
cf is cut off frequency and
fs= sampling frequency

Different order and cut-off frequency are chosen and Butterworth filter is designed and noise is removed using those filter and corresponding graphs are plotted below

Results and observations:

i. **Order 3, cut-off frequency 10Hz;**

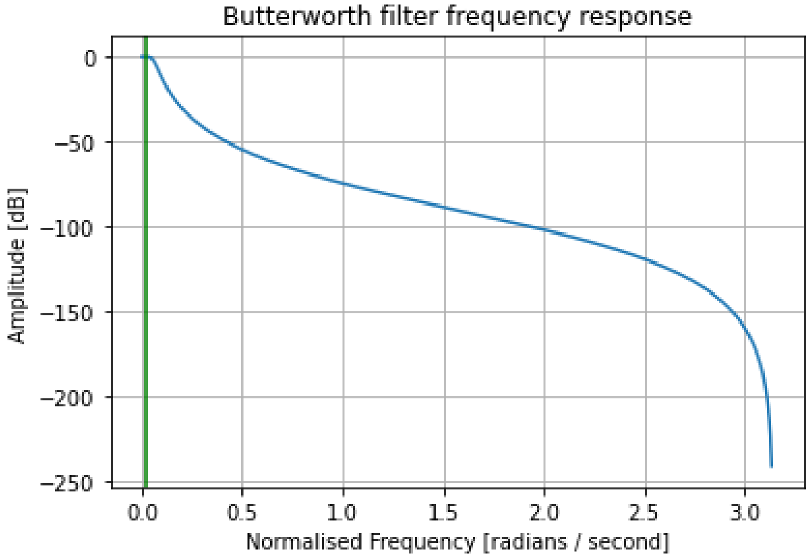

Figure 3a: Butterworth filter frequency response for Order 3, cut off frequency 10hz

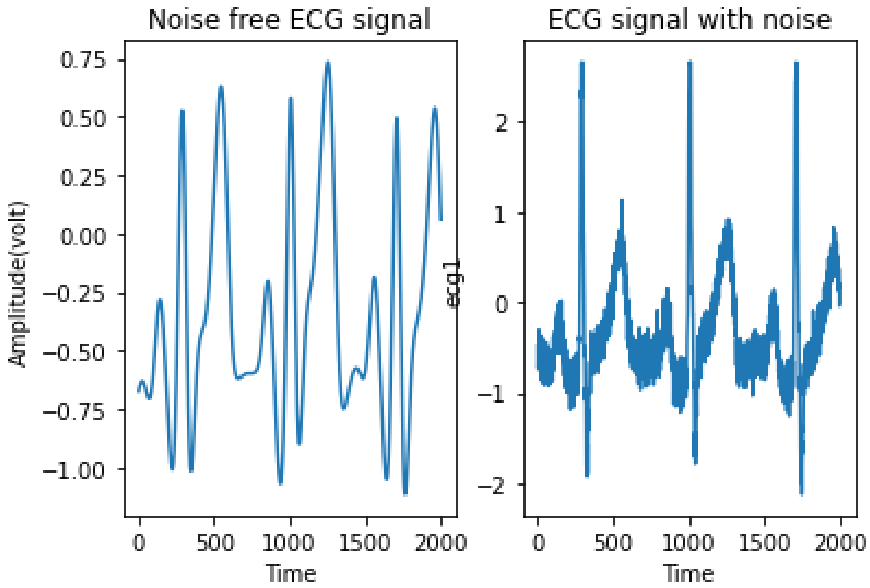

Figure 3b: comparison of ecg signal filtered with Butterworth filter of N=3 cut-off frequency =10 and noisy ecg signal

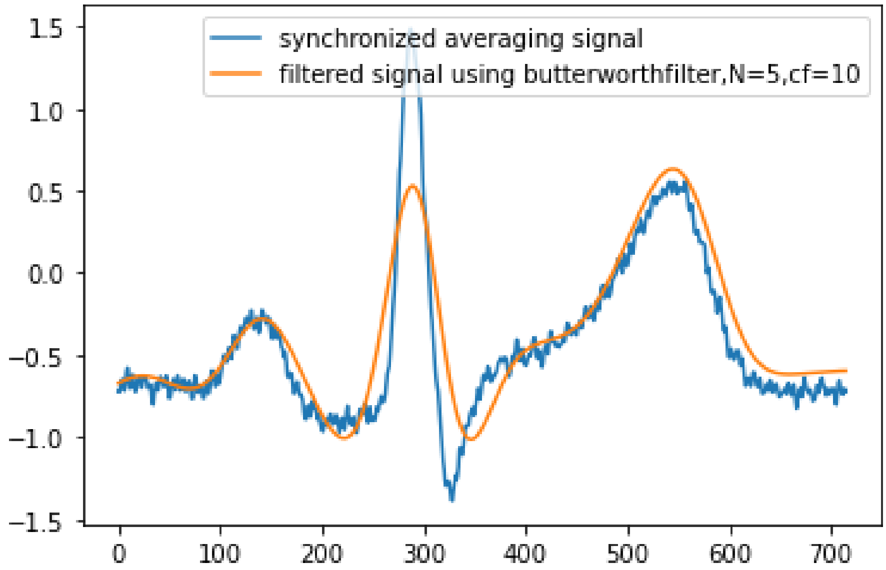

Figure 3c: Comparison of synchronized averaging signal with Butterworth filter of Order N=3, cut off frequency=10hz

ii. Order 5, cut-off frequency 10Hz;

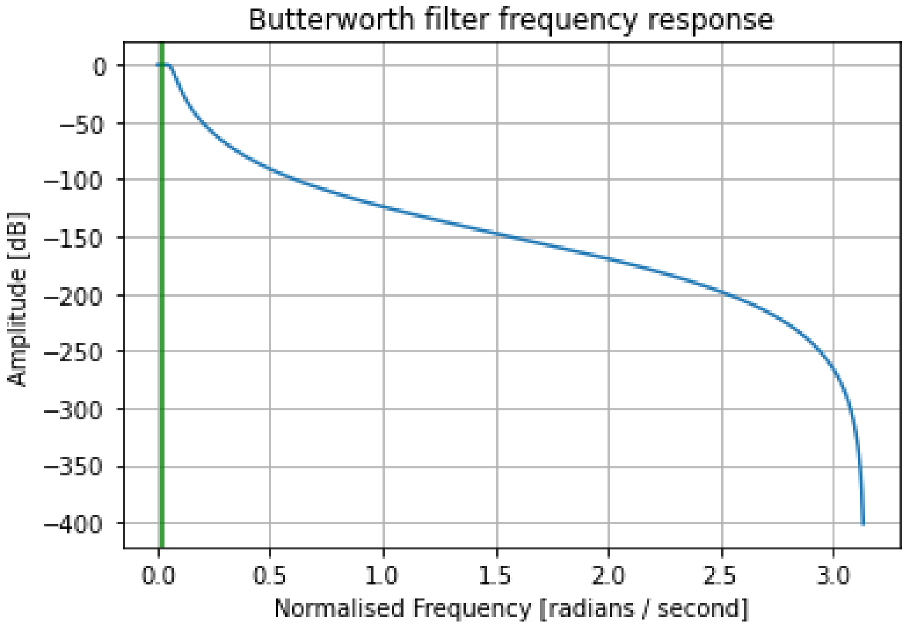

Figure 3d: Butterworth filter frequency response for Order 5,cut of frequency 10hz

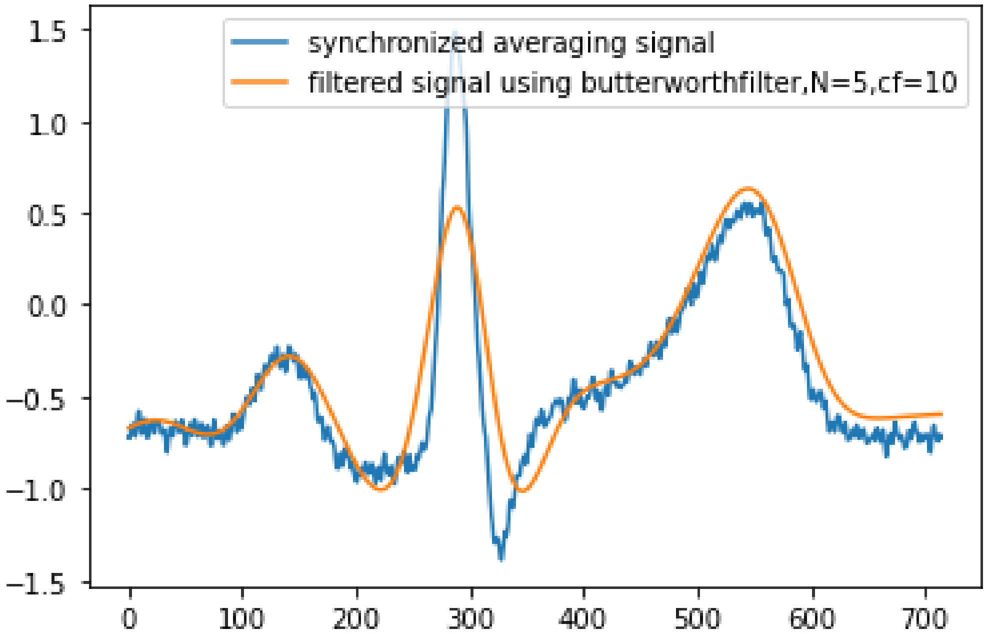

Figure 3e: Comparison of synchronized averaging signal with Butterworth filter of Order N=5, cut off frequency=10hz

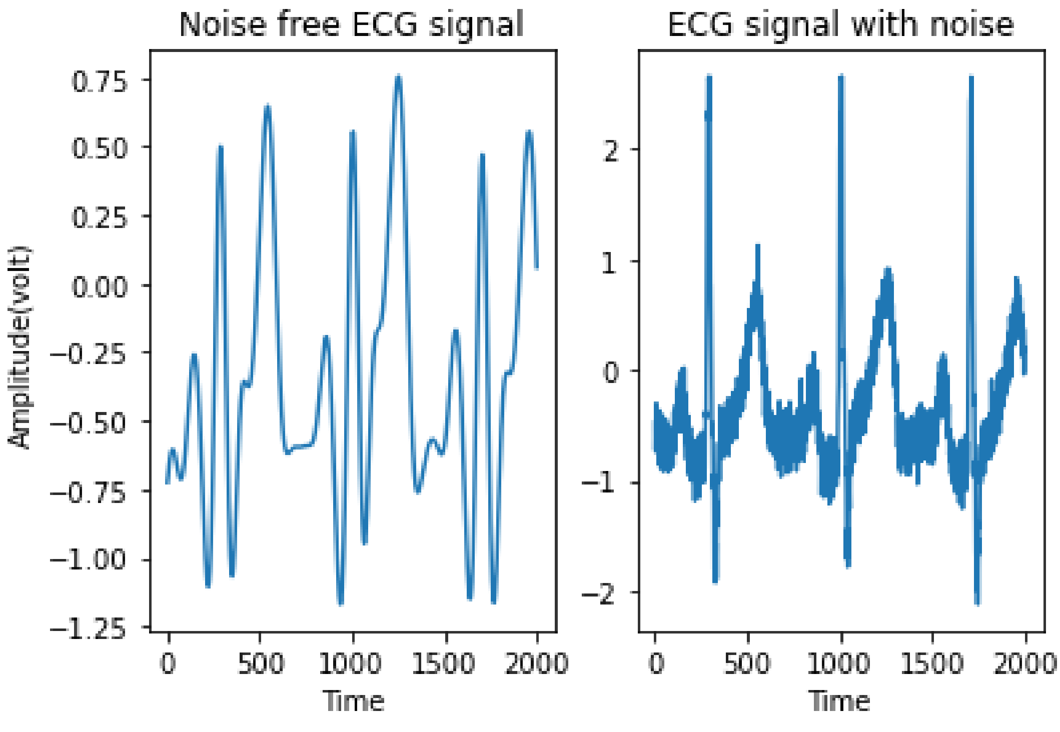

Figure 3f: comparison of ecg signal filtered with Butterworth filter of N=5 cut-off frequency =10 and noisy ecg signal

iii. Order 8, cut-off frequency 30Hz;

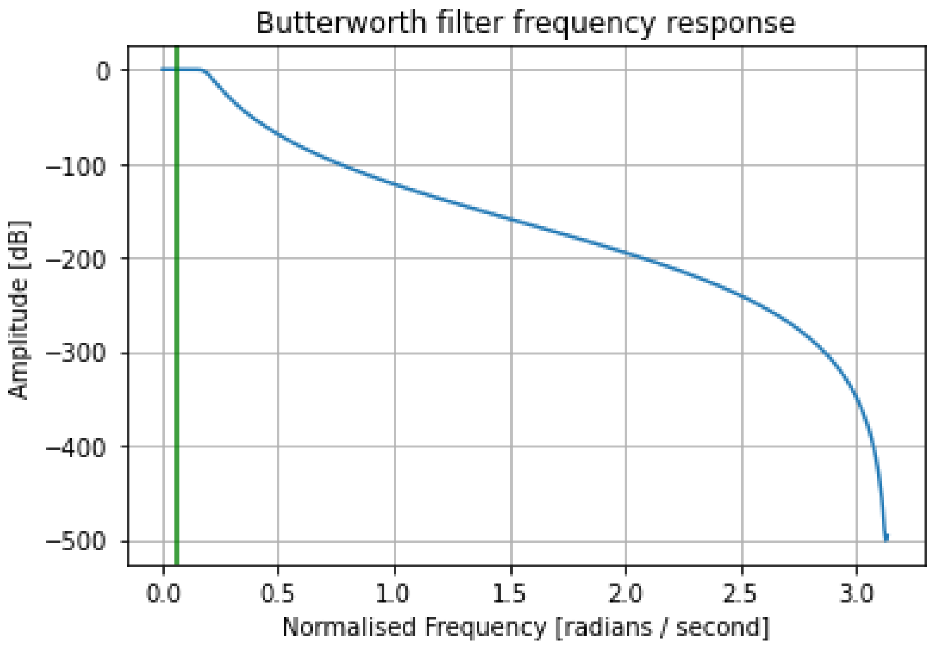

Figure 3g: Butterworth filter frequency response for Order 8, cut off frequency 30hz

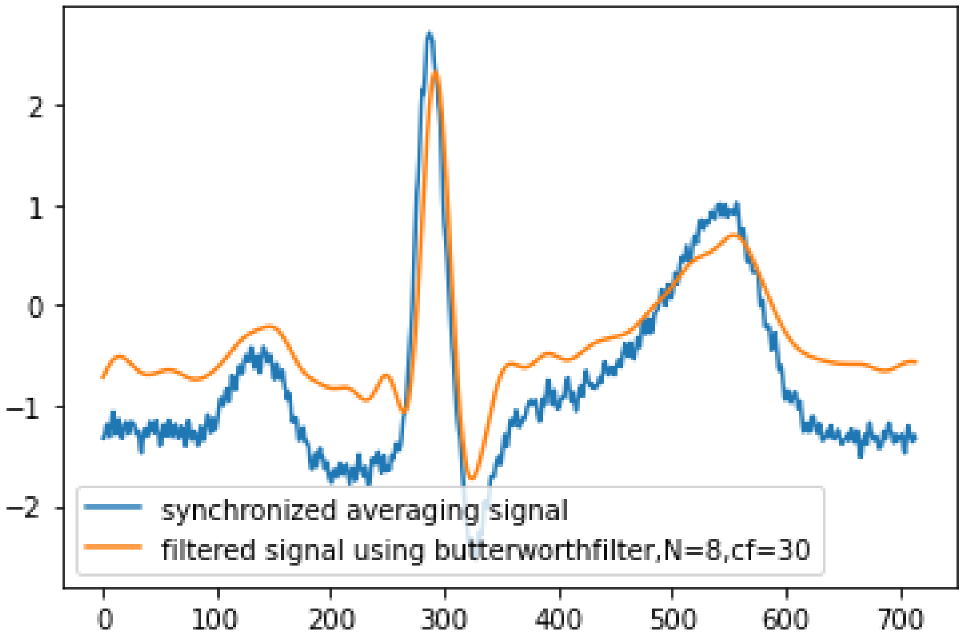

Figure 3h: Comparison of synchronized averaging signal with Butterworth filter of Order N=8, cut off frequency=30hz

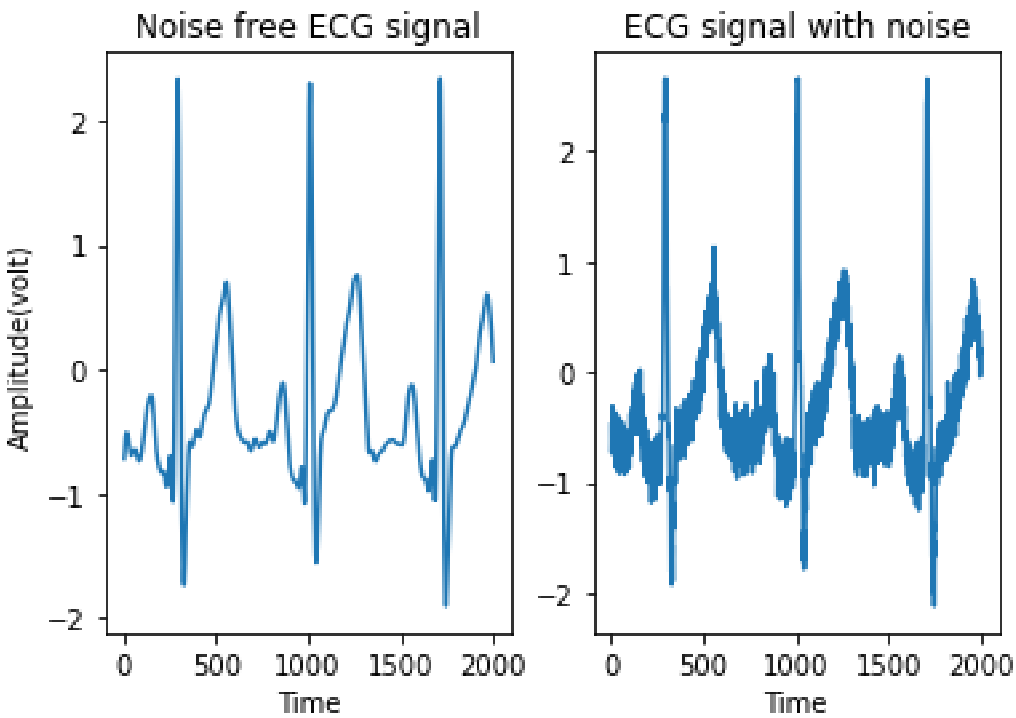

Figure 3i: comparison of ecg signal filtered with Butterworth filter of N=8 cut-off frequency =30 and noisy ecg signal

iv. Order 8, cut-off frequency 50Hz;

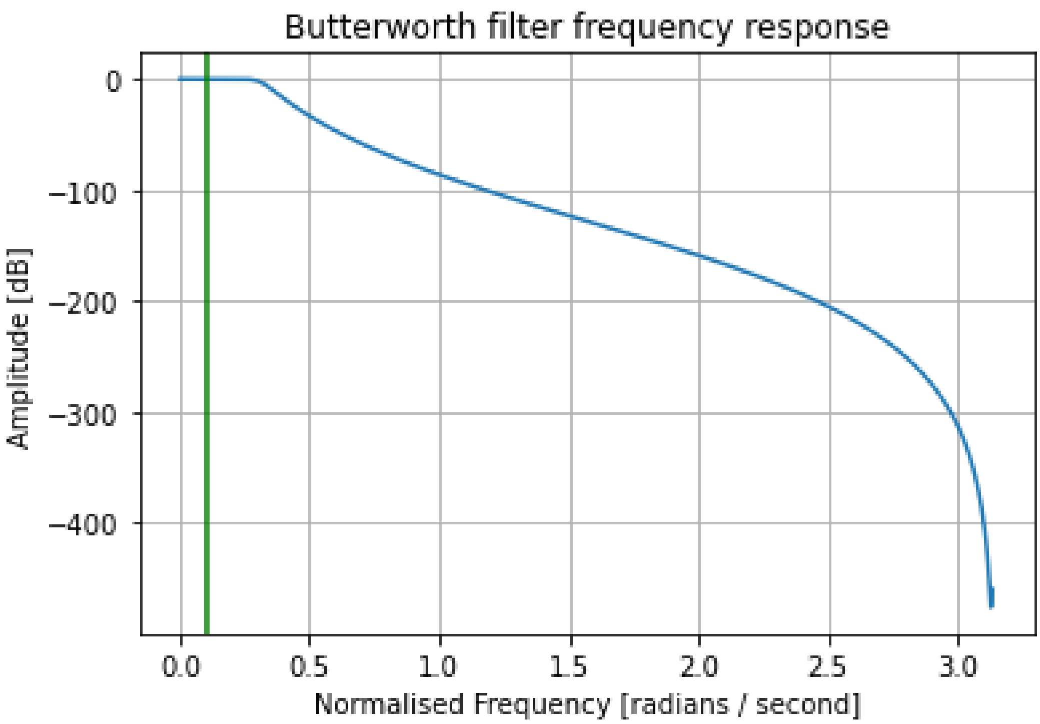

Figure 3j: Butterworth filter frequency response for Order 8, cut off frequency 50hz

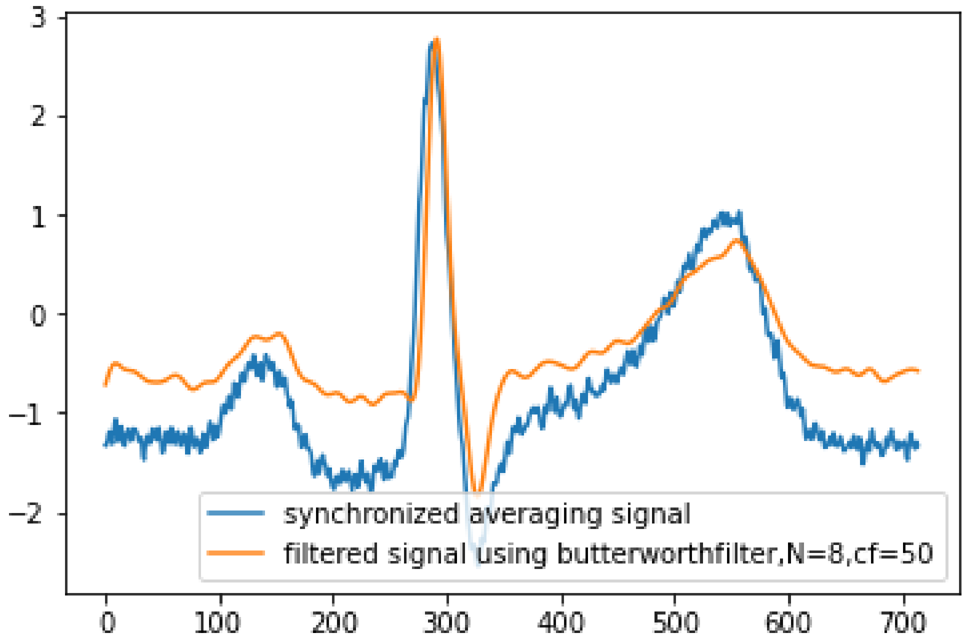

Figure 3k: Comparison of synchronized averaging signal with Butterworth filter of Order N=8, cut off frequency=50hz

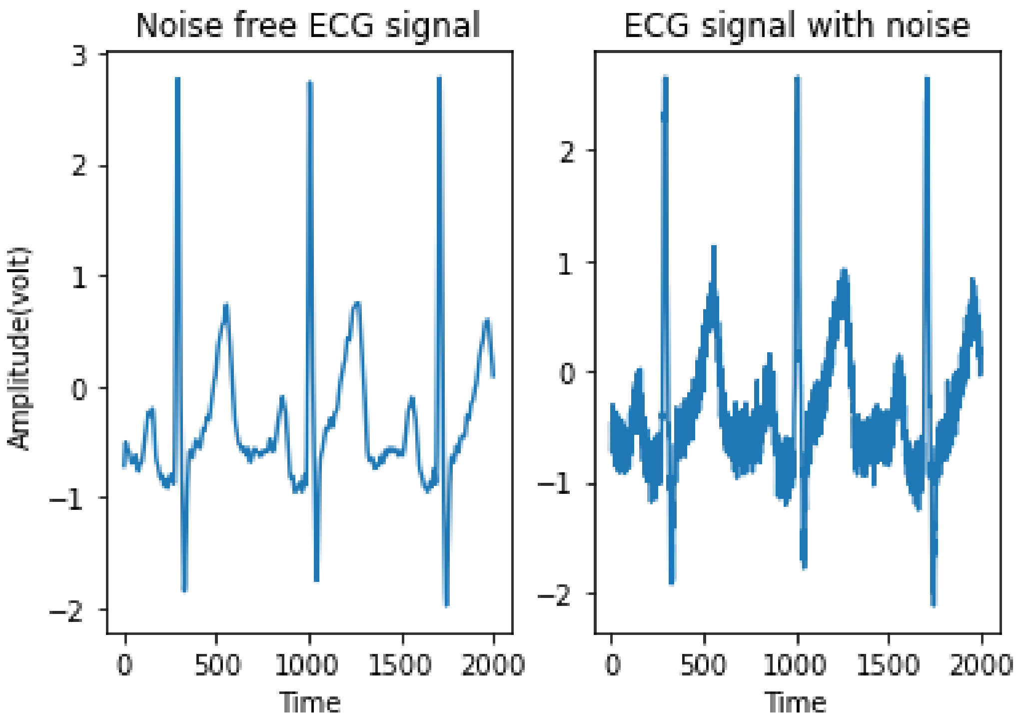

Figure 3l: comparison of ecg signal filtered with Butterworth filter of N=8 cut-off frequency =50 and noisy ecg signal

Conclusions and Discussions:

When we filter noisy ECG with least cut-off frequency Butterworth filter (from figure 3a,3b), we are getting unnecessary peaks of T peak is very high which is not desired.
If the order is increased then the noise is removed better i.e., higher order filter gives less noisy output than lower order filter.

Compared to the Synchronized average filter the response of filtered output using Butterworth filter is observed as follows, (Note here N is order and cf is cut off frequency)

N=3, cf=10: QRS peak of Synchronized average filter is more compared to Butterworth filtered output, but the Butterworth filtered output is smooth with less noise

N=5, cf=10: QRS peak of Synchronized average filter is more compared to Butterworth filtered output, but the Butterworth filtered output is smooth with less noise

N=8, cf=30: QRS peak of Synchronized average filter is still more compared to Butterworth filtered output, but better than above two the Butterworth filtered output is smooth with less noise

N=8, cf=50: QRS peak of Synchronized average filter is same compared to Butterworth filtered output, but the Butterworth filtered output is smooth with less noise.

So Butterworth filter is performed well for order 8 and cut-off frequency 50hz

b. Comment on the filtered outputs for different order but same cut-off frequency and for same order but different cut-off frequency

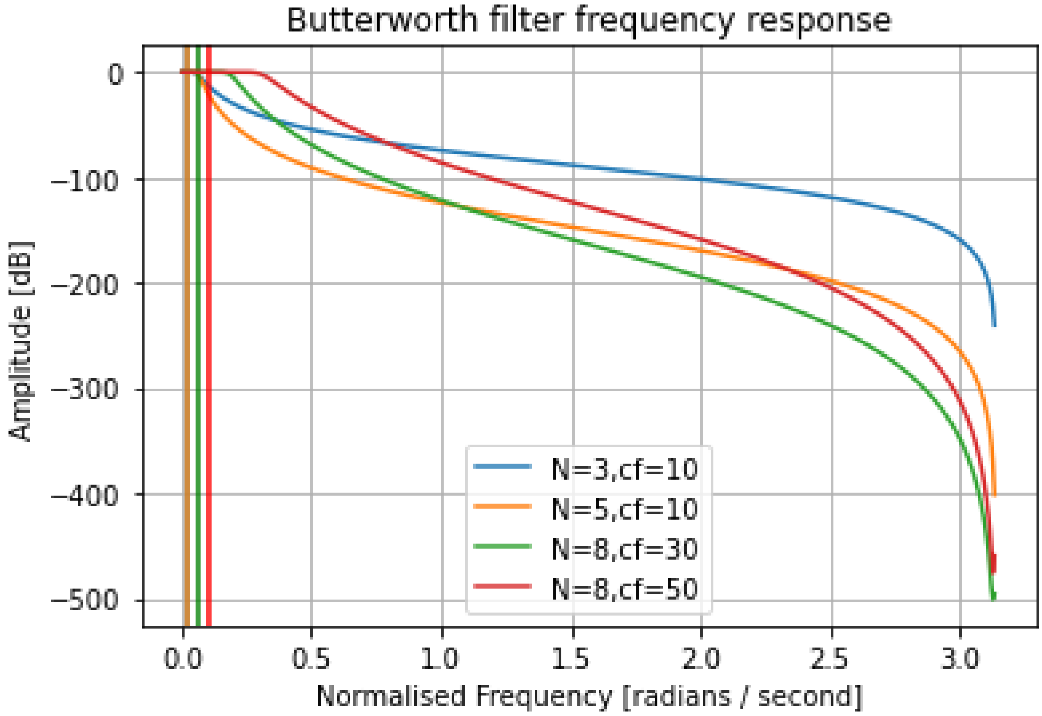

Figure 3m: Comparison of Butterworth frequency response for different orders and cut off frequency.

Conclusions and Discussions

From above figure 3m,

As order increases the attenuation is less (from N=3, N=5, cf=10).N is order, cf is cut-off frequency.

As the cut-off frequency increases the low pass Butterworth filter filters a smaller number of frequency components (from N=8, cf=30,50).

The filter with more order is better to filter the noise perfectly but the cost of implementation in real world would be high