Department of Electronic & Telecommunication Engineering

University of Moratuwa

BM4151 : Biosignal processing

MATLAB Assignment 2 – Wiener and adaptive filtering

Name	**Index number**
D.S Weerapperuma	180685C

This is submitted as a partial fulfillment for the module BM4151 Biosignal Processing Department of Electronic and Telecommunication Engineering University of Moratuwa

Date of Submission:

27/12/2022

Table of contents

Table of Figures

1 Weiner filtering (on stationary signals)

Weiner filters are a collection of optimum filters in comparison to traditional FIR and IIR filters, which provide the optimum filter of a particular order taking into account the statistical properties of the signals. By reducing the mean square error between the desired signal to be obtained and the filtered signal, optimization is carried out. Weights for Weiner filters are determined using the Weiner-Hopf equation.

$$\mathrm{W}_0 = \phi_{\mathrm{X}}^{-1}\Theta_{\mathrm{Xy}}$$

In the above equation, ϕ_x is the autocorrelation of the sampled signal and the Θ_{Xy} is the cross-correlation between the desired signal and sampled signal which include the noise. The derivation of this equation is based on several assumptions which are as follows.

- Signal and noise processes are independent.
- Signal and noise are stationary.
- The desired signal is known.
- Noise characteristics are known.

With the above assumptions, we can derive the following equation for noise removal.

$$\mathrm{W}_0 = (\phi_Y + \phi_N)^{-1}\Theta_{\mathrm{Xy}}$$

1.1 Discrete-time domain implementation of the Wiener filter

The time domain representation of the ideal ECG and the derived noisy signal is shown in the following figures.

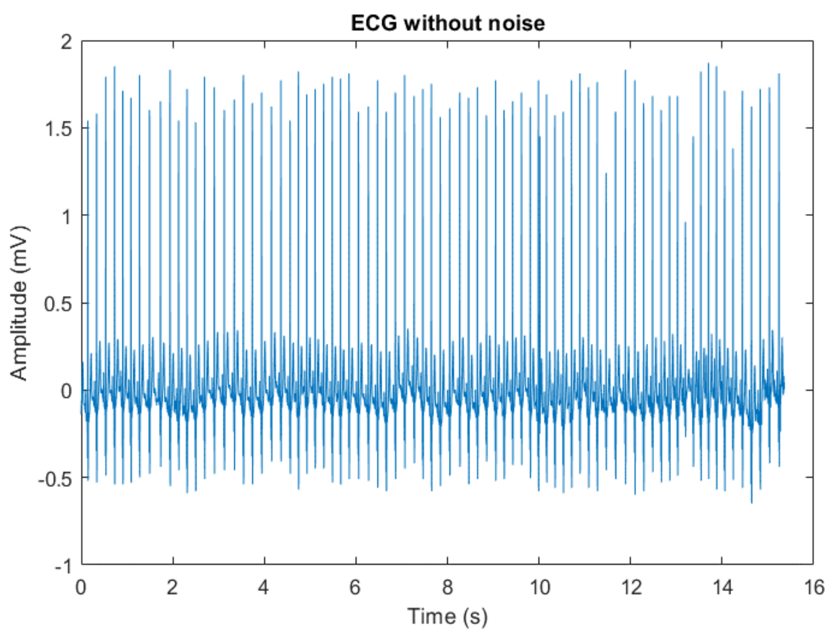

Figure 1.1 Ideal ECG

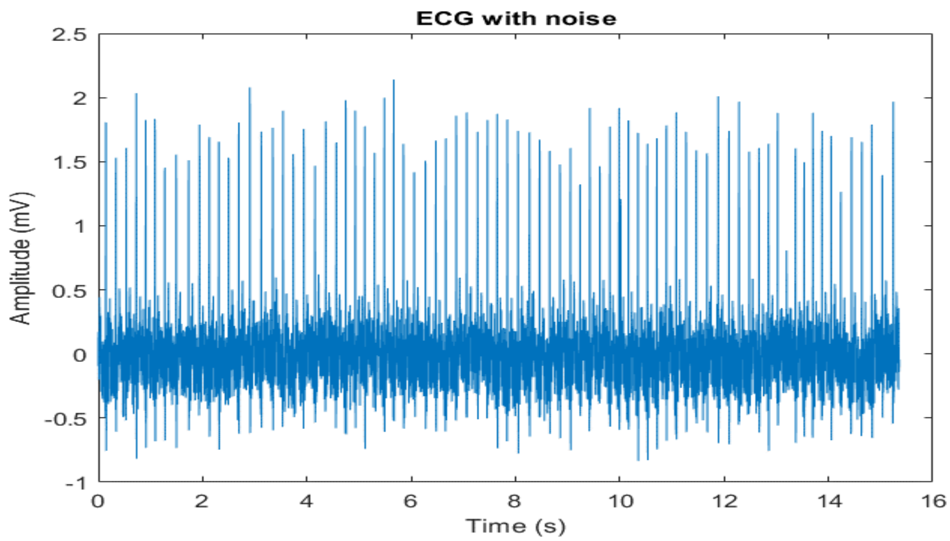

Figure 1.2 ECG with noise

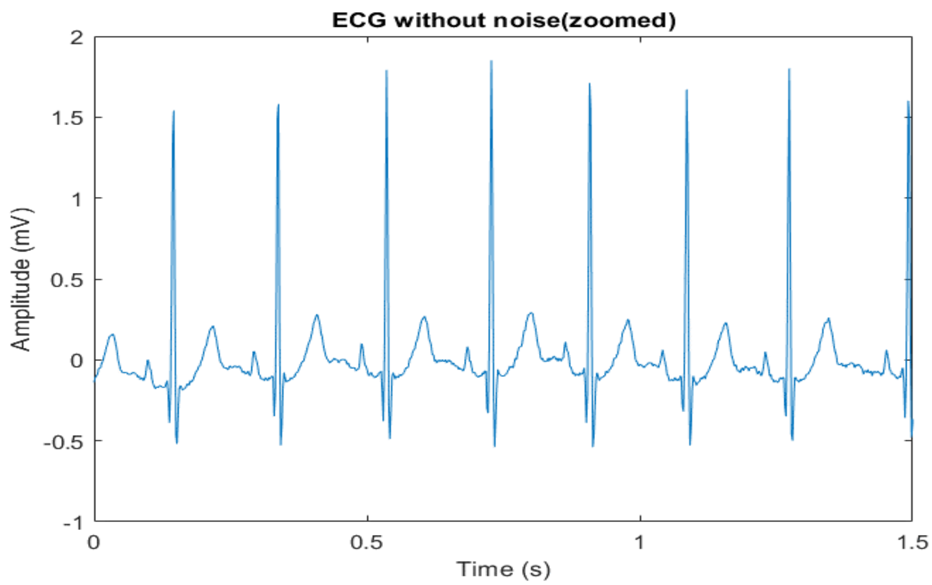

Figure 1.3 Ideal ECG (zoomed)

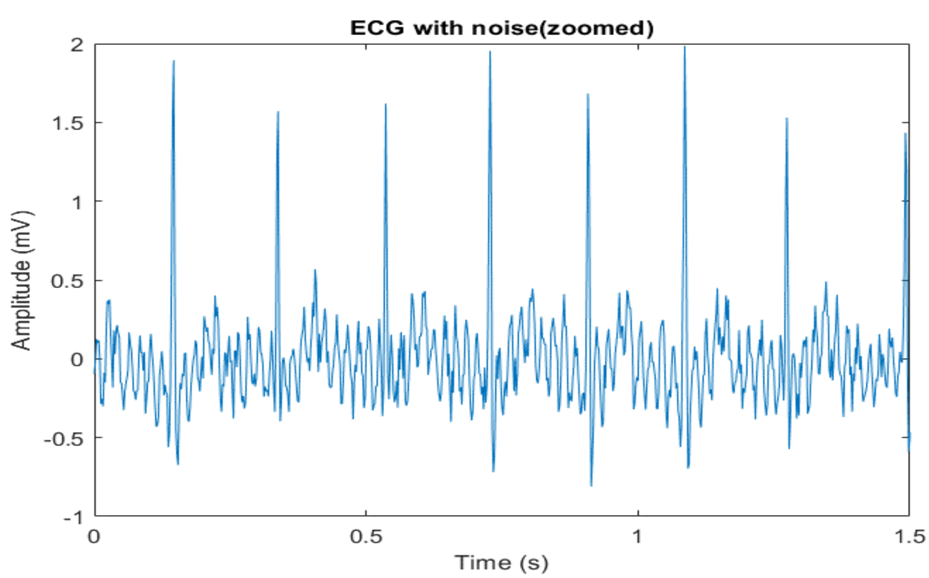

Figure 1.4 ECG with noise (Zoomed)

Part 1

In this case, we selected a single ECG beat for the desired signal which is located between 132 and 223 sample points. So that it has 92 sample points. For the Noise component, we extract a signal segment from the T wave to the P wave (121 to 143) of the next ECG beat (isoelectric segment) of $x(n)$. So that the length of the noise component is ¼ the of the desired signal. Therefore it will replicate 4 times to match the single beat. This selected beat and the noise components are shown in the following figures.

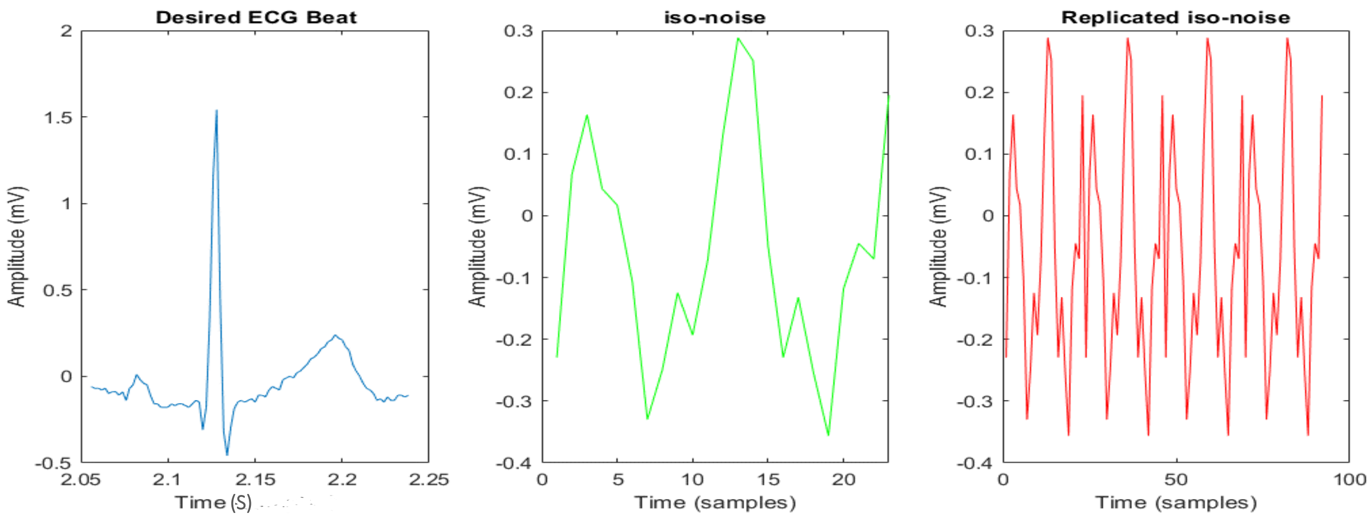

Figure 1.5 Desired ECG beat and noise components

- For arbitrary filter order case

Order = 12

W_0	W_1	W_2	W_3	W_4	W_5	W_6	W_7	W_8	W_9	W_{10}	W_{11}
0.6909	0.3198	-0.0481	-0.1225	0.0088	0.1089	0.1335	0.0783	0.0268	0.0446	0.0401	0.0196

The filtered signal and the noise signals are shown in the following figure.

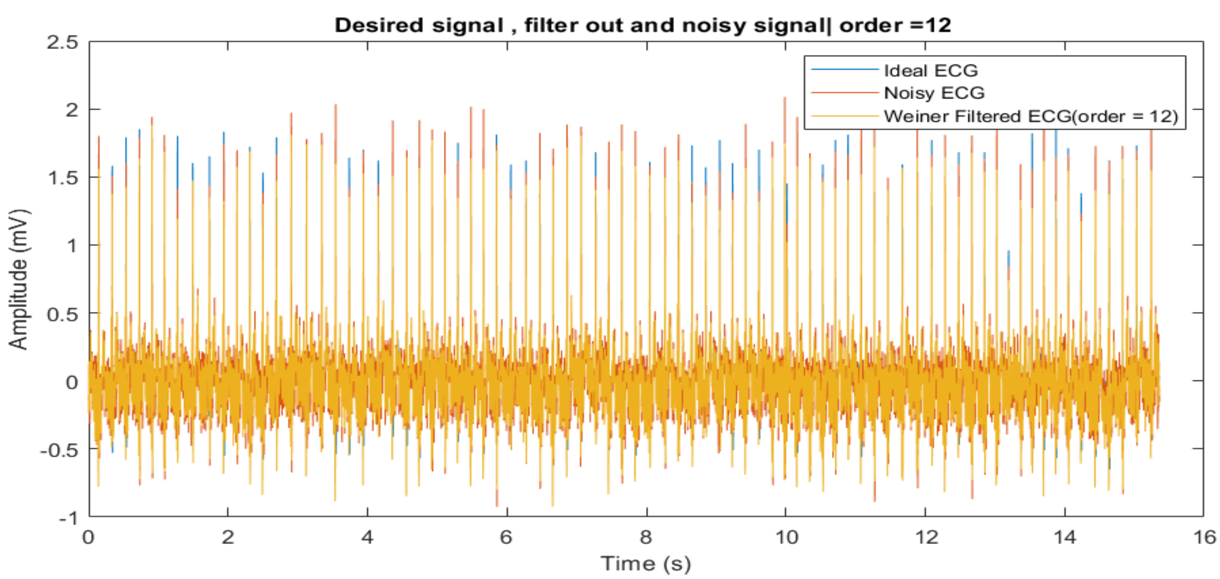

Figure 1.6 Weiner filter output (M = 12)

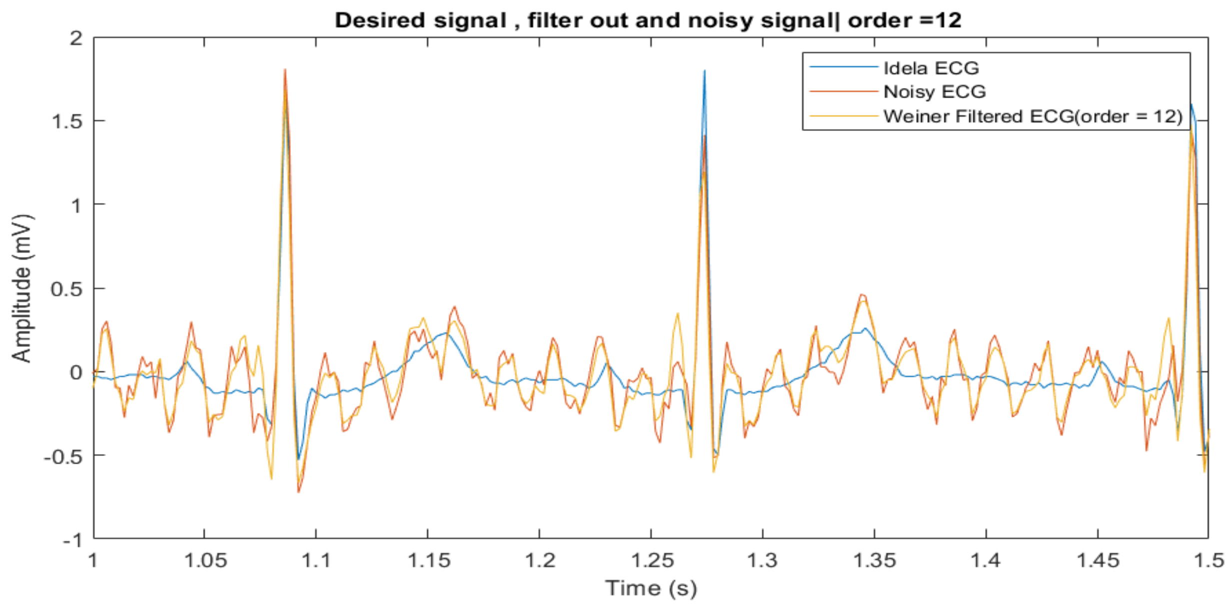

Figure 1.7 Weiner filter output zoomed (M = 12)

We can find the optimum filter order by calculating the mean square error (MSE) at each filter order of the given range and finding the corresponding order which gives the lowest MSE. The distribution of MSE values with their filter orders is shown in the below figure. In this case, we considered the range of 2 - 50 for the filter orders.

Optimum order : 8

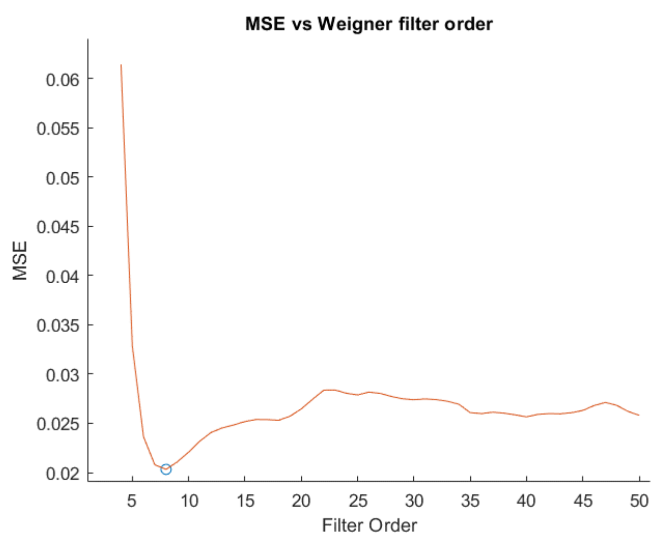

Figure 1.8 MSE vs Weiner filter order

W_0	W_1	W_2	W_3	W_4	W_5	W_6	W_7
0.6900	0.3327	-0.0063	-0.0754	0.0927	0.1856	0.1295	0.0695

According to figure 1.8, we can see that filter order 8 is the optimum filter order since it gives the lowest MSE compared to the other orders.

The following figure shows the magnitude and phase responses of the filter which has optimum filter order.

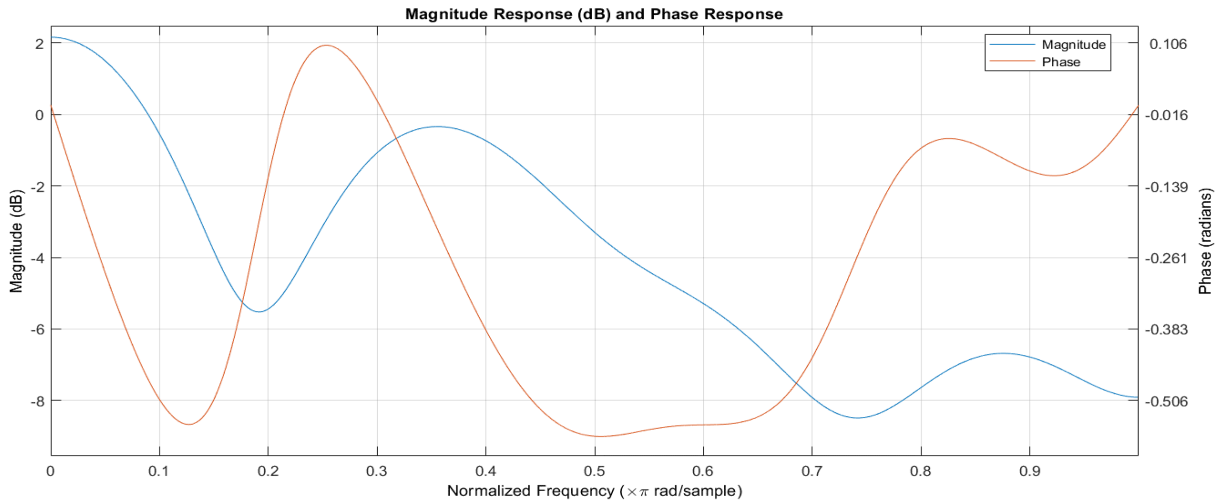

Figure 1.9 phase and magnitude responses of the Weiner filter with optimum order

We applied that optimum filter order-based filter to the noisy signal and the following figure shows the time domain and frequency domain representations of the resultant signal with the desired signal.

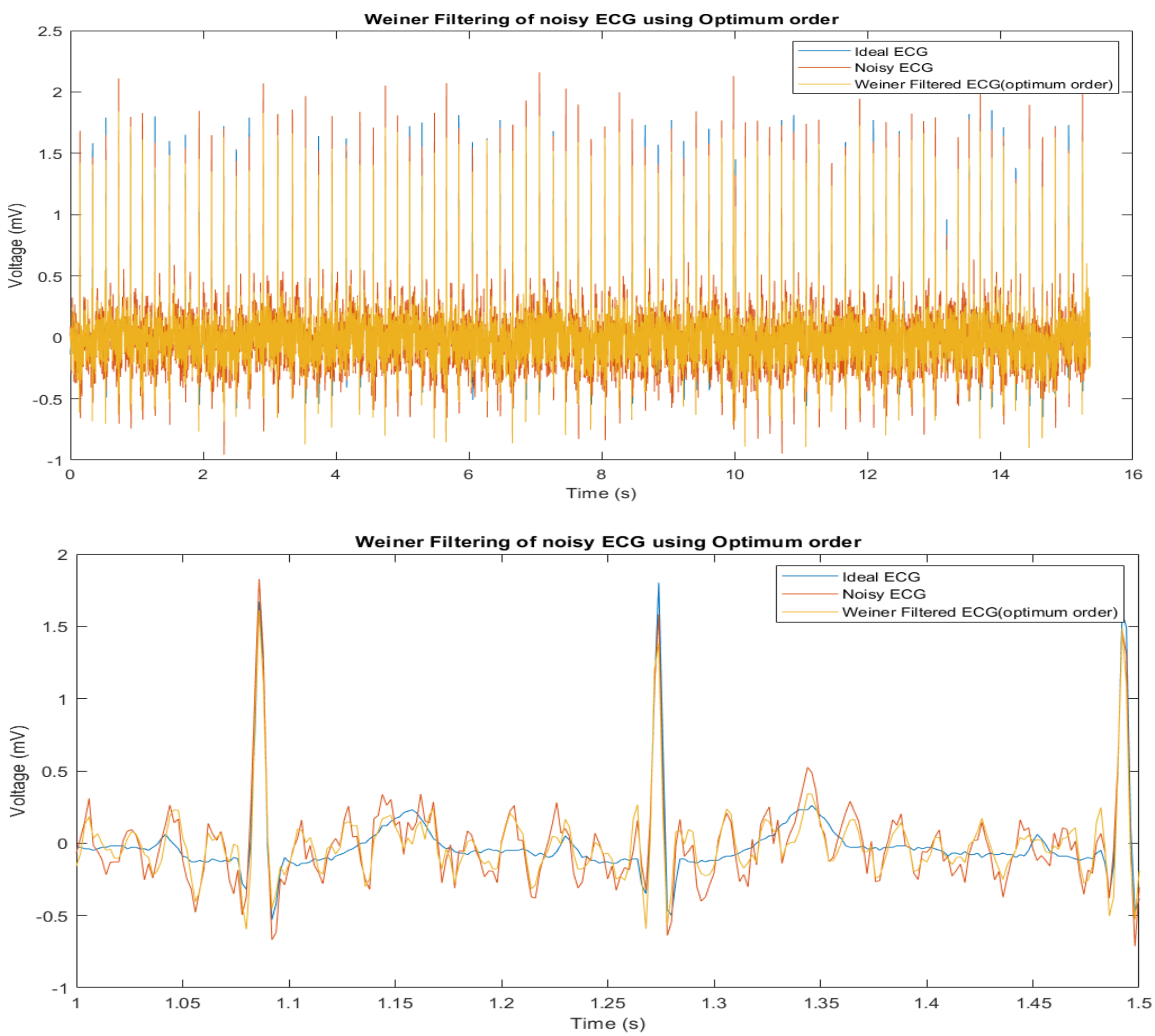

Figure 1.10 Filtered signal of optimum order Weiner filter(part 1)

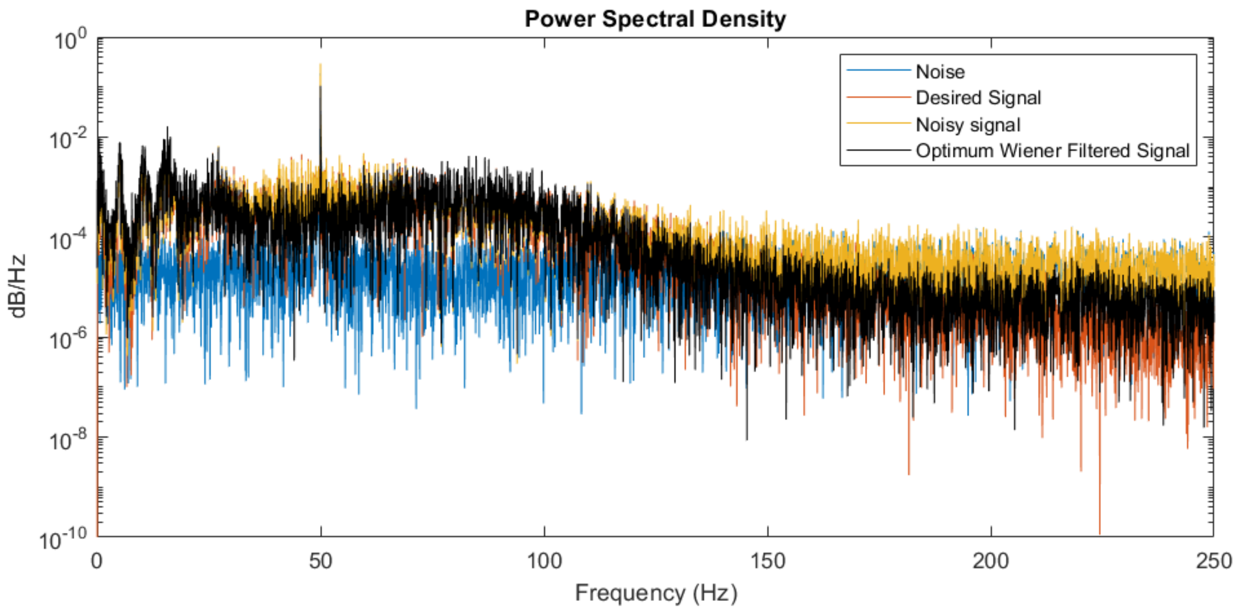

Figure 1.11 PSD of input and output signals of optimum order Weiner filter(part 1)

According to figure 1.10, we can clarify that the Weiner filter which has optimum order mostly worked on the high-frequency noise component compared to the low frequencies. It is clear that it can't remove the most of the low-frequency noises in this case. For example 50Hz power line noise is still there, and the filter couldn't remove that part.

Part 2

In this case, we consider a linear model having the same length and a comparable morphology to that of a single ECG beat of the ideal signal $y_i\,(n)$. This is more realistic because it is hard to find an ideal ECG signal. Since we know the basic characteristics of the ECG signal, we can use a linear model as the desired signal for the filtering process. The constructed linear model of the ECG signal is shown in the below figure. For the noise component, we use the same noise which use in the previous case.

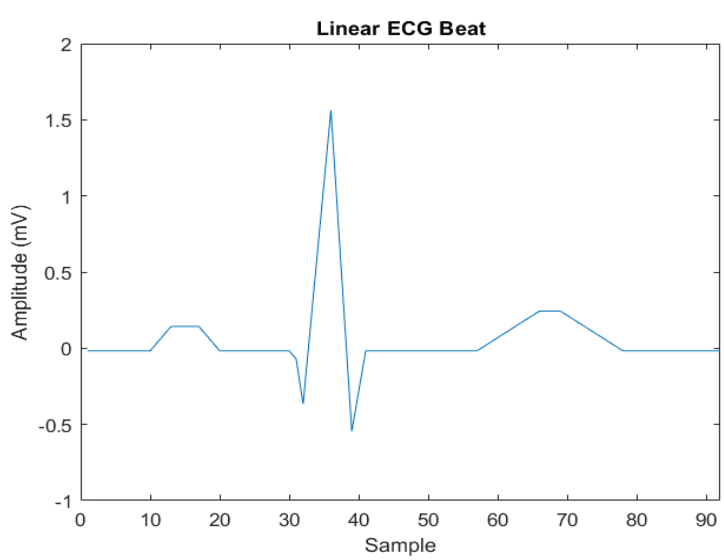

Figure 1.12 Linear model of ECG Beat

- For arbitrary filter order case

Order = 12

W_0	W_1	W_2	W_3	W_4	W_5	W_6	W_7	W_8	W_9	W_{10}	W_{11}
0.5915	0.2575	0.0362	-0.0847	-0.0666	0.0616	0.0825	0.1040	0.0311	0.0026	0.0252	0.0314

The filtered signal and the noise signals are shown in the following figure

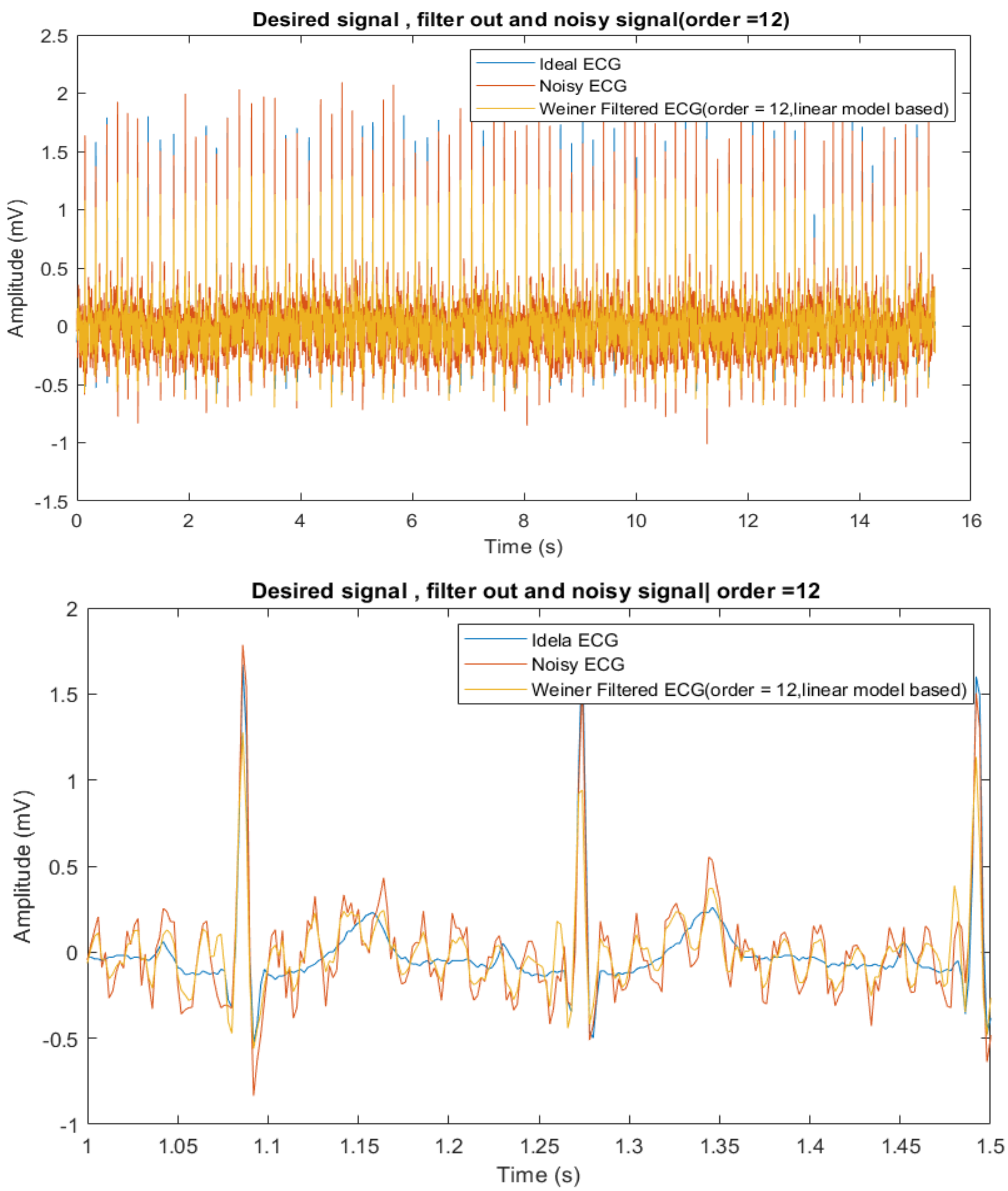

Figure 1.13 Filtered signal of optimum order Weiner filter(part 2)

As same as the previous case, we found the optimum filter order which gives minimum MSE and applied that order to filter the noisy signal. The resultant signals in the time domain and frequency domain and the filter characteristics are shown in the following figures.

Optimum order : 62

Figure 1.14 MSE vs Weiner filter order (linear model

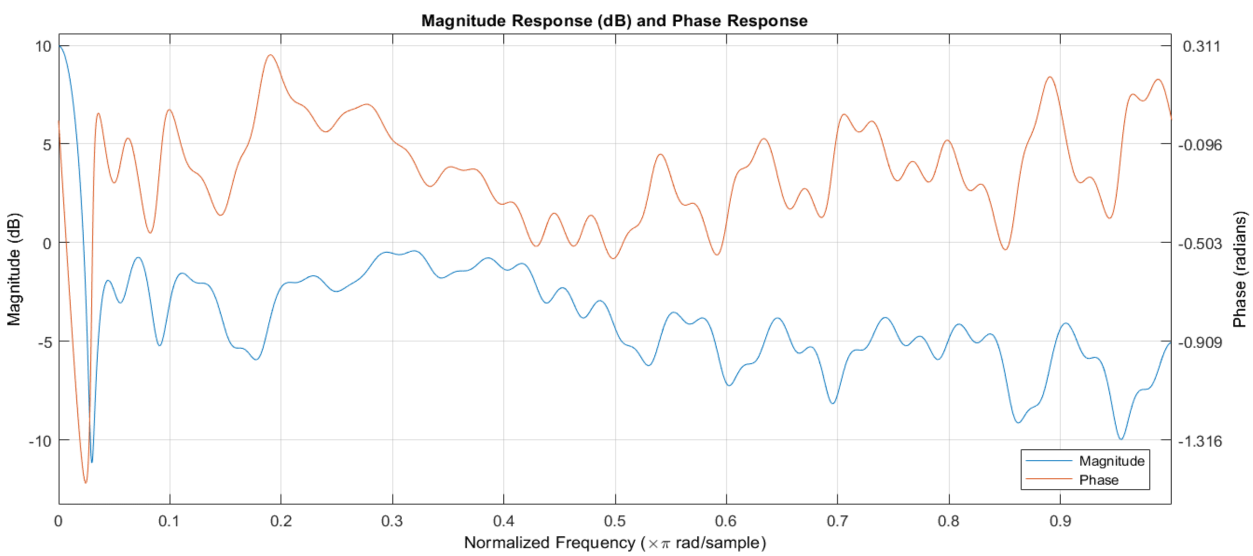

Figure 1.15 Phase and magnitude responses of optimum order Weiner filter

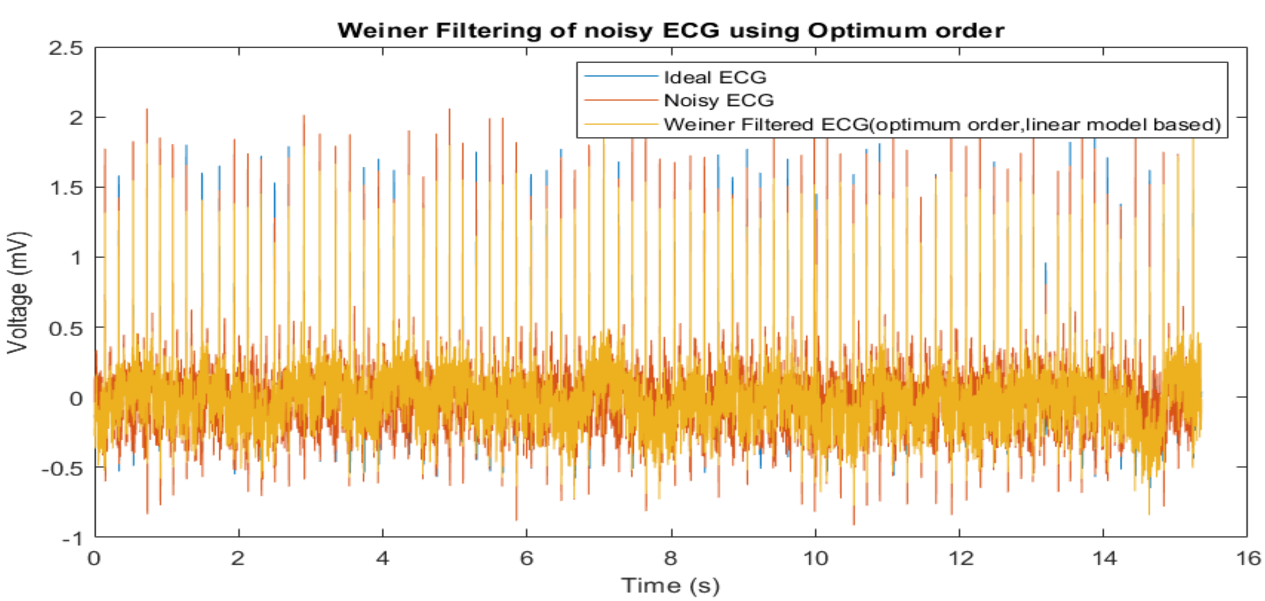

Figure 1.16 Filtered signal of optimum order Weiner filter (linear model)

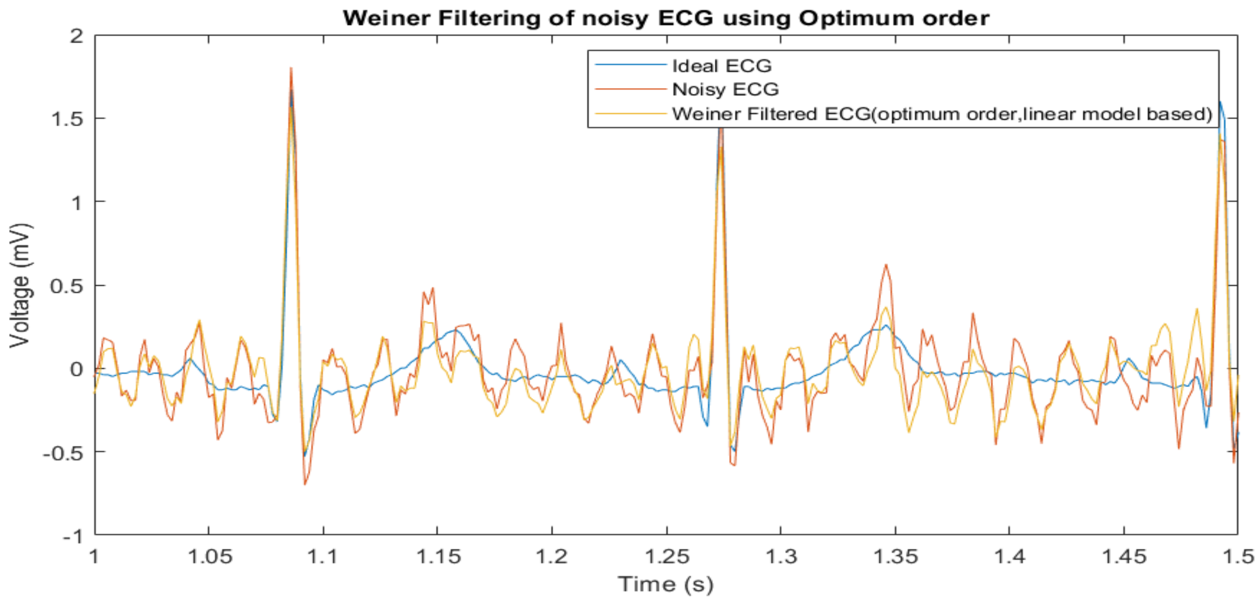

Figure 1.17 Filtered signal of optimum order Weiner filter (linear model-based, zoomed)

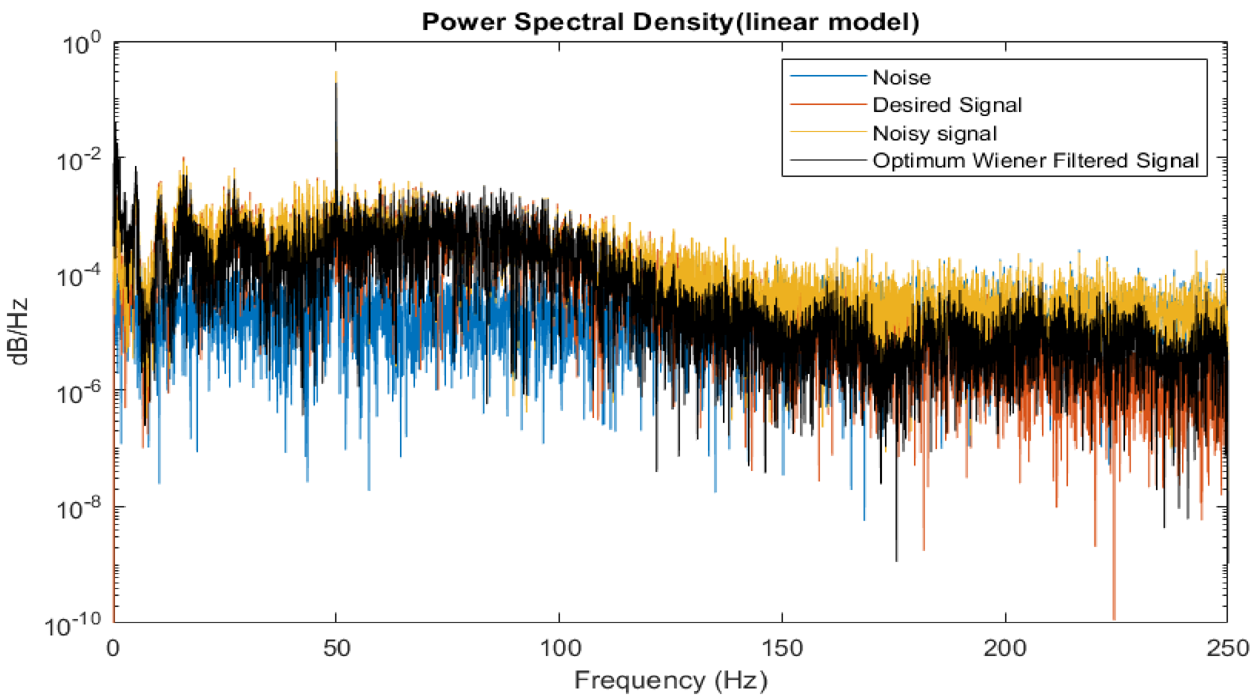

Figure 1.18 PSD of the filtered signal of optimum order Weiner filter (linear model based)

In this case, we can see that the wiener filter of optimum order which performing the same as in the previous case (part 1). So that it couldn't remove most of the low-frequency noises as same as in the previous case (Power line noise is still there).

1.2 Frequency domain implementation of the Wiener filter

The frequency domain optimum wiener filter is shown as follows.

$$W(f) = \frac{S_{YY}(f)}{S_{YY}(f) + S_{NN}(f)}$$

$Szz(f)$ is the PSD of a signal $z(n)$. Alternatively, $Szz(f)$ is the power of the Fourier transform of the template $z(n)$. Therefore, $Szz(f) = (absoulte(Fourier\ transform[z(n)]))^2$. In this case, $S_{YY}(f)$ is the PSD of the desired signal and $S_{NN}(f)$ is the PSD of noise.

We implemented the frequency domain optimum wiener filter using the above equation and filtered the signal mentioned in part 1. The representation of results in time and frequency domains and the comparison plots are shown in the figures below.

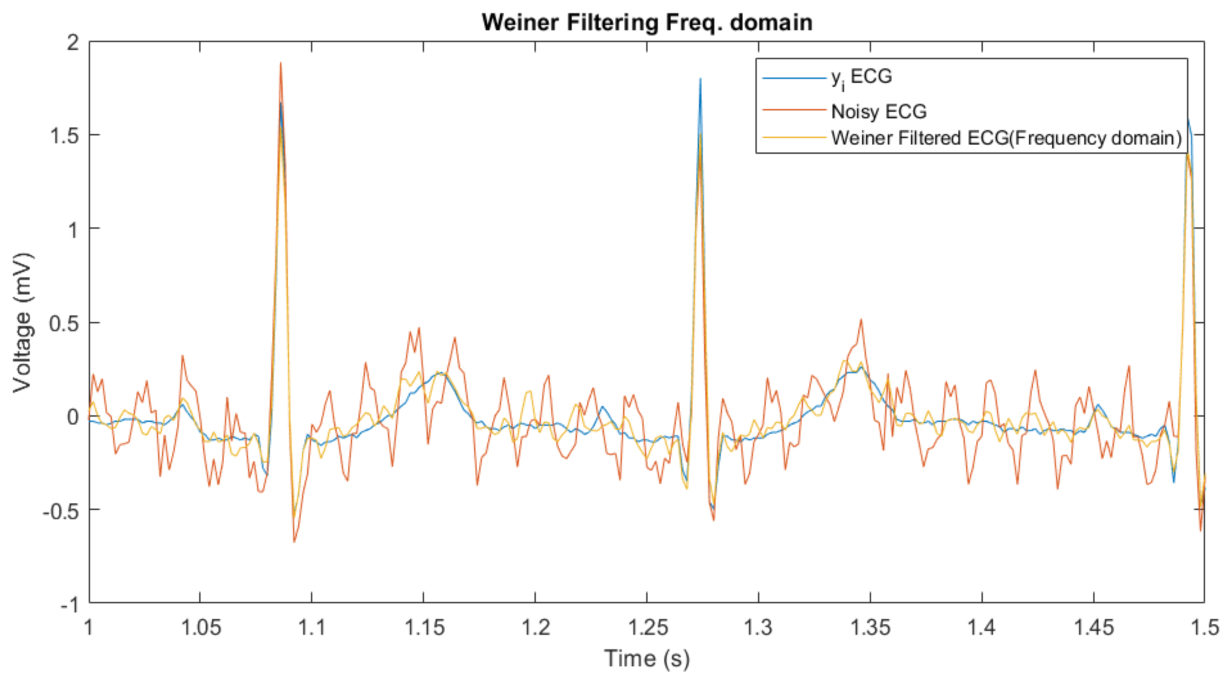

Figure 1.19 Weiner filter frequency domain implementation outputs

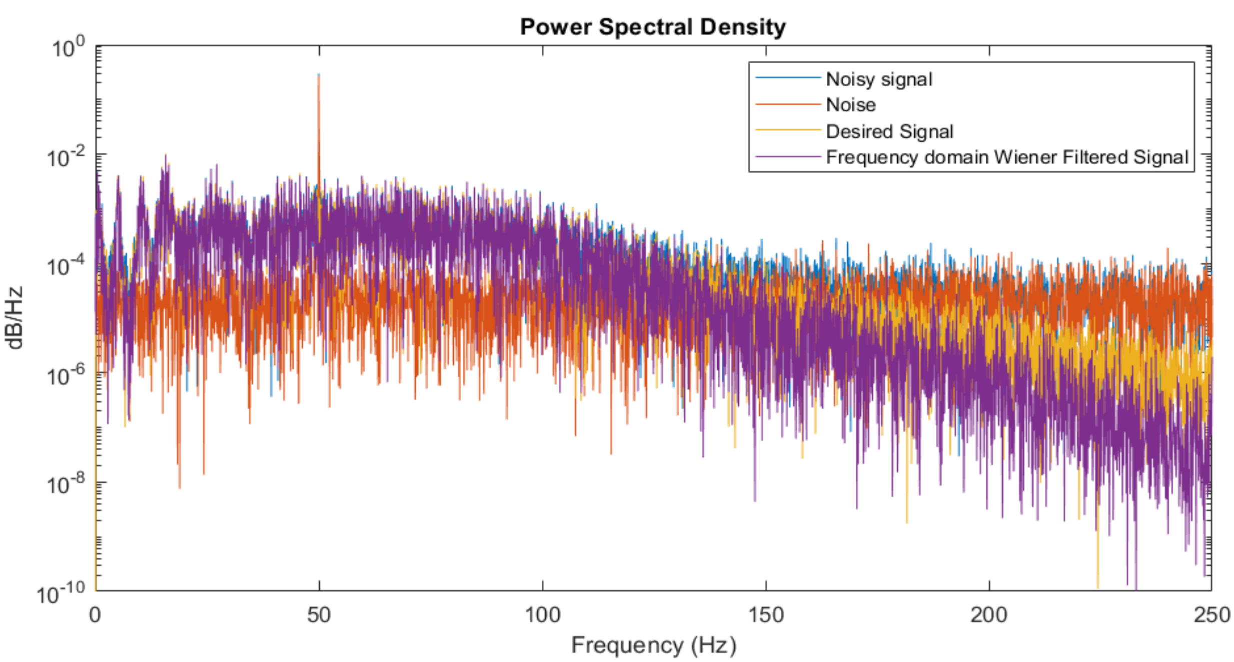

Figure 1.20 PSD of the filtered signal of Weiner filter freq. domain

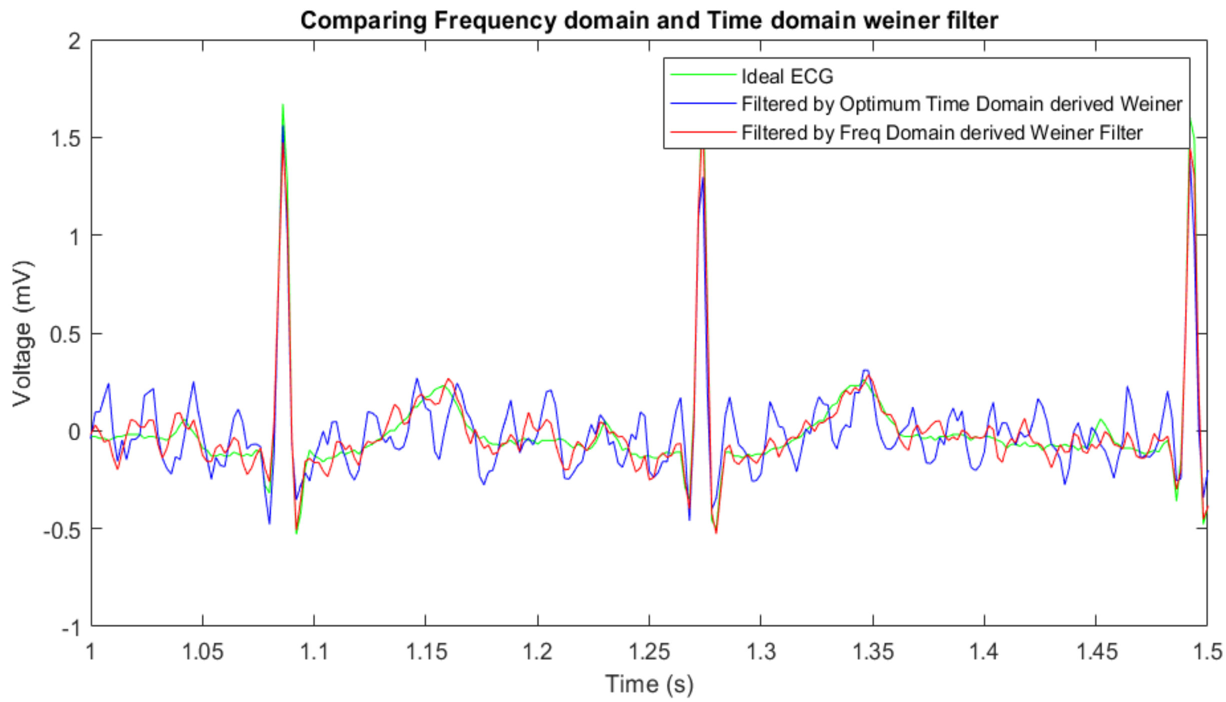

Figure 1.21 Filtered signals from the time domain and freq. domain implementations

Mean Square Error (Time domain, optimum order) : 0.0217

Mean Square Error (Frequency domain) : 0.0033

When comparing with the time domain implemetation, we can see that the frequency domain implementation of the Wiener filter performs very well since the output signal of the frequency domain implementation is more close to the desired signal compared to the output of the time domain implementation. We can further clarify that by observing the PSD of the output signal of the frequency implementation of the optimum wiener filter because the Frequency domain Weiner Filter is able to eliminate both high- and low-frequency noise from the noisy signal. So it removes the power line noise as well. We can further justify this as the MSE value is lower in the case of the Frequency domain implemented filter compared to the other type.

1.3 Effect of non-stationary noise on the Wiener filtering

This section explores the behavior of the Wiener filter when the noise characteristics are nonstationary. This is demonstrated by changing the 50 Hz noise to 100 Hz halfway through the signal duration.

By observing figure 1.22 we can see that the optimum wiener filter (frequency domain implemented) was able to reduce the low-frequency noise, but it was unable to do so for the high-frequency noise that was added later. We can further clarify that by observing figure 1.23 there is a large spike at 100Hz and a very lower(almost non) spike at 50Hz. That implies the optimum wiener filter cannot handle non-stationary noises.

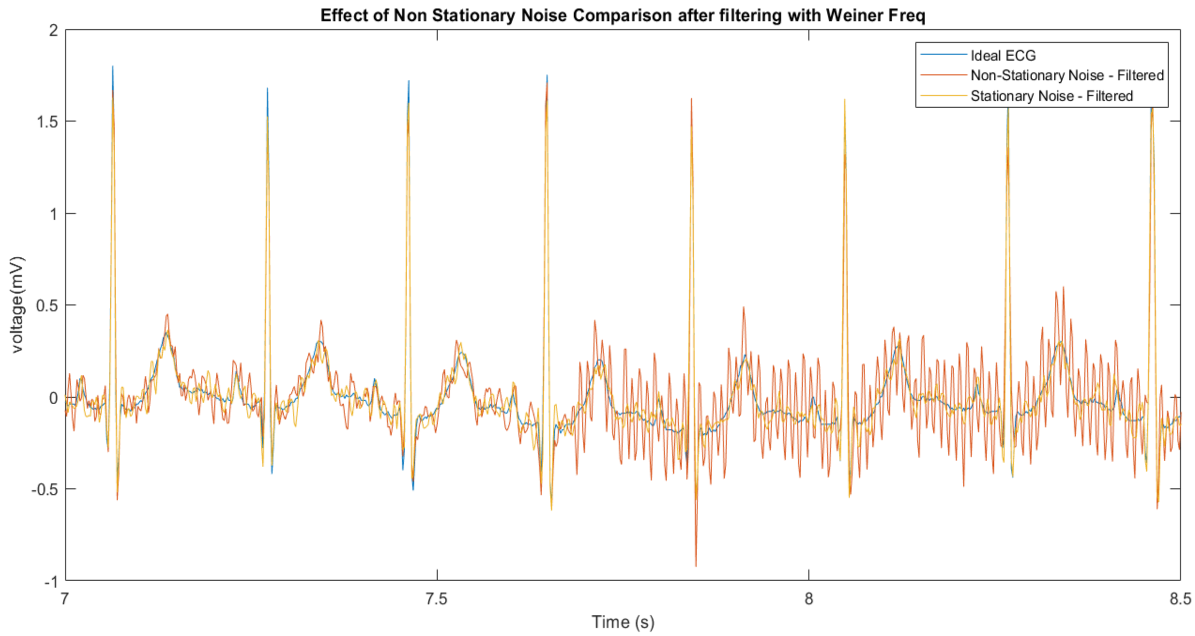

Figure 1.22 Comparison between filter responses for stationary and nonstationary noises

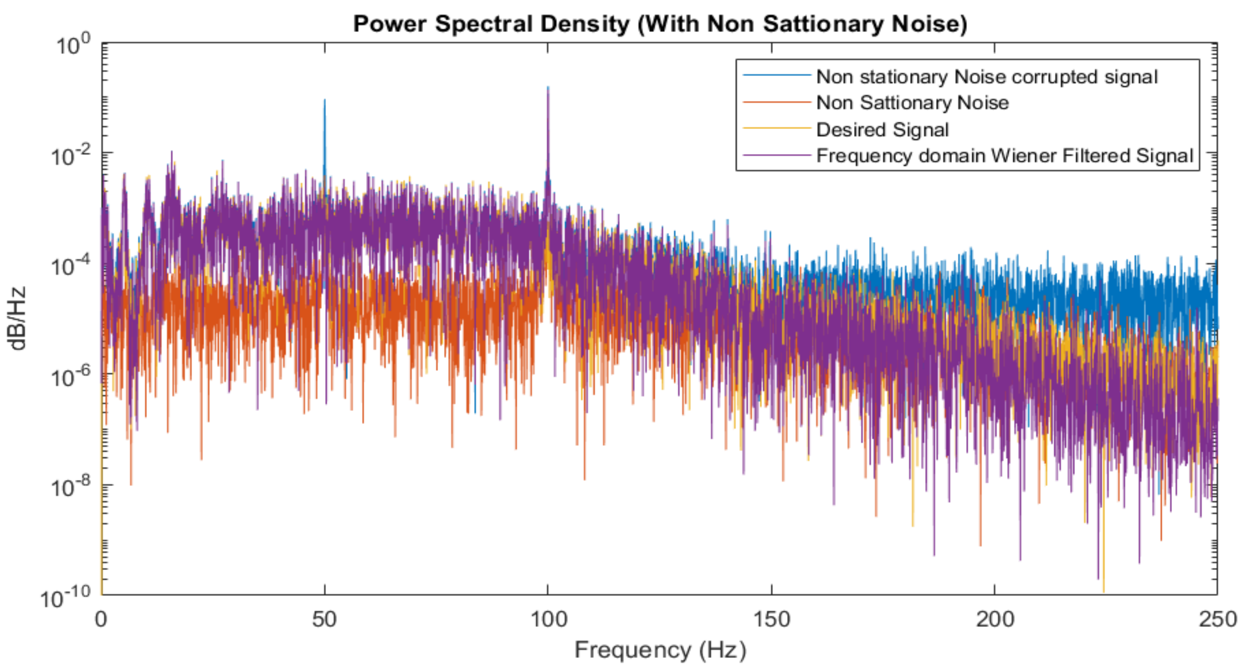

Figure 1.23 PSD of filter output for non-stationary noise

2 Adaptive filtering

Since conventional filters are based on the assumption of stationary, they can't be used the filter out the nonstationary signals. A possible solution is to use methods that can adapt according to the signal and noise variations. That is the main reason for the concept of Adaptive filtering. Adaptive filters use two distinct types of algorithms to implement the optimum filter since the Weiner-Hopf equation requires a lot of time to calculate the inverse of the matrix. The Least Mean Square (LMS) algorithm and the Recursive Least Square (RLS) algorithm are those algorithms.

We used a Sawtooth waveform with a width of 0.5 and nonstationary noise with it. Those input waveforms are shown below figures. To generate the noise signal we use the following values gains and phases.

$$a = 1.987, \phi_1 = \frac{\pi}{3}, \phi_2 = \frac{\pi}{4}$$

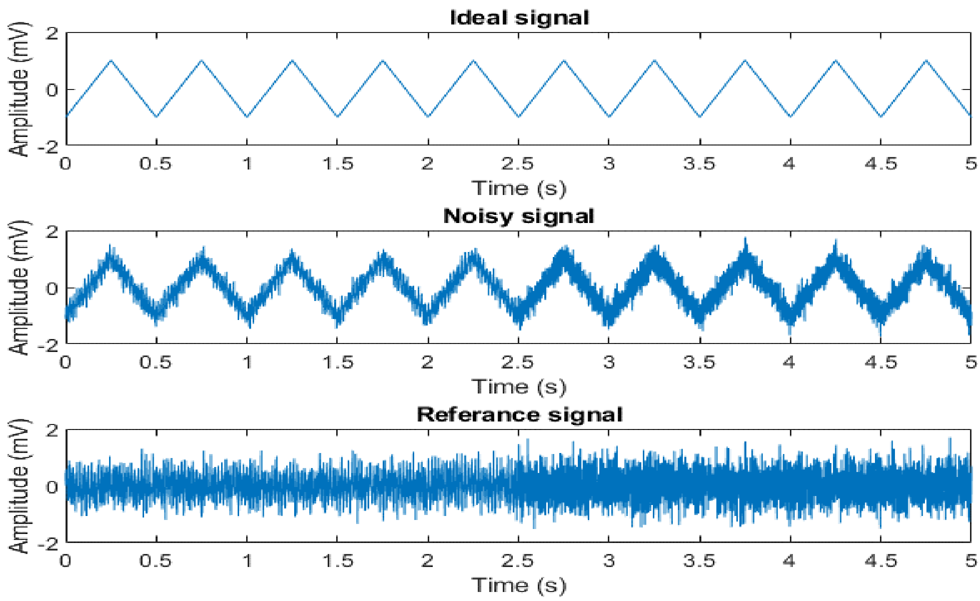

Figure 2.1 Input signals and their components

2.1 LMS method

We use the following equation to implement the LMS algorithm.

$$\mathrm{w(n + 1) = w(n) + 2\mu e(n)R}^T\mathrm{(n)}$$

To explore the rate of adaptation we change the rate of convergence μ and the order of the adaptive filter Since the filter depends on both order and the rate of convergence. To quantify the adaption, we calculated the mean squared error respective to the desired signal. The MSE values vs their respective orders and rate of convergence values are shown in the following figure.

optimum order : 18

optimum mu : 0.0040183

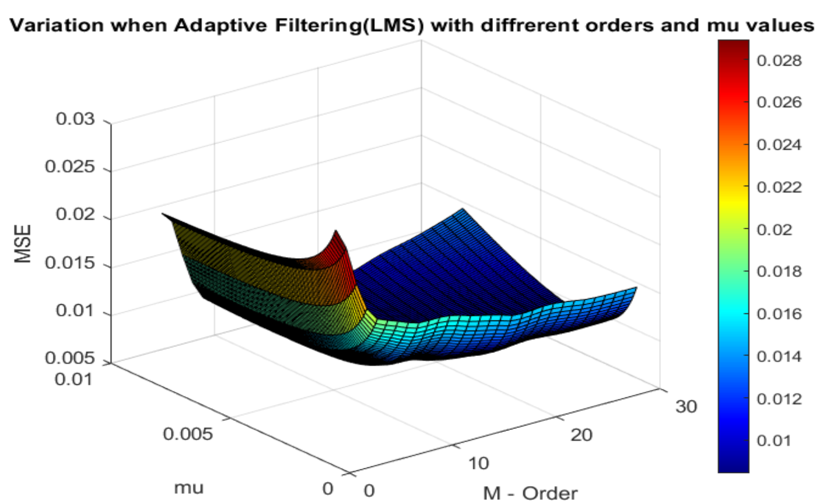

Figure 2.2 MSE vs LMS filter order and mu

The filtered signal corresponds to the optimum parameters shown in the below figure.

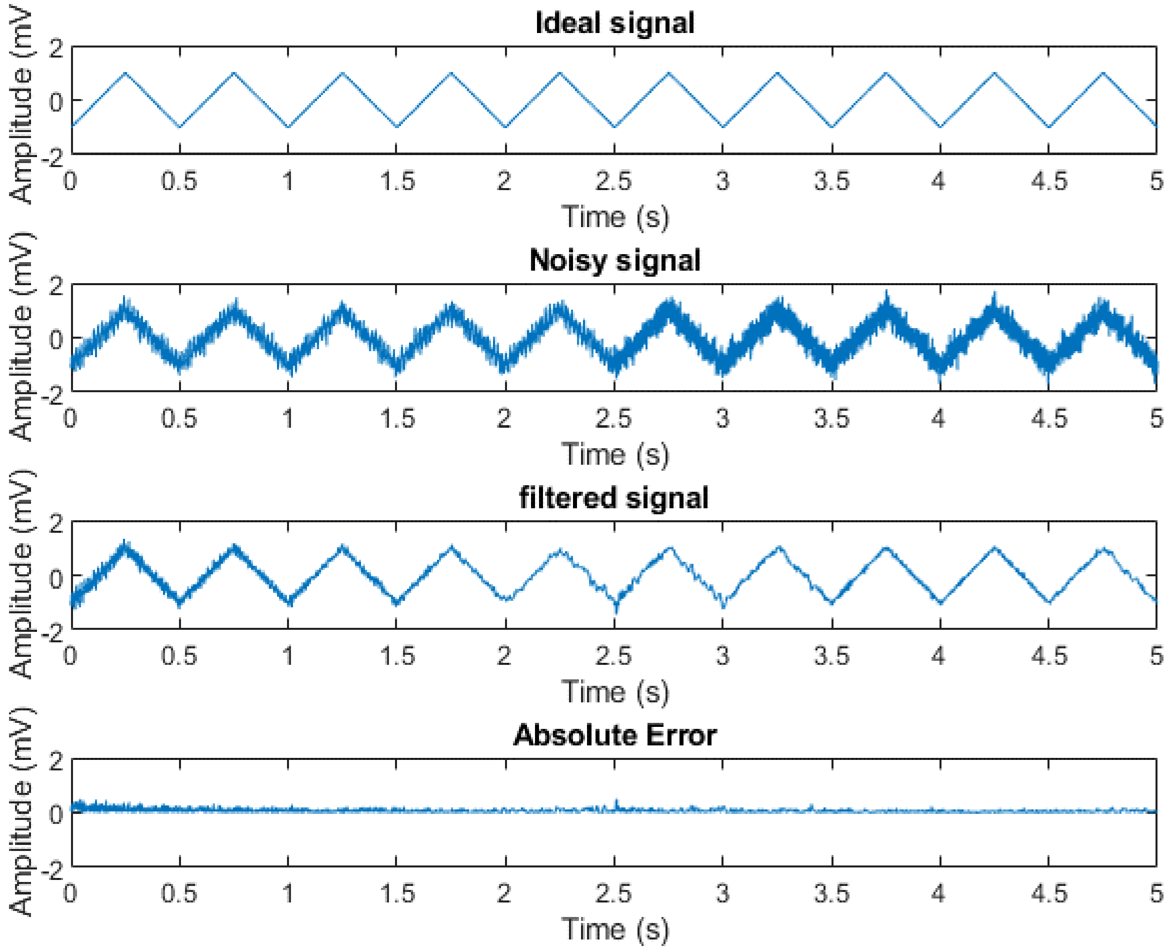

Figure 2.3 LMS filter output and input signals for optimum parameters

If the rate of convergence is too low then the algorithm takes a large time (use a large number of samples) to converge to the minimum error. So, for a given number of samples, the error is higher for the lower rate of convergence. The error rises for very large values of rate of convergence because the filter is unable to make small modifications and may not be able to converge. Also, for higher order number of iterations(steps) will be lower and for the lower order filter, it only depends on the recent samples, not on the large number of previous samples. In both cases, errors may be increased. Therefore, there is an optimum point that gives the best filter output.

2.2 RLS method

To explore the rate of adaptation we change the forgetting factor λ and the order of the adaptive filter Since the filter depends on the both order and the forgetting factor. To quantify the adaption we calculated the mean squared error respective to the desired signal. The MSE values vs their respective orders and forgetting factor values are shown in the following figure.

optimum order : 8

optimum lambda : 0.9979

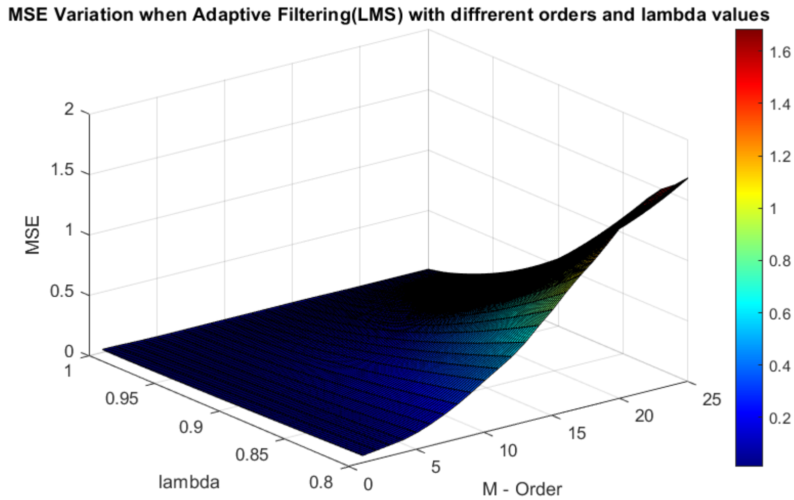

Figure 2.4 MSE vs RLS filter order and lambda

The filtered signal corresponds to the optimum parameters shown in the below figure.

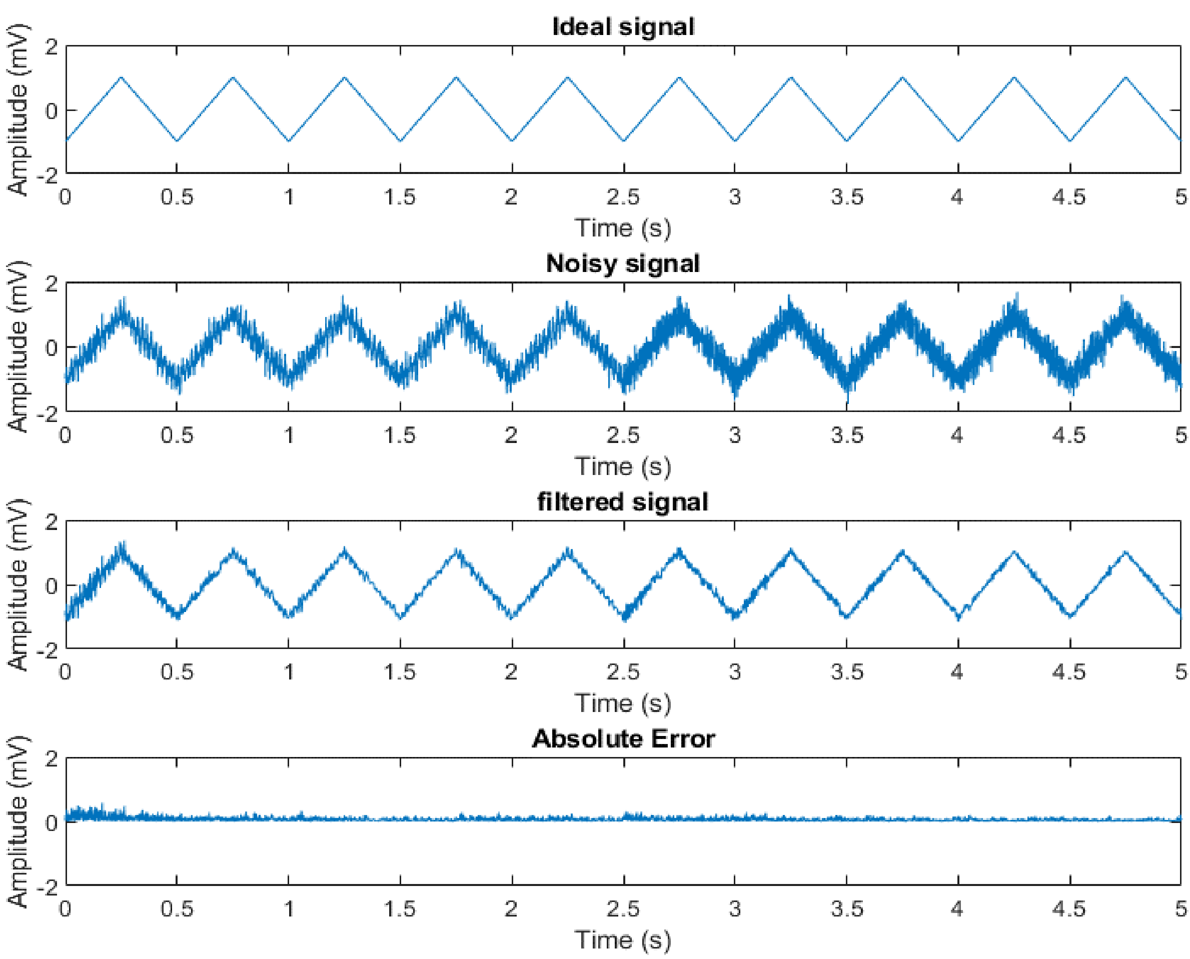

Figure 2.5 LMS filter output and input signals for optimum parameters

In order for the contribution from earlier repetitions to be significant, the value should be close to 1. Therefore, the optimum value of lambda is close to 1. According to the above figure, the lambda values should be closer to 1 for large filter orders to minimize the error.

Comparison of the performance of LMS and RLS algorithms

For comparison purposes, we use the same filter order(15) for both algorithms and the same input signal. The following figure shows the absolute error of the filtered signals compared to the desired signal.

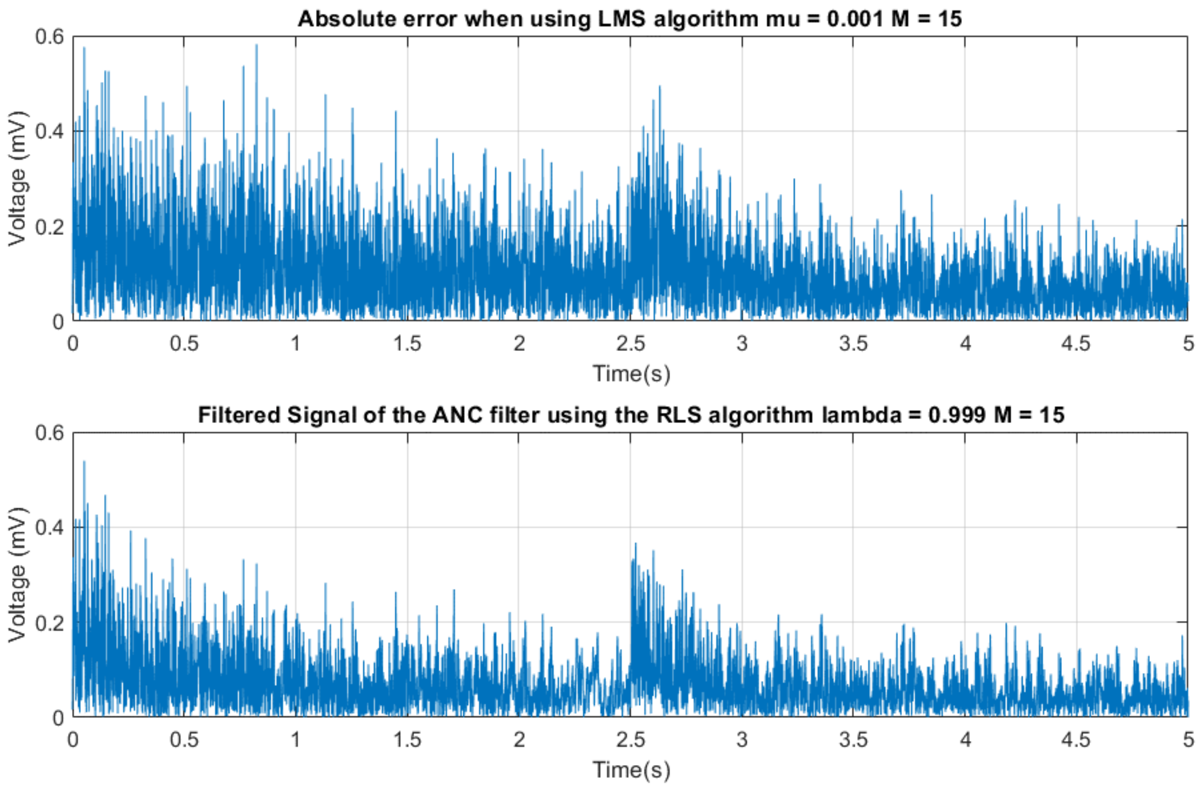

Figure 2.6 Absolute error comparison between LMS and RLS filter outputs

We can see that the RLS approach converges more quickly than the LMS algorithm by examining the absolute error graphs in figure 2.6 because absolute errors are reduced faster in RLS compared to LMS. So, the absolute errors of the RLS algorithm are lower than the absolute errors of the LMS algorithm through time.

Test LMS and RLS algorithms using the $yi(n)$ as the 'idealECG.mat'

We use the same noise used with the sawtooth signal to generate the noisy ECG signal.

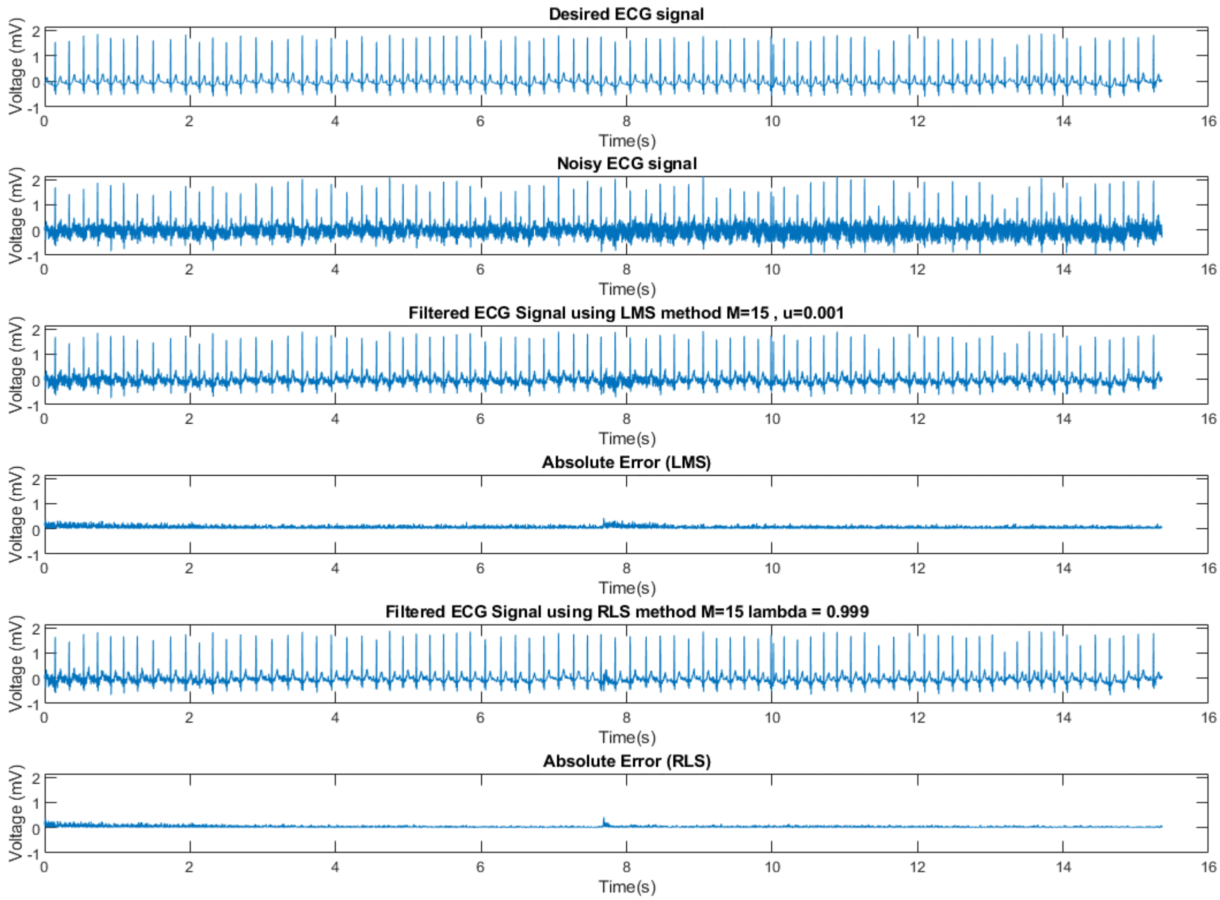

Figure 2.7 Filtered noisy ECG signals using LMS and RLS algorithms

By observing the above figure, we can clarify that both LMS and RLS algorithm-based adaptive filters are performing well in reducing the noise. But in this case, we can see that the filtered signal of the LMS algorithm is noisier than the RLS output and the LMS output contains higher frequency noise components compared to the RLS output. In contrast to the RLS method, the LMS algorithm was unable to suppress high-frequency noise components in this case, using the given number of samples and the given order.