Department of Electronic and Telecommunication Engineering

University of Moratuwa

BM4152 - Biosignal Processing

Assignment 2 - Optimum and Adaptive Filters

PUSHPAKUMARA H.M.R.M. 200488E

Date: October 11, 2024

Table of Content

1 Weiner filtering

Data Construction

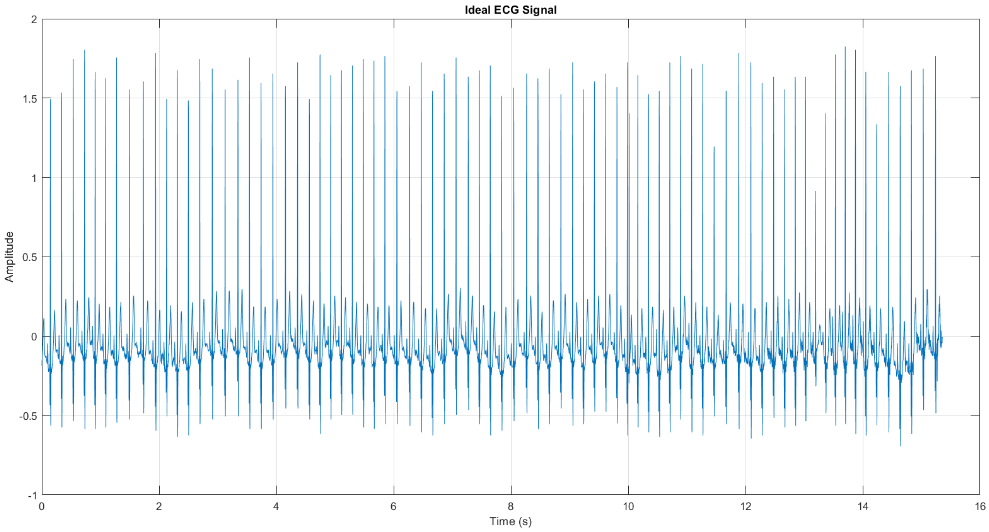

Figure 1: Full ECG Signal

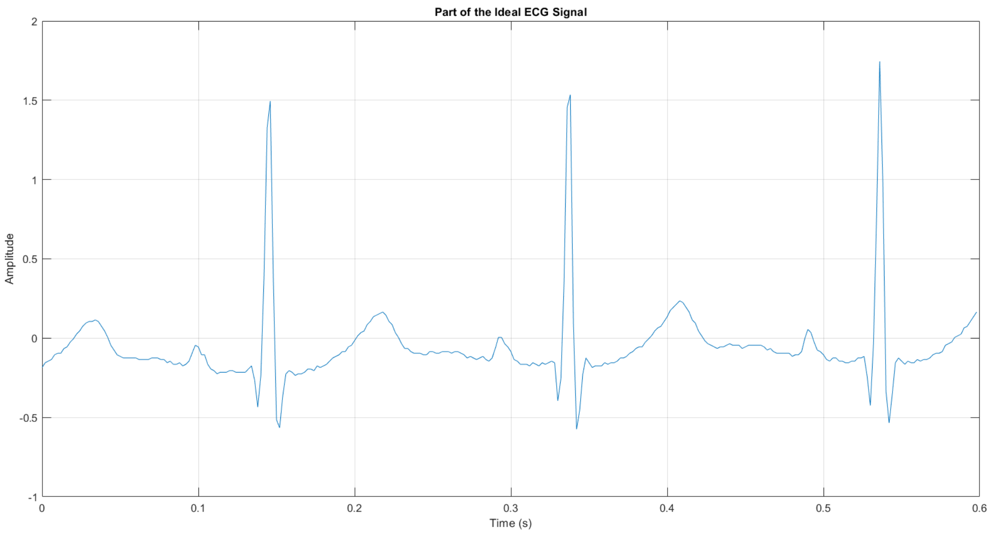

Figure 2: Part of the ECG Signal

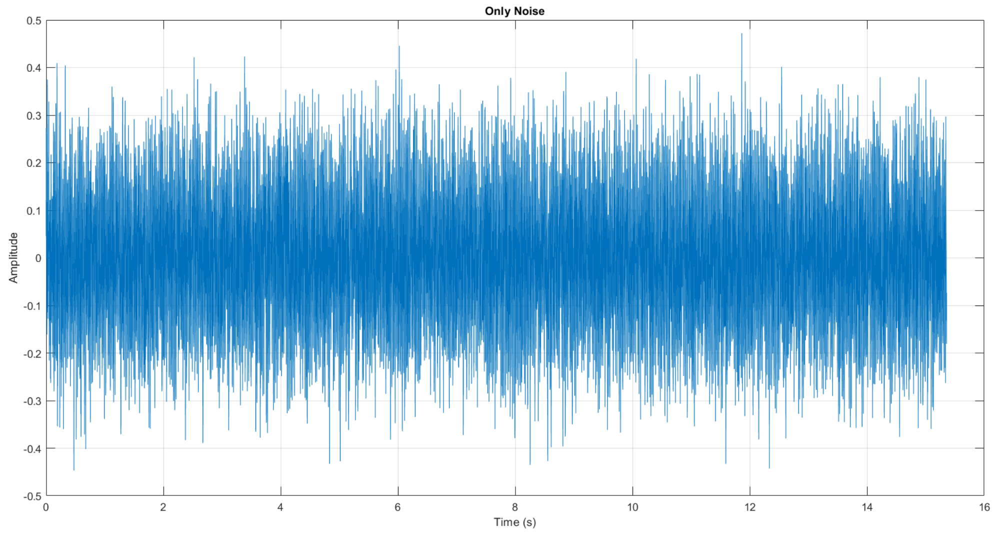

Figure 3: Generated nose

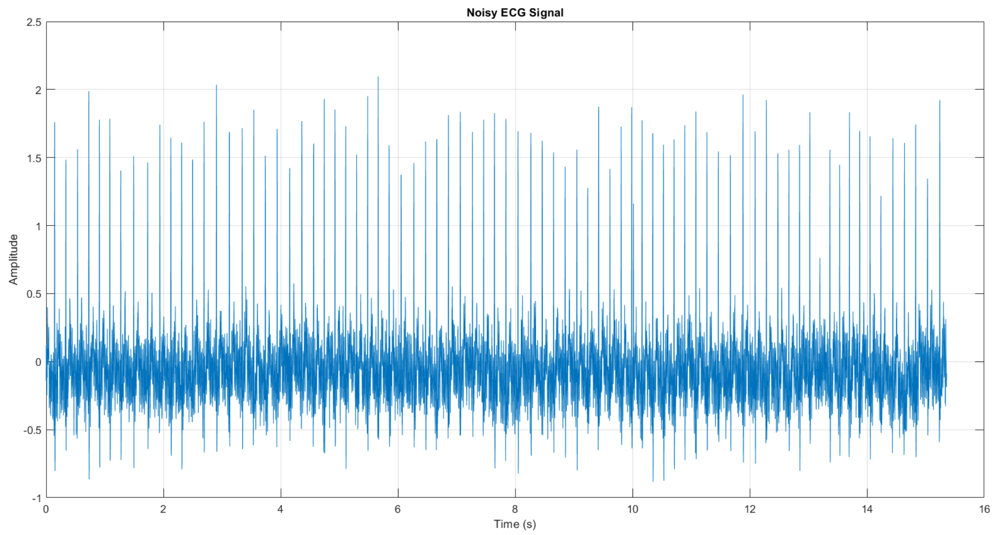

Figure 4: Noisy ECG signal

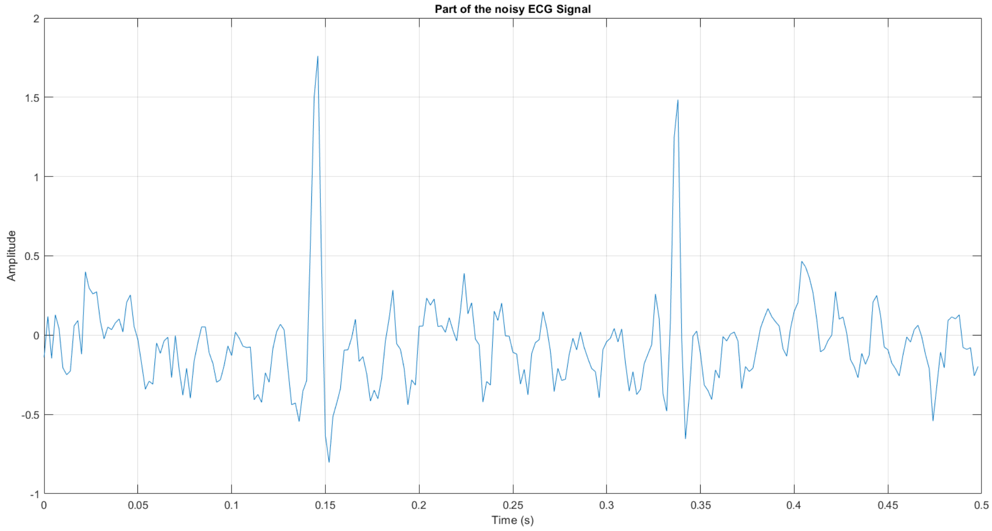

Figure 5: Part of the noisy ECG signal

1.1 Discrete time-domain implementation of the Wiener filter

1.1.1 Part 1

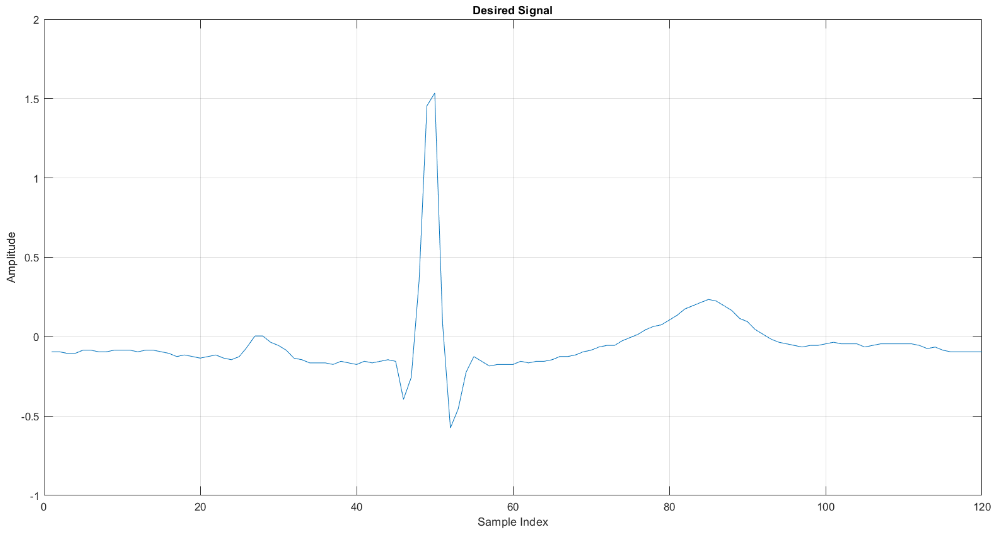

Figure 6: Expected Signal

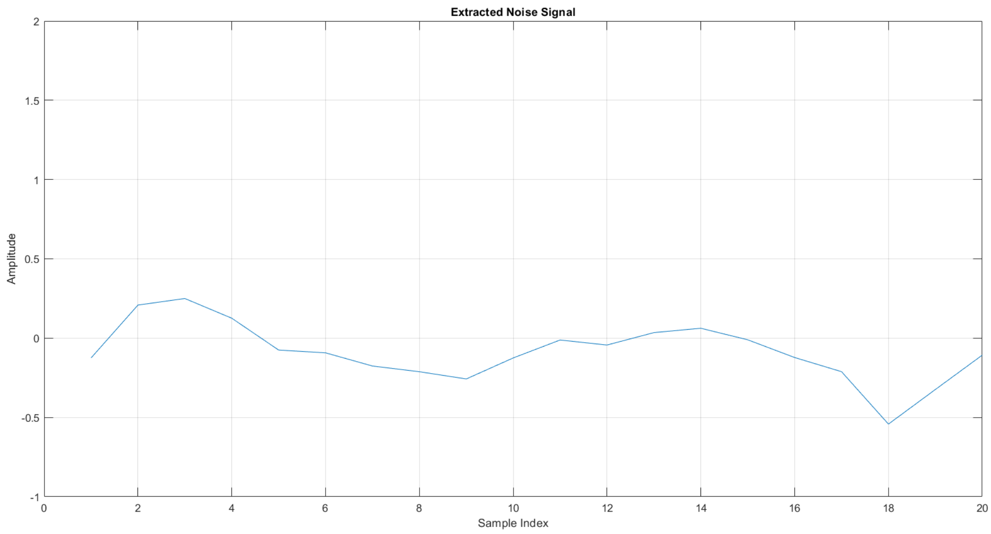

Figure 7: Extracted noise signal

a) Calculate the optimum weight vector for an arbitrary filter order.

- Order = 20

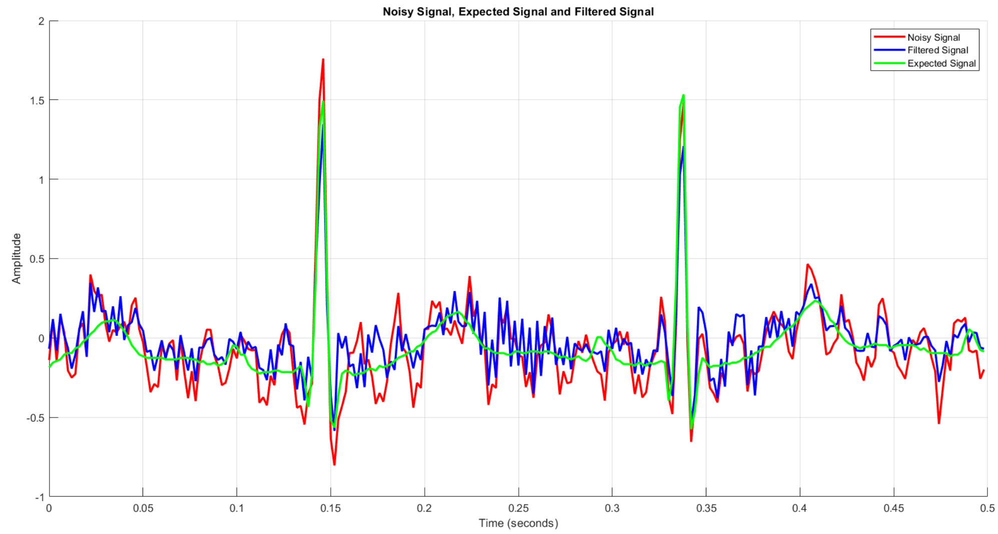

Figure 8: Noisy ECG signal, Expected ECG signal and the Filtered ECG signal with order 20 weiner filter.

b) Find the optimum filter order and its coefficients. Plot the magnitude response

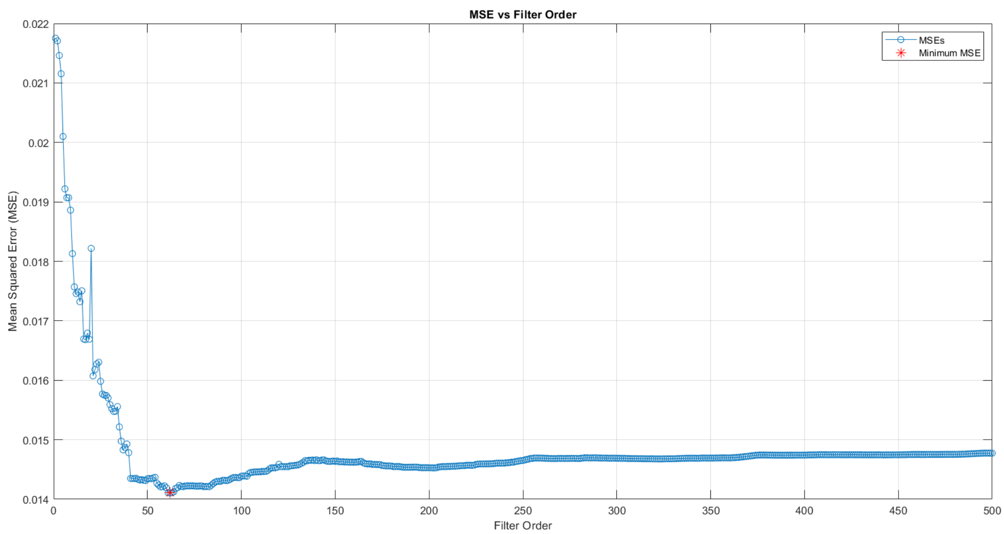

Figure 9: MSE values with filter orders

According to the obtained plot, the optimum filter **order is 62** and the corresponding **MSE is 0.014107**.

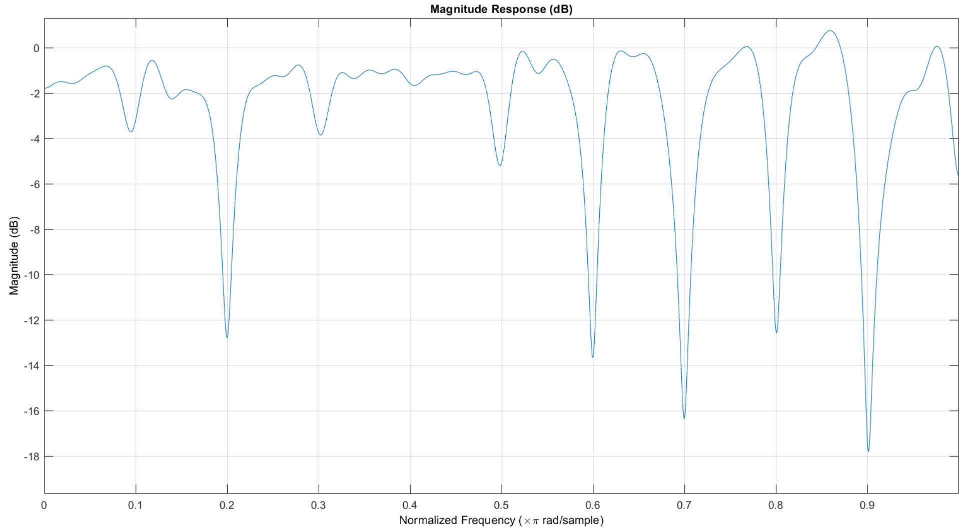

Figure 10: Magnitude response of the optimum filter

c) Obtain the filtered signal using the filter obtained above

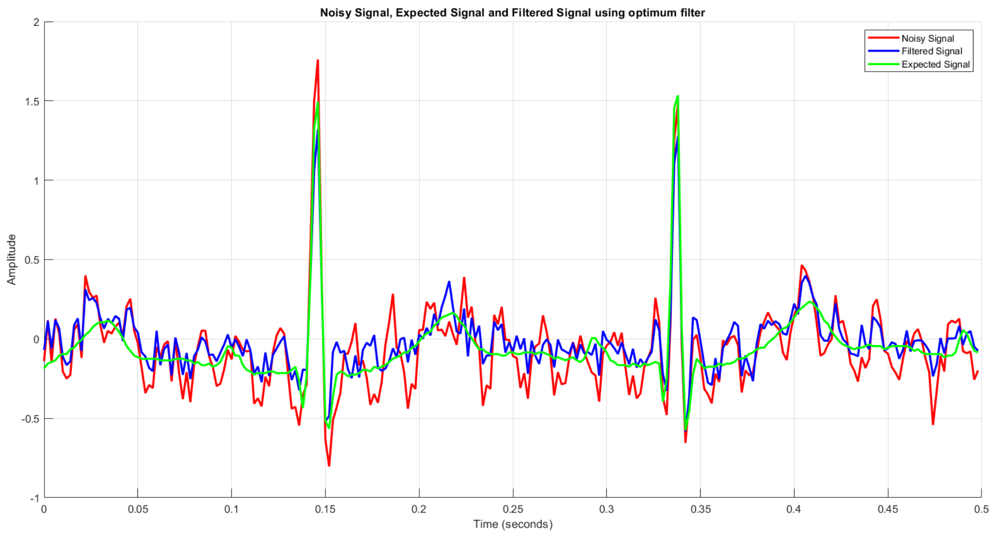

Figure 11: Filtered ECG signal with optimum filter order

d) Plot the spectra

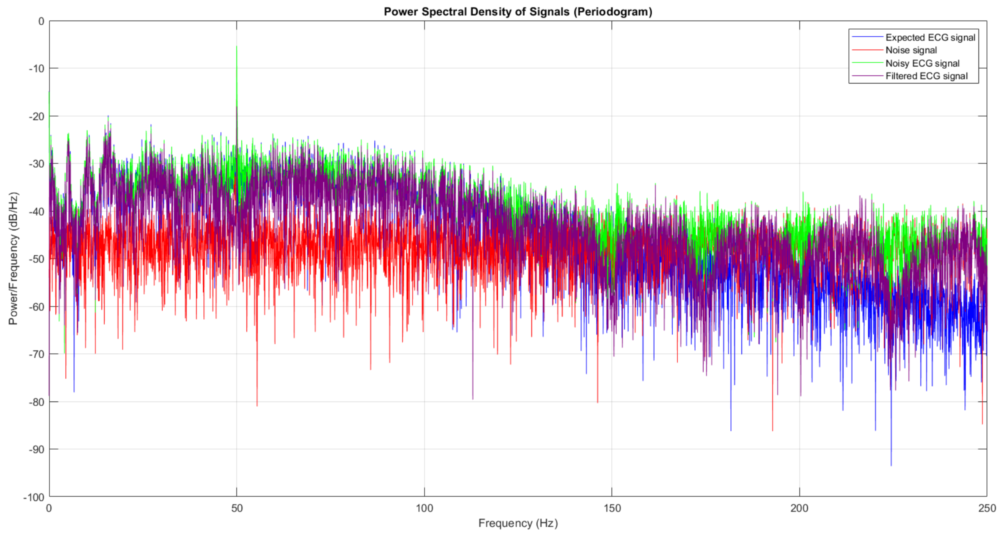

Figure 12: Power Spectral densities of the filtered ECG signal, noisy ECG signal, expected ECG signal and the noise.

Here we can see the filter has removed the 50Hz noise well and also has removed other unnecessary noises quite better.

1.1.2 Part 2

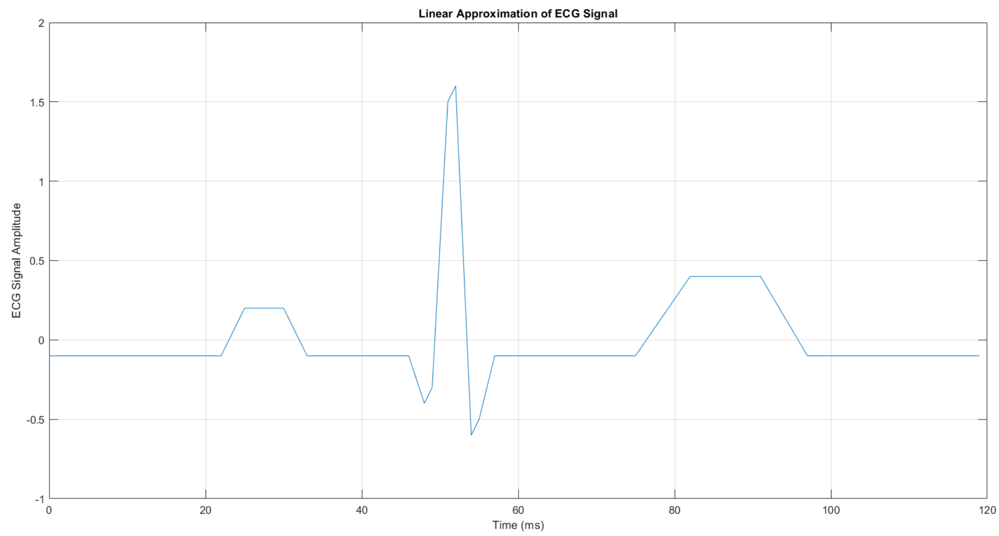

Figure 13: Expected Signal

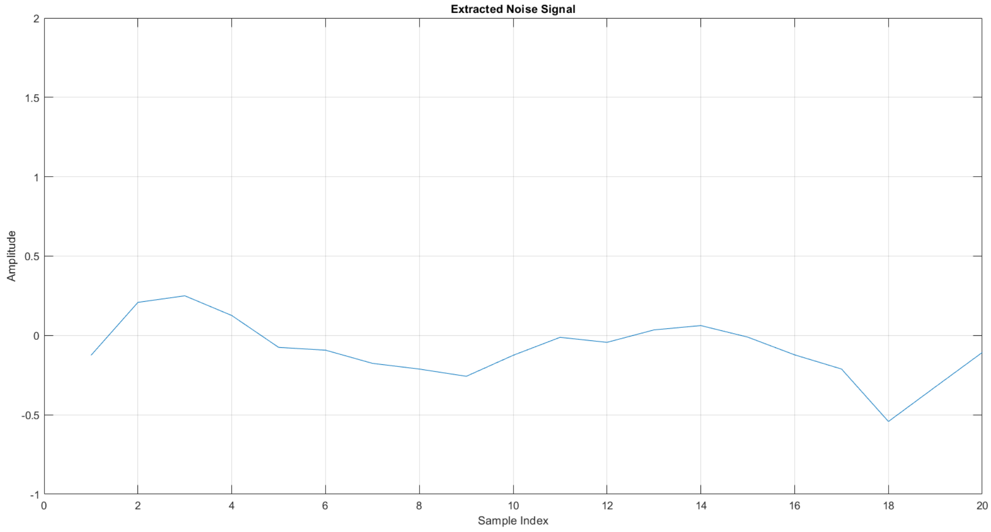

Figure 14: Extracted noise signal

a) Calculate the optimum weight vector for an arbitrary filter order.

- Order = 20

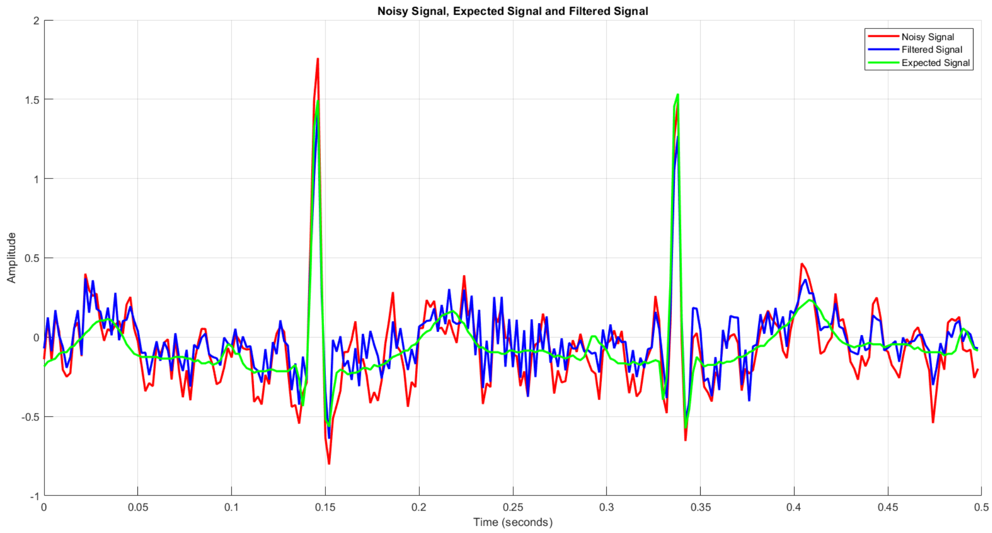

Figure 15: Noisy ECG signal, Expected ECG signal and the Filtered ECG signal with order 20 weiner filter.

b) Find the optimum filter order and its coefficients. Plot the magnitude response

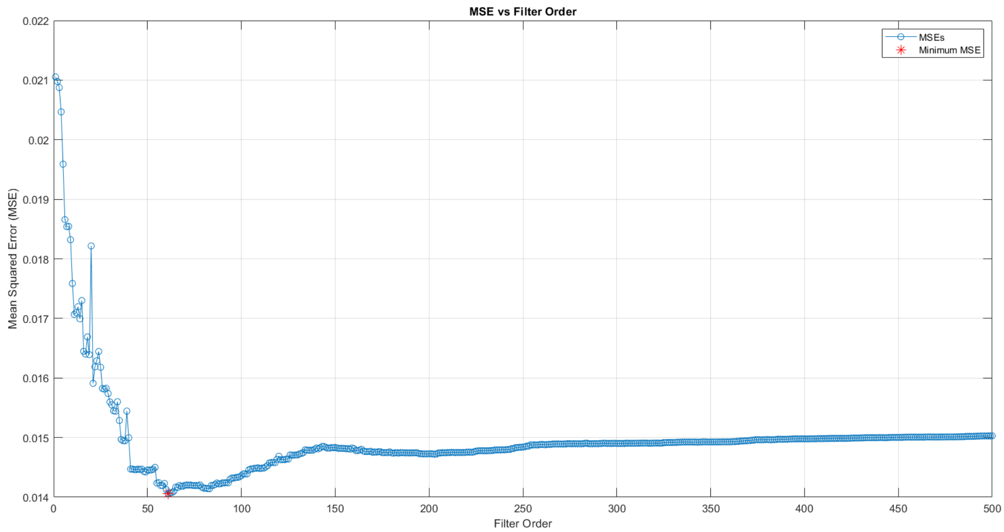

Figure 16: MSE values with filter orders

According to the obtained plot, the optimum filter **order is 61** and the corresponding **MSE is 0.014062**.

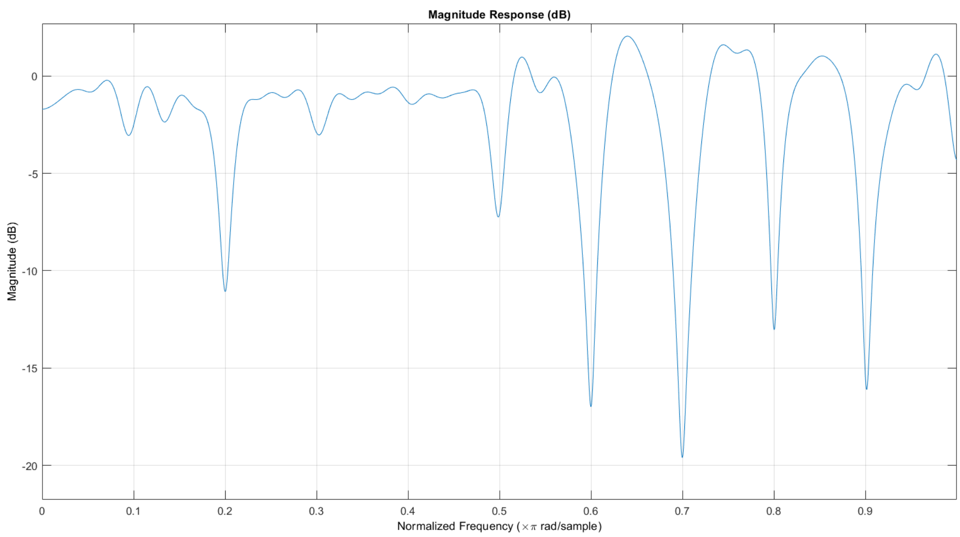

Figure 17: Magnitude response of the optimum filter

c) Obtain the filtered signal using the filter obtained above

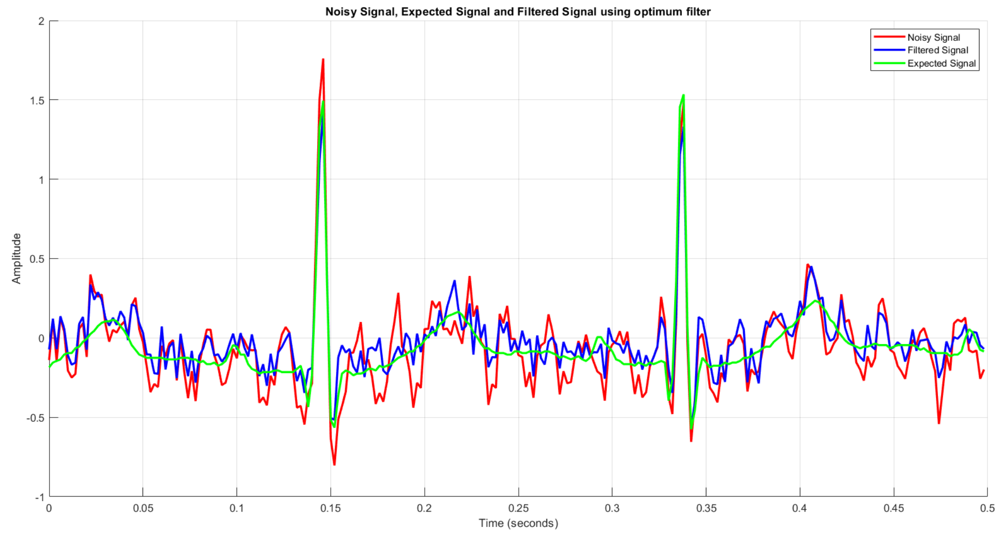

Figure 18: Filtered ECG signal with optimum filter order

d) Plot the spectra

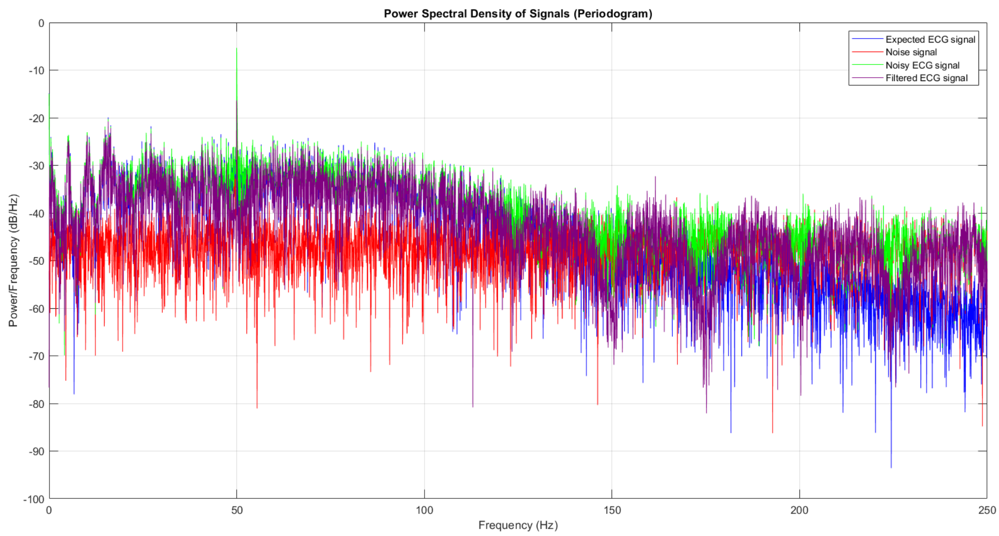

Figure 19: Power Spectral densities of the filtered ECG signal, noisy ECG signal, expected ECG signal and the noise.

Here we can see the filter has removed the 50Hz noise well and also has removed other unnecessary noises quite better. But it has more noise than the spectral in part 1.

1.2 Frequency domain implementation of the Wiener filter

a) Implement the frequency domain version of the Wiener filter for the two cases given in Part 1 and Part 2 above.

1.2.1 Part 1

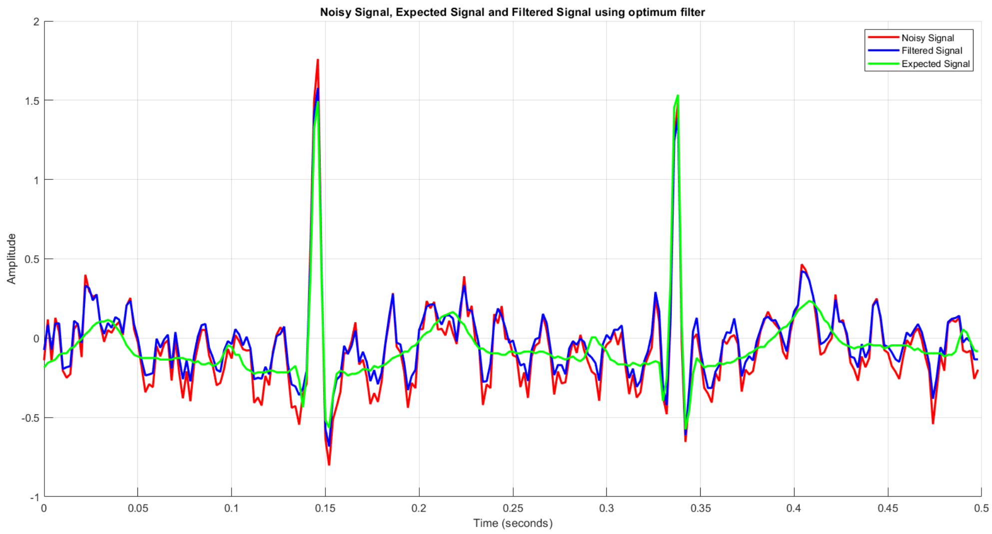

Figure 20: Frequency domain weiner filtering using extracted single beat as the desired signal and the isoelectric segment as the noise signal.

1.2.2 Part 2

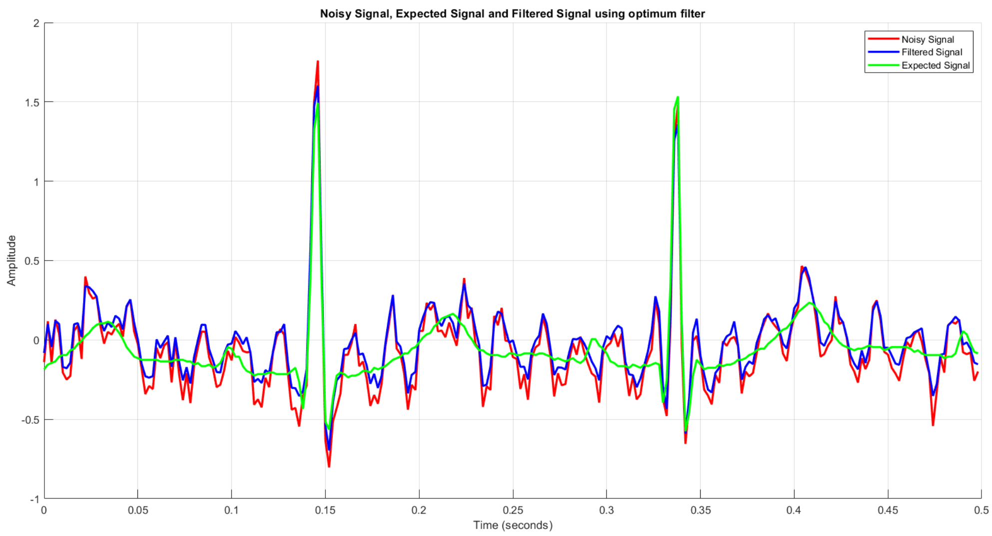

Figure 21: Frequency domain weiner filtering using the constructed liner ECG as the desired signal and the isoelectric segment as the noise signal.

b) Compare the filtered signal with Part 1 and Part 2 with the aid of a plot and the mean squared error with respect to the ideal signal

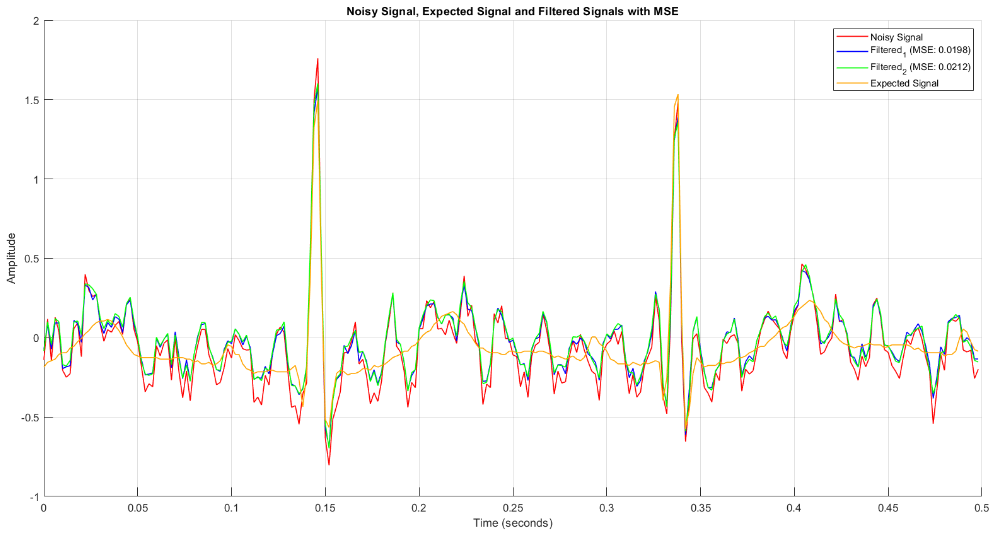

Figure 22: Frequency domain weiner filtering comparison (Part 1 & Part 2)

Here we can see the MSE of the filtered signal by using an extracted single ECG beat as the desired signal is **0.0198** and the MSE of the filtered signal by using the constructed linear ECG model is **0.0212**. This MSE difference is because the extracted signal is more similar to an actual ECG than the constructed one. Therefore the MSE value for the extracted signal is lower than the constructed signal.

1.3 Effect on non-stationary noise on Wiener filtering

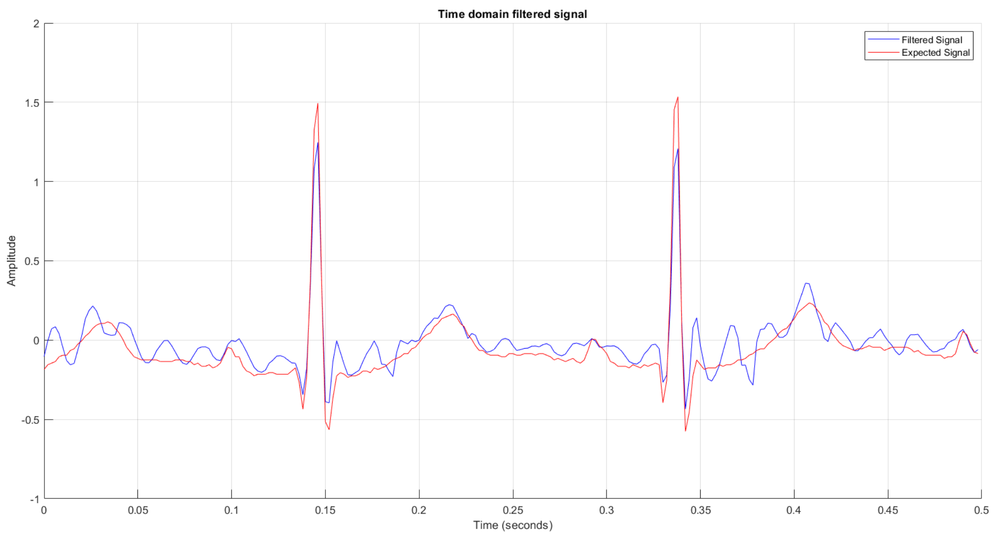

Figure 23: Apply time domain weiner filter for the non-stationary signal

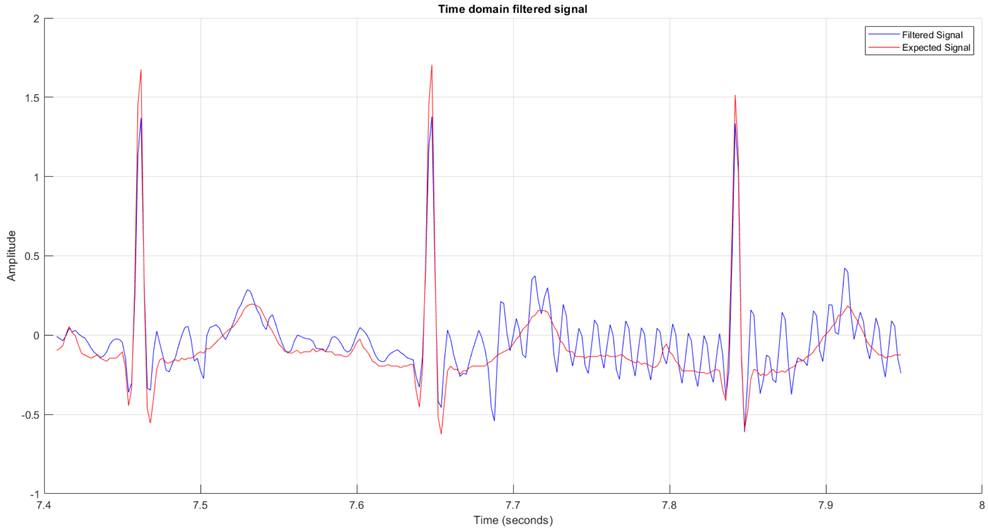

Figure 24: Time domain weiner filter for the non-stationary signal when the noise is changing (50Hz to 100Hz)

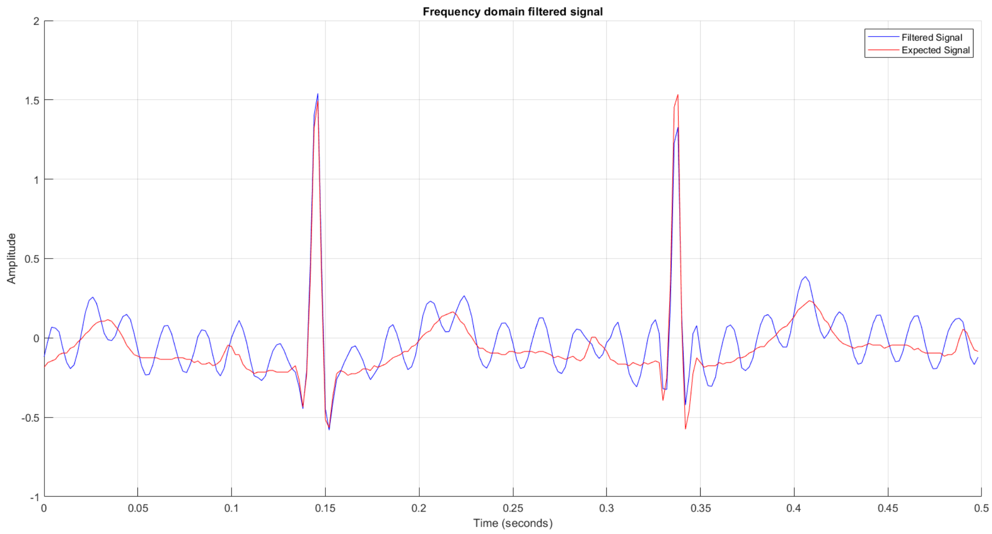

Figure 25: Apply frequency domain weiner filter for the non-stationary signal

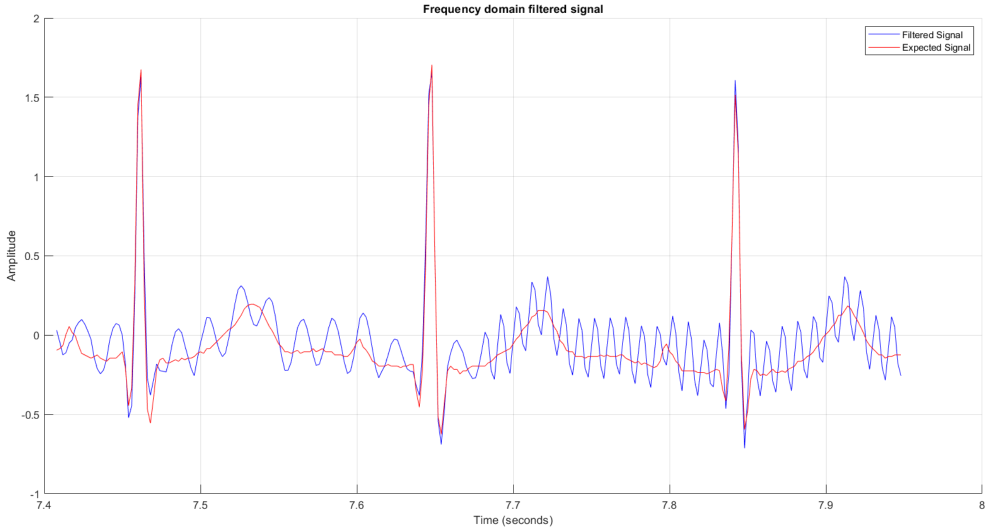

Figure 26: Frequency domain weiner filter for the non-stationary signal when the noise is changing (50Hz to 100Hz)

Weiner filter is designed for signals with statistical properties (such as mean, variance, and autocorrelation) that do not change over time. When we apply that to non-stationary noise, we can see that it has not performed well. We can see in the plots, the signal has under-filtered (allowing more noise to pass through) in regions where the signal properties deviate from the stationary assumption.

2 Adaptive filtering

Data Construction

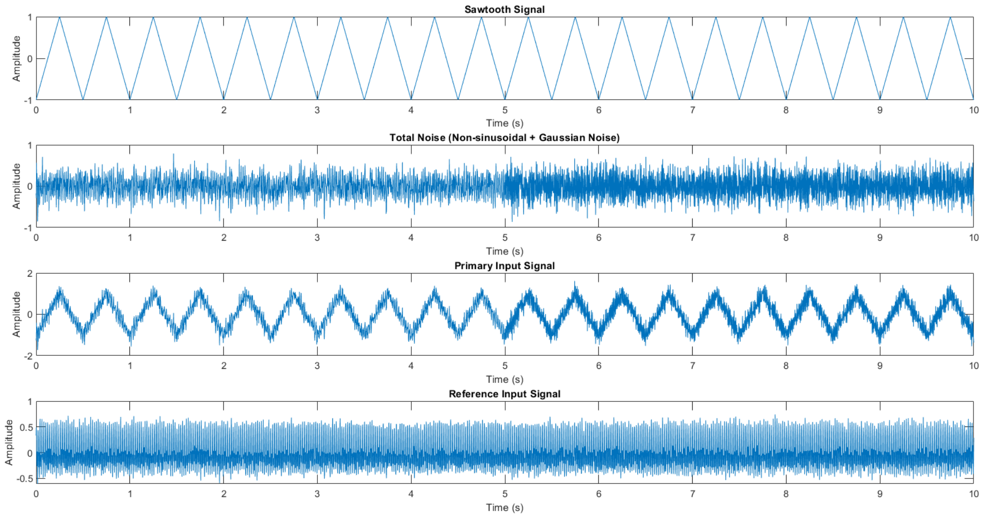

Figure 27: Constructed sawtooth wave, noise wave, primary input wave (sawtooth wave + noise wave), and reference input (noise reference)

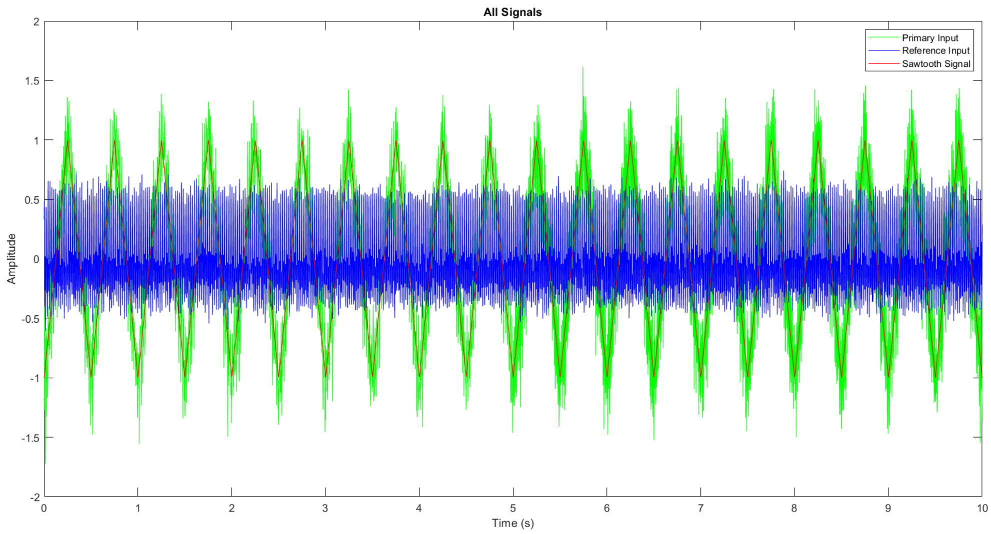

Figure 28: Sawtooth wave, primary input and the reference input

2.1 LMS method

a) Implement the LMS method as per the equation.

b) Plot the signals

- Filter order = 20

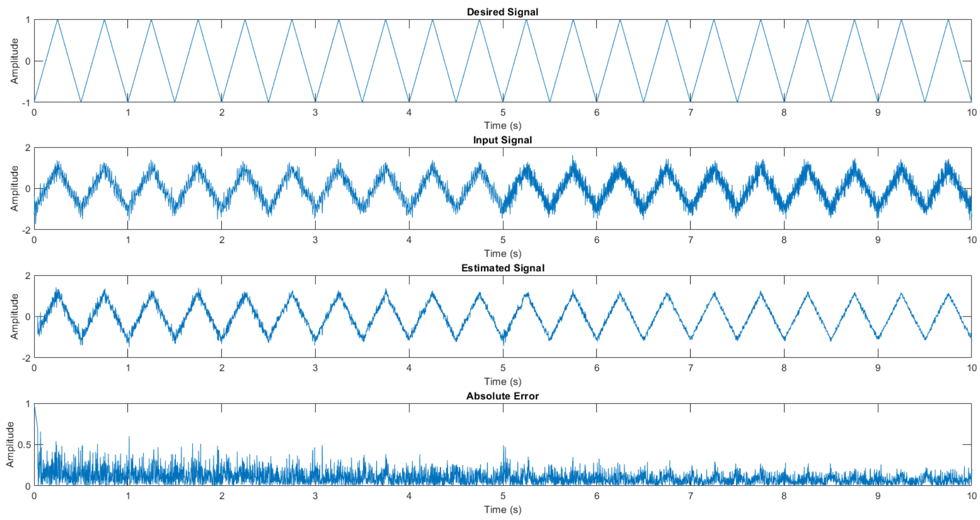

Figure 29: Plot of the desired signal, primary input (noisy signal), filtered signal, and the absolute error.

c) Explore the rate of adaptation by varying the rate of convergence and the order of the adaptive filter. To quantify you may calculate the mean squared error with respect to the desired signal.

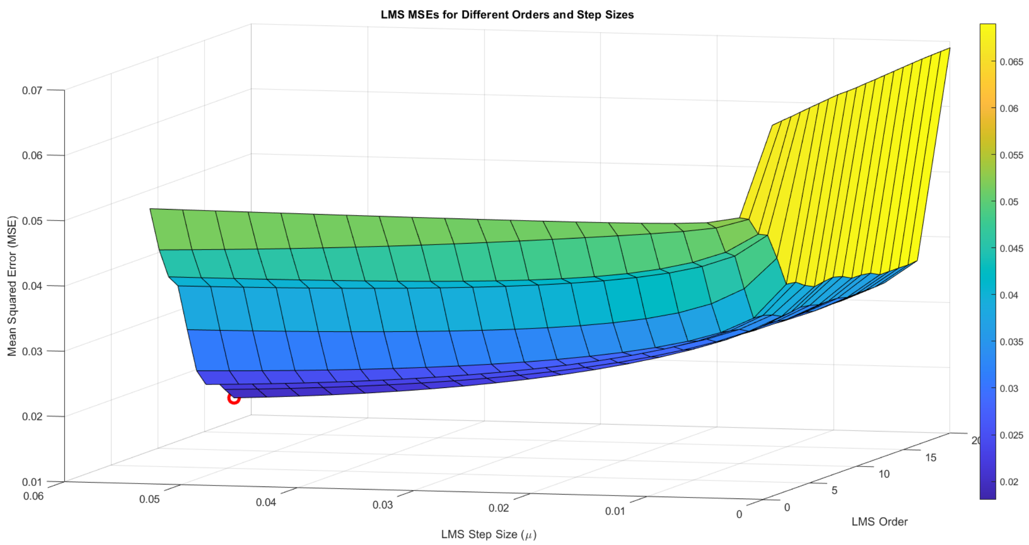

Figure 30: Plot of the MSEs with the rate of convergence and the filter order.

Here, for the rate of convergence μ, I took 20 values between 0 and 0.0535. 0.0535 is the maximum value that μ can take according to the inequality,

$$0 < \mu < \frac{2}{\lambda_{\text{max}}} where \lambda_{\text{max}} = 20 * M * Px \tag{1}$$

M is the filter order and the *Px* is the signal power of the reference input.

The obtained optimum values are;

- filter order: 10
- rate of convergence (μ): 0.053456
- MSE value: 0.018048

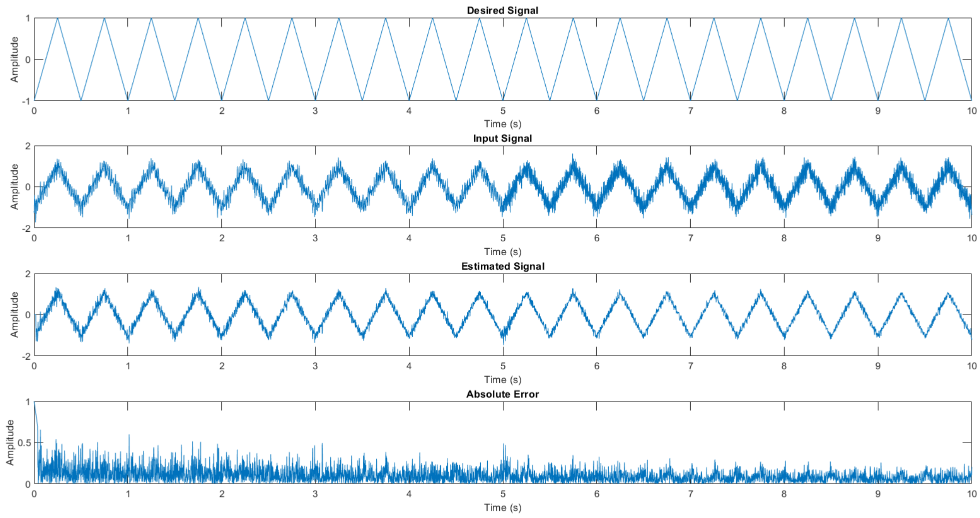

Figure 31: Desired signal, primary input (noisy signal), filtered signal using the optimum filter values, and the absolute error.

2.2 RLS method

a) Implement the RLS method as per the equations.

This is the output signal from this implemented RLS filter.

- filter order = 15

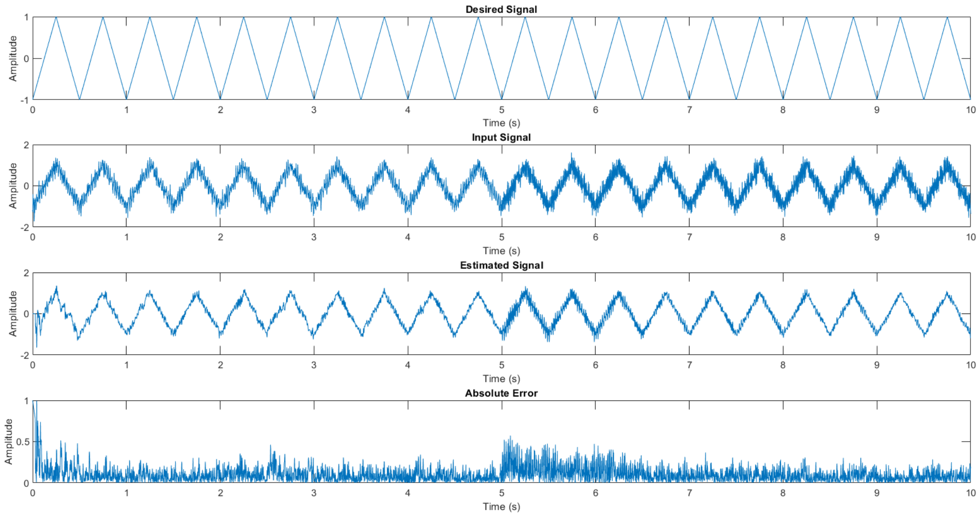

Figure 32: Desired signal, primary input (noisy signal), filtered signal, and the absolute error.

b) Explore the rate of adaptation by varying the forgetting factor and the order of the filter. To quantify you may calculate the mean squared error with respect to the desired signal.

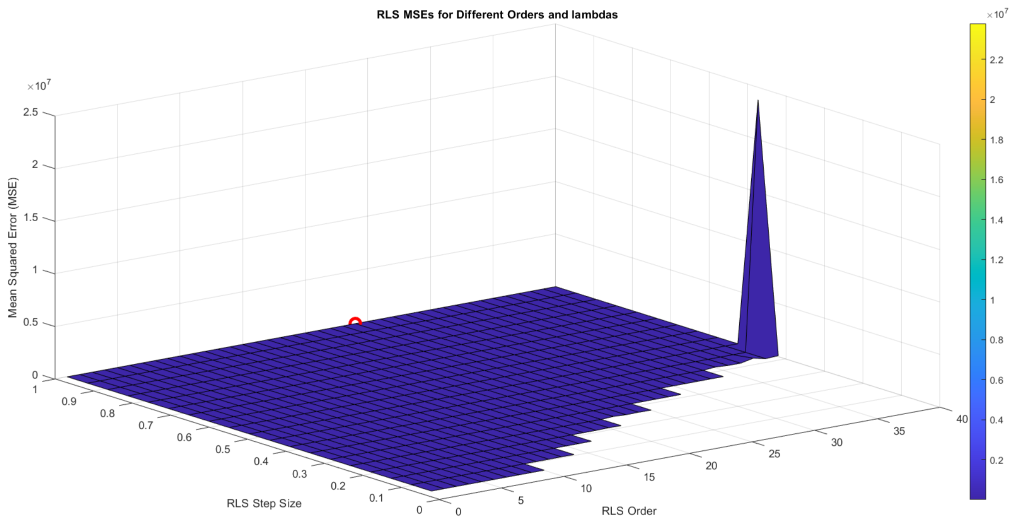

Figure 33: Plot of the MSEs with the forgetting factor and the filter order

Here, for the forgetting factor, I have taken 20 values between 0 and 1 (λ_{max} should be equal to 1). The filter order has increased from 0 to 40. From this plot, the obtained optimum values are

- filter order: 24
- forgetting factor (λ): 1
- MSE value: 0.029491

But I observed that, when forgetting factor (λ) closes to 1, this performs even better when I design the filter by using arbitrary values in the above part a).

Therefore, I again did this plot by using different value range for the forgetting factor. This time, the selected value range contained 20 values in between 0.99 and 0.9999, and the filter orders were same as before.

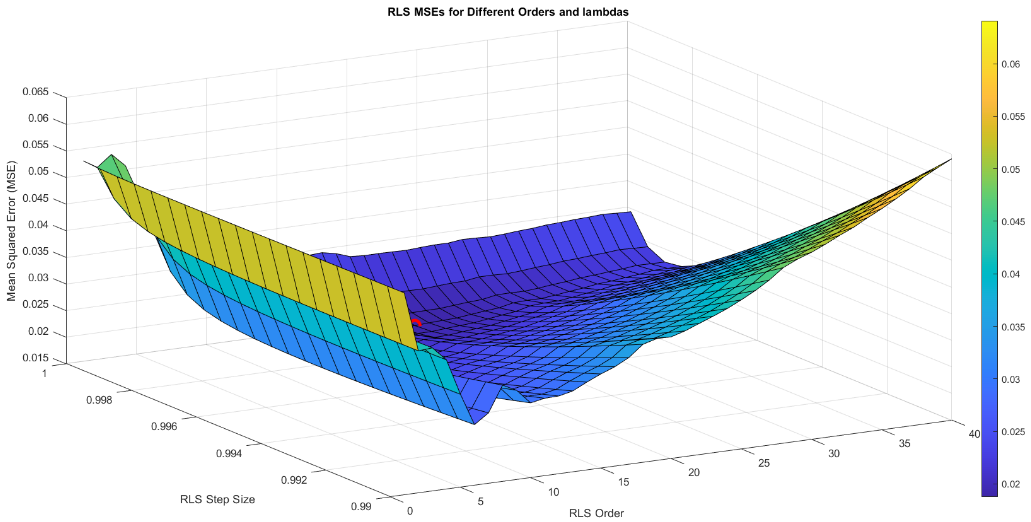

Figure 34: Plot of the MSEs with the forgetting factor and the filter order

From this plot, the obtained optimum values are

- filter order: 21
- forgetting factor (λ): 0.099834
- MSE value: 0.018765

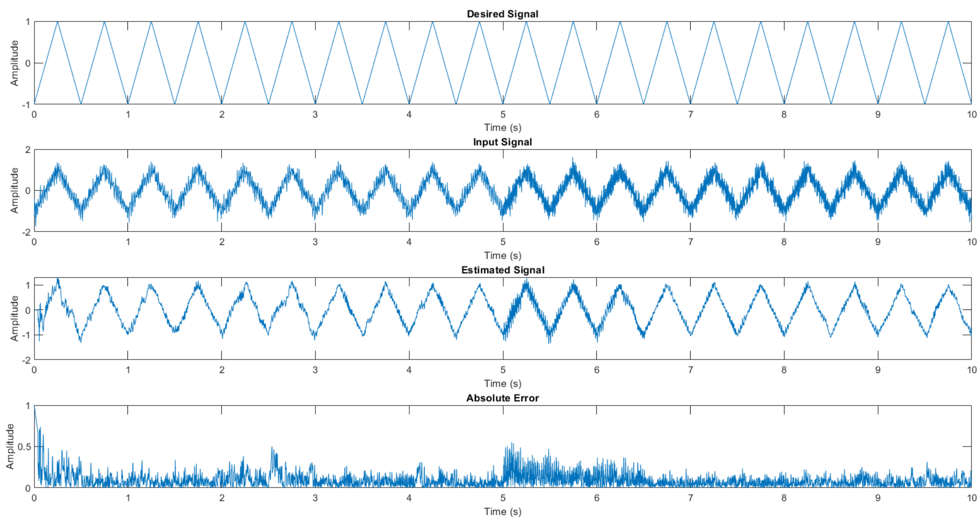

Figure 35: Plot of the desired signal, primary input (noisy signal), filtered signal using the optimum filter values, and the absolute error.

c) Compare the performance of LMS and RLS algorithms using a plot

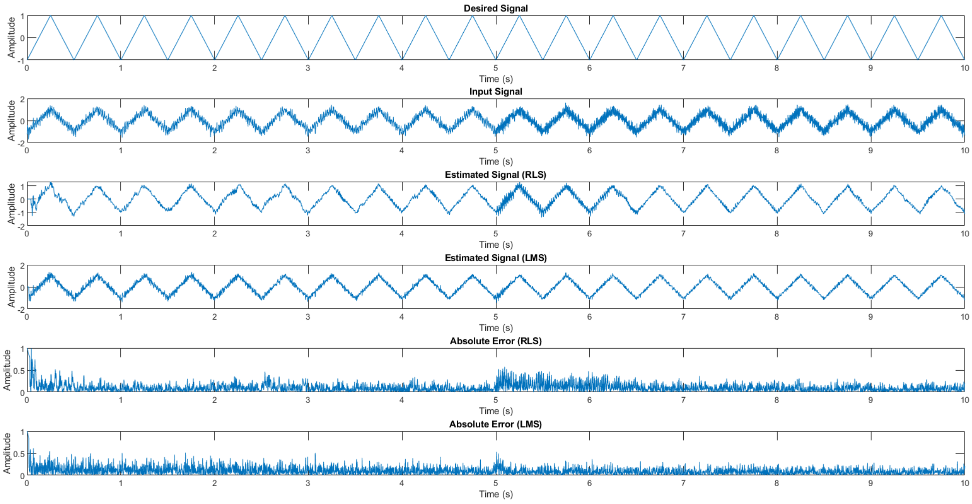

Figure 36: Plot of the desired signal, primary input (noisy signal), filtered signals using the optimum RLS and LMS filters, and the absolute errors of RLS and LMS filters.

By comparing the above plot we can say, in this case, LMS has performed well over RLS. We can see, from the beginning, RLS trying to converge more rapidly than the LMS, but because of that, there might be some instability has occurred and the signal has not filtered properly in the RLS approach.

d) Now test LMS and RLS algorithms using the idealECG.mat

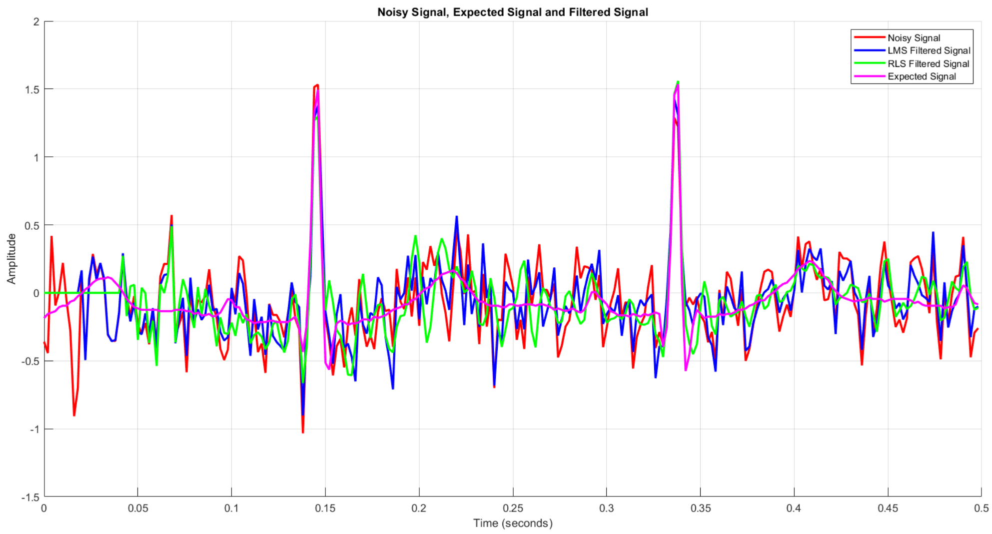

Figure 37: Filtering the noisy ECG signal by using RLS and LMS algorithms.

Here we can see, that RLS has performed quite well than the LMS.