Department of Electronic & Telecommunication Engineering
University of Moratuwa

BM4151 – Biosignal Processing

Assignment - II
Wiener and adaptive filtering

Name: Sahan Sulochana Hettiarachchi
Index: **180237G**

This is submitted as a partial fulfilment for the module BM4151 – Biosignal Processing

27th of December 2022

`

Table of Contents

`

`

1 Weiner filtering (on stationary signals)

Wiener filtering is a technique used to remove noise from a signal to improve its quality. The basic idea behind Wiener filtering is to minimize the mean squared error between the original signal and the filtered signal. This is achieved by adjusting the filter coefficients to minimize the difference between the two signals. The filter coefficients are determined using a set of data that includes both the original signal and the noise. The optimum Weiner filter can be obtained by the following equation.

$$W_0 = {\Phi_x}^{-1}\Theta_{xy}$$

In here Φ_x is the autocorrelation function of the input signal $x(n)$ and Θ_{xy} is the crosscorrelation function of the input signal $x(n)$ and the ideal signal $y(n)$. In Weiner filter designing we assumed that the signal component and noise component are independent of each other. Then the Weiner-Hopf equation can be reconstructed as follows.

$$W_0 = (\Phi_Y + \Phi_N)^{-1}\Theta_{xy}$$

Using the following equation filter coefficients can be calculated with the estimated signal and noise signal.

1.1 Discrete-time domain implementation of the Wiener filter

1.1.1 Data construction

Ideal signal: $y_i(n)$
Sampling frequency: f_s = 500Hz
Input signal: $x(n) = y(n) + \eta(n)$
Additive white gaussian noise that SNR 10dB: $\eta_{wg}(n)$
$\eta50(n) = 0.2\sin(2\pi f1n)$ where $f_1 = 50\ Hz$
Noise signal: $\eta(n) = \eta_{wg}(n) + \eta_{50}(n)$

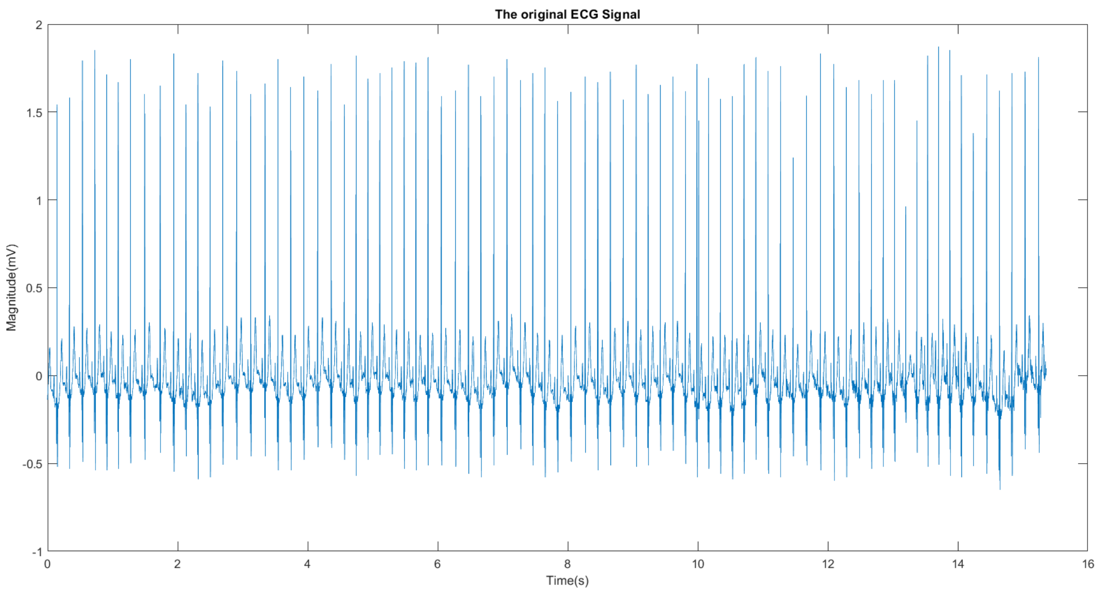

Figure 1: The Original ECG signal

`

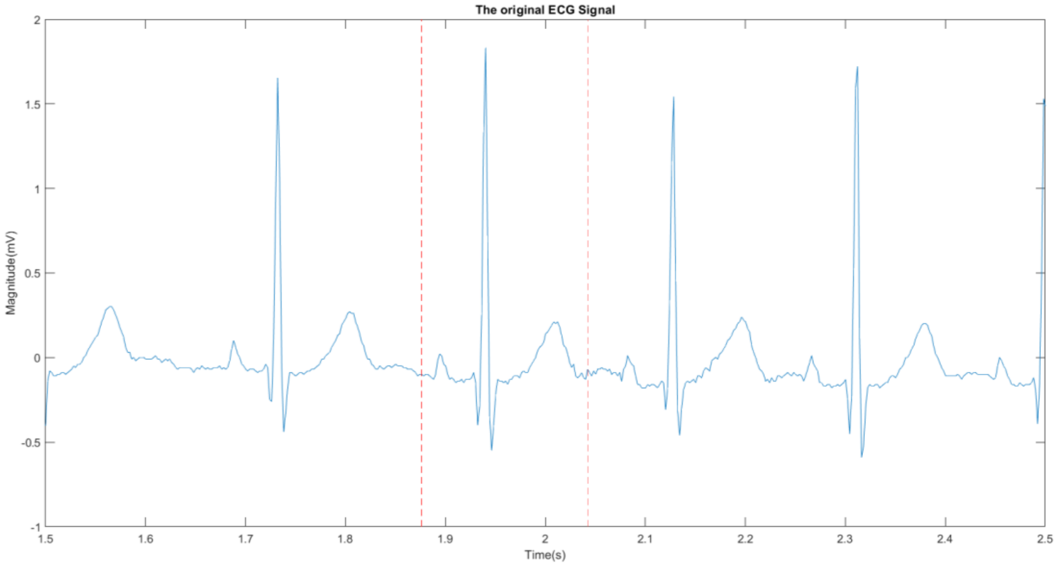

Figure 3: Zoomed ideal ECG signal

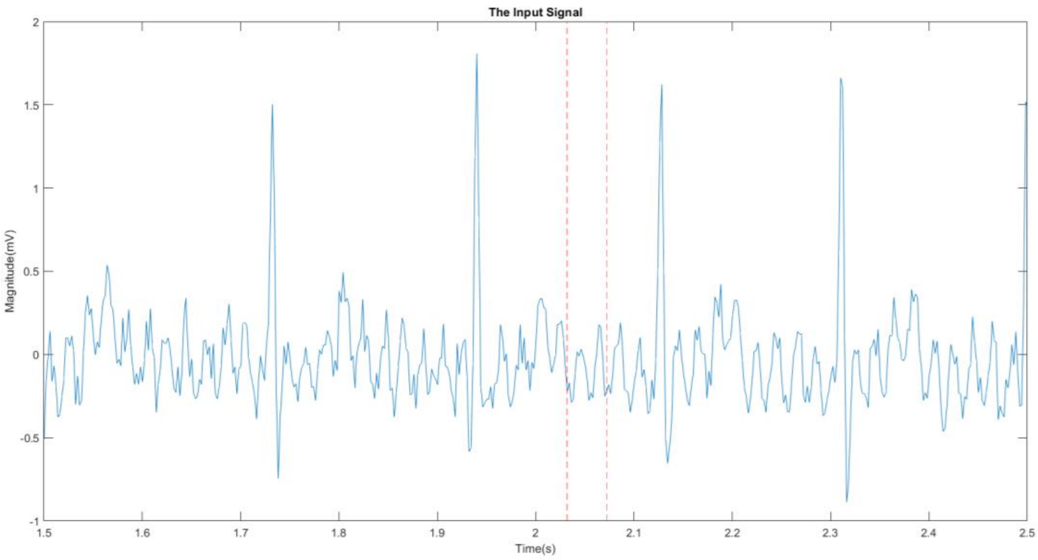

Figure 2: Zoomed noised added signal(Input signal)

1.1.2 Part I

For designing the wiener filter randomly selected ECG signal pulse is used. Figure 3 marked the randomly selected ECG signal pulse and Figure 4 shows the single ECG pulse. Also, a part of a noise-added signal was obtained. The ECG signal that is obtained has 84 sample lengths and the selected noise signal part (Figure 2) has 21 samples. Therefore, the obtained signal (isoelectric segment) concatenates the same signal four times. Thus the sample size signals can be obtained.

`

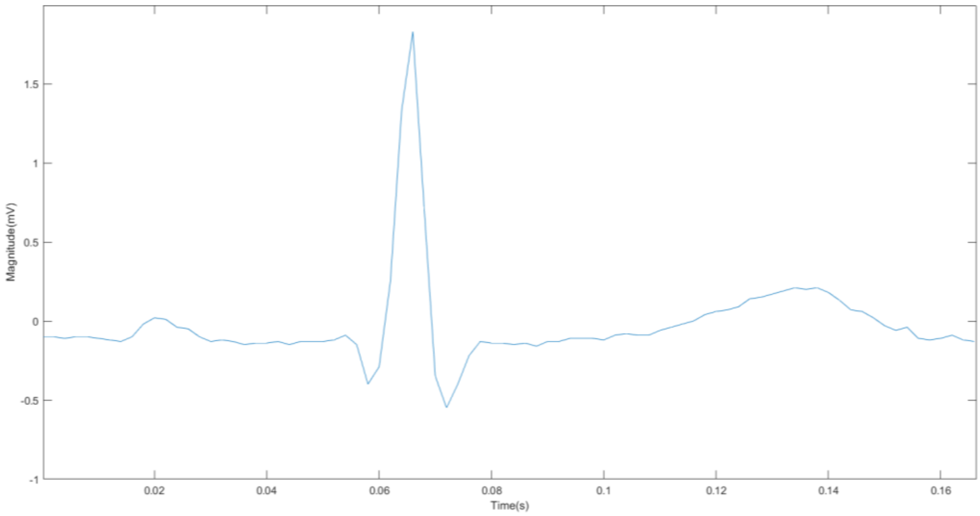

Figure 6: Concatenated isoelectric Segment

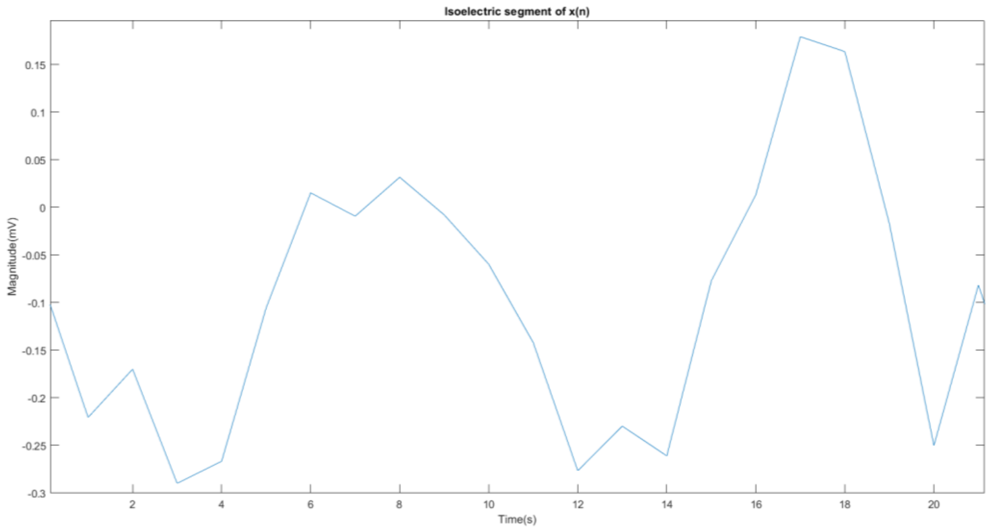

Figure 4: Isoelectric Segment

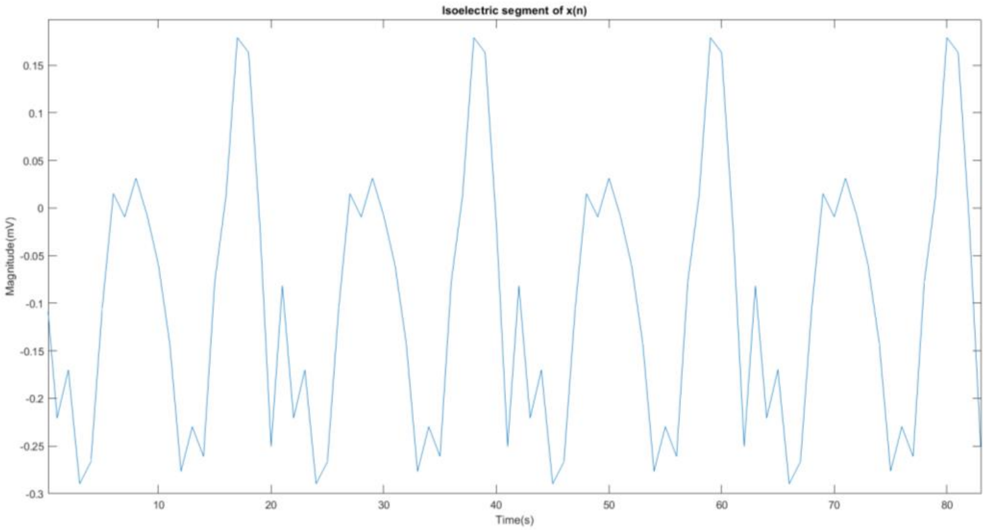

Figure 5: Concatenated isoelectric segment

`

Without a proper calculation, the optimum filter order cannot be obtained. Before calculating the optimum filter order let's take an arbitrary filter order (20). Table 1 shows the weight matrix for the selected filter order (20).

Table 1: Weight matrix for filter order 10

0.6712	0.3016	-0.0337	-0.0959	-0.0332	0.1103	0.1243	0.0739	0.0362	0.0079

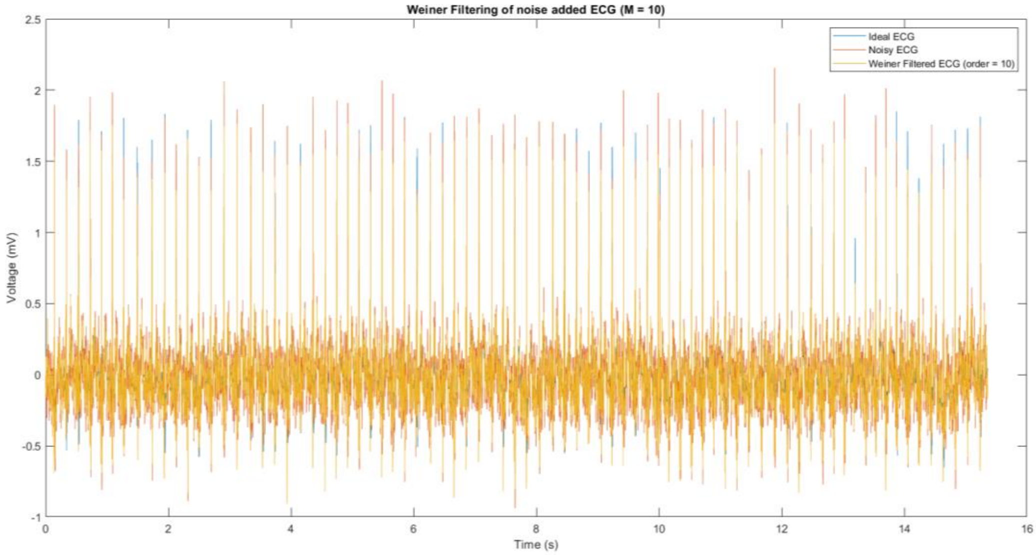

Figure 8: Ideal ECG, Noisy ECG and Weiner filter applied noise signal (order =10)

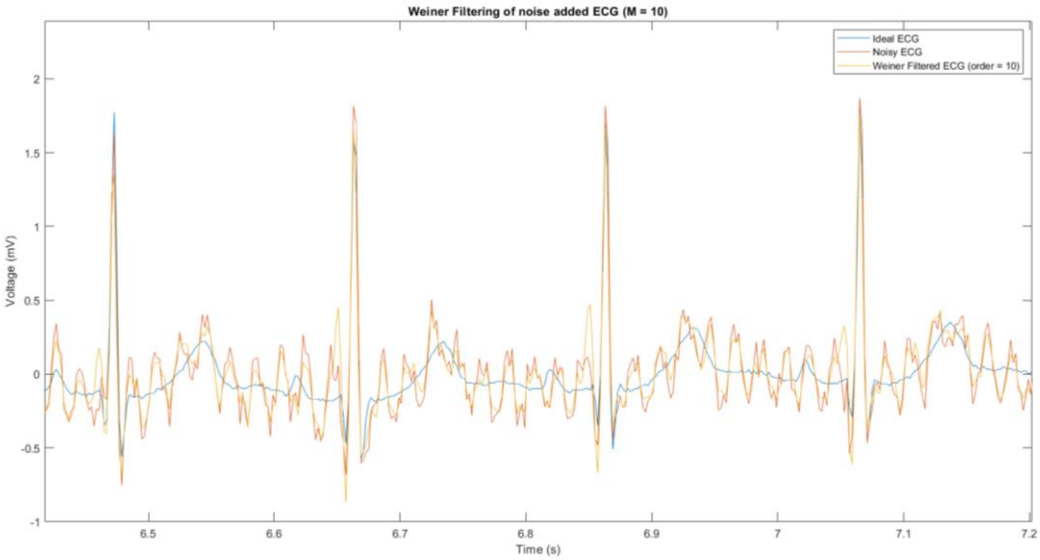

Figure 7: Zoomed preview(Ideal ECG, Noise ECG and filtered noise ECG)

For the optimum filter order calculation Mean Squared Error(MSE) calculation is utilized. Here 0-50 values are used as the order and each result is plotted into the graph. The mean square error for the following application is calculated by the below equation.

$$Mean\ Squared\ Error = E((y_i - \hat{y})^2)$$

`

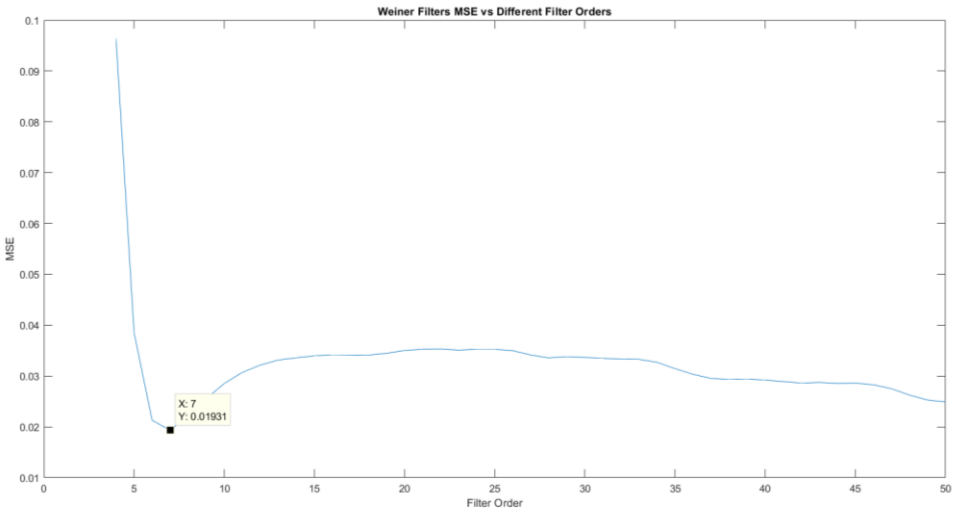

Figure 11: Weinier Filter MSE vs Filter Order

According to the above figure, the optimum filter order is 7 as it claims the lowest mean square error value.

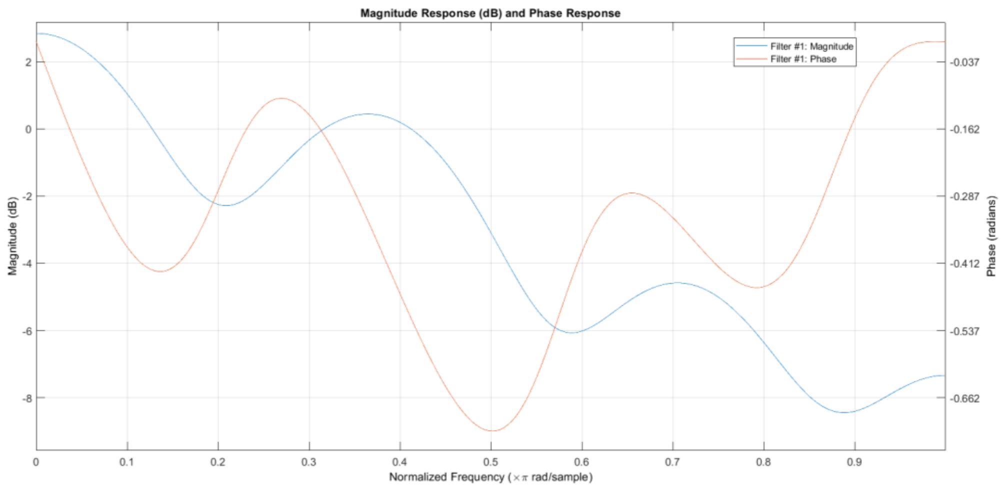

Figure 10: Phase response and Magnitude response of the filter

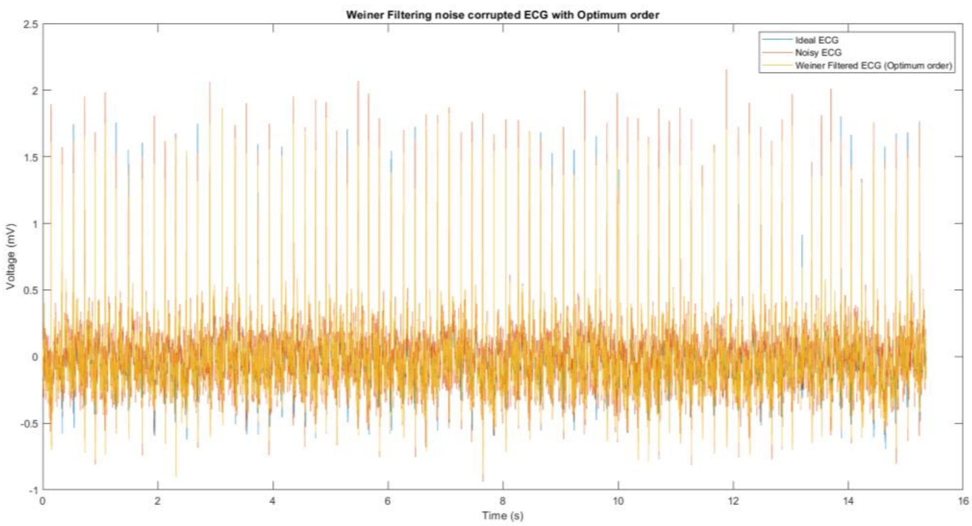

Figure 9: Wenier filtered signal (Optimum filter order)

`

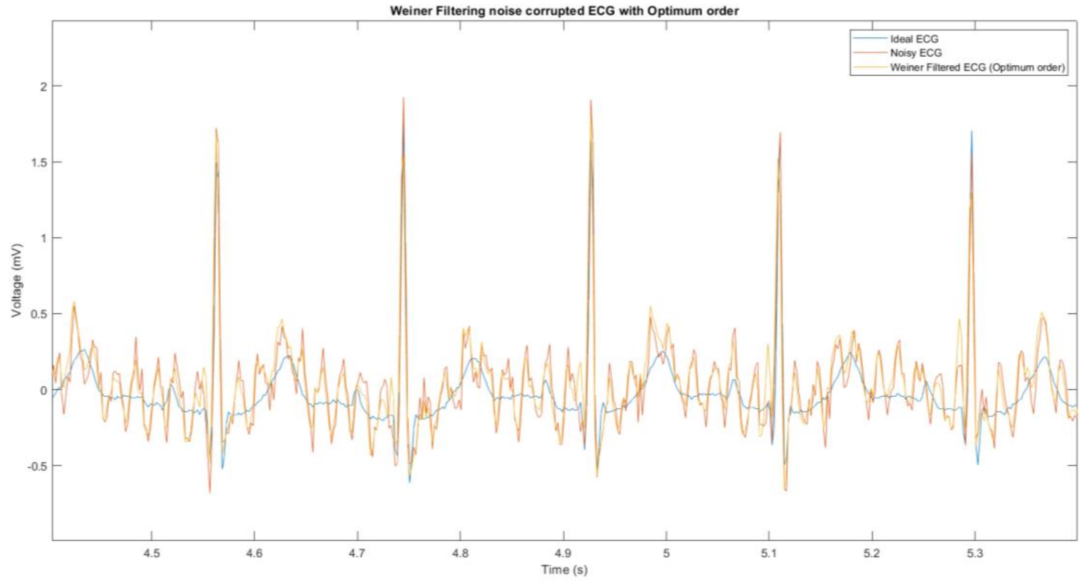

Figure 13: Zoomed image of filtered signal (M=7)

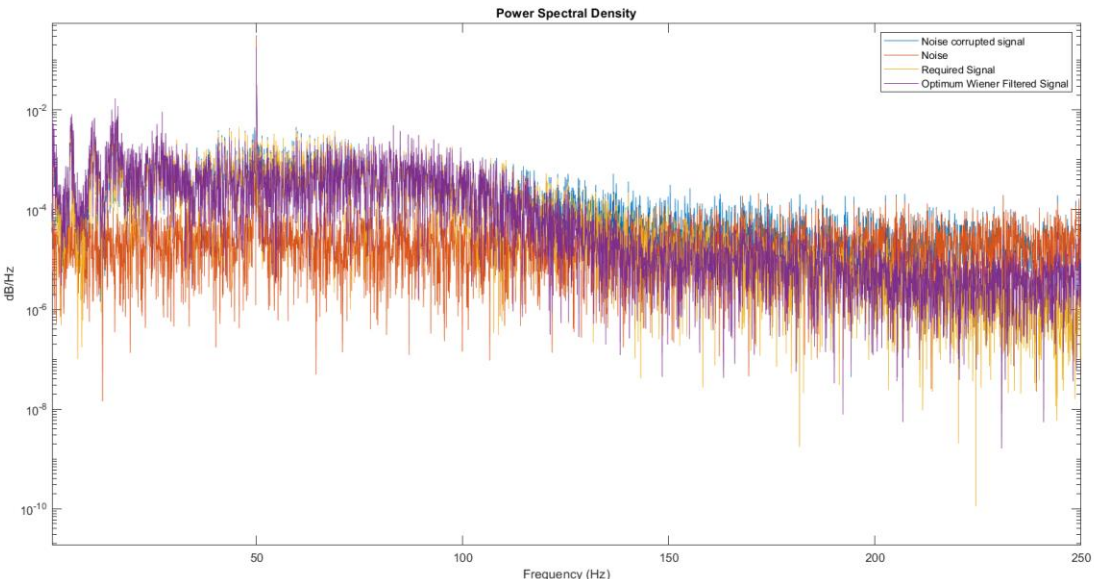

Figure 12: Power Spectral Density

According to Figure 12, the filter has filtered low-frequency components, but when it comes to higher-frequency components that filter has failed. Also in 50Hz frequency spike shows that the filter was unable to remove the sinusoidal noise.

`

1.1.3 Part II

In this section, the linear modelled ECG pulse is used as the ECG signal. Figure 14 shows the linear modelled ECG signal which has 84 sample lengths.

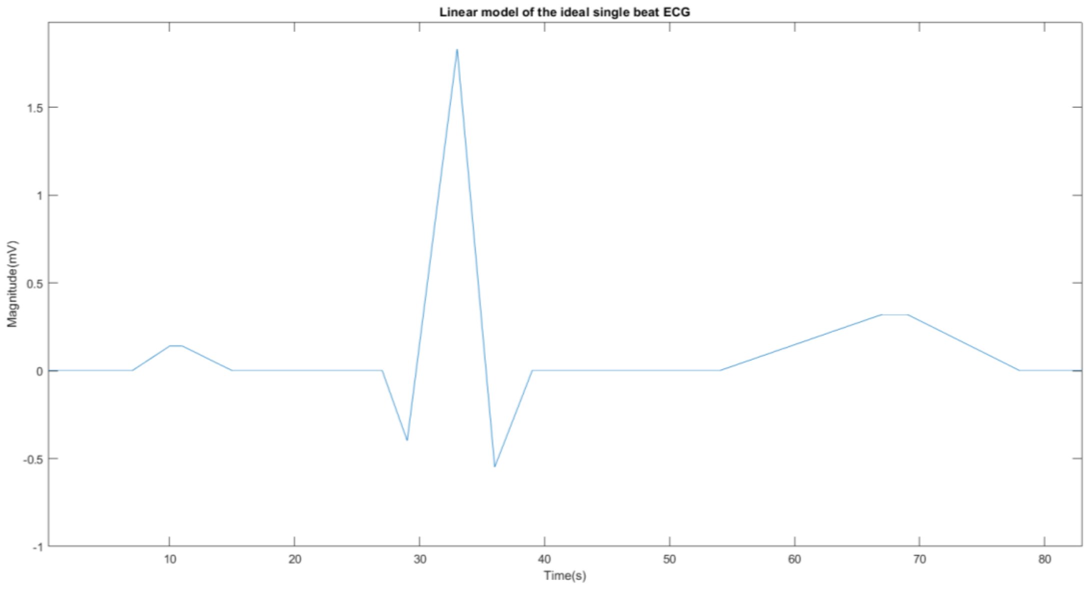

Figure 14: Linear modelled ECG signal(Single beat)

As mentioned in part I, the optimum filter order is obtained using MSE of the estimated signal and desired signal. The following figure 15 shows the Weiner filter MSE with the filter order.

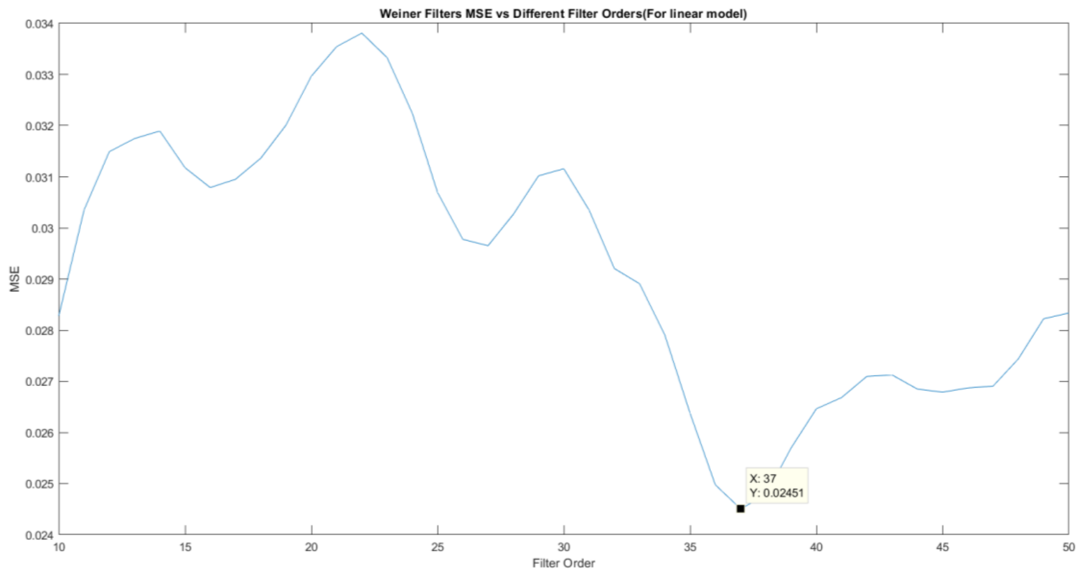

Figure 15: Weiner Filter MSE vs Filter order(Linear model)

According to the above figure, the optimum filter order for the linear modelled ECG signal is 37.

`

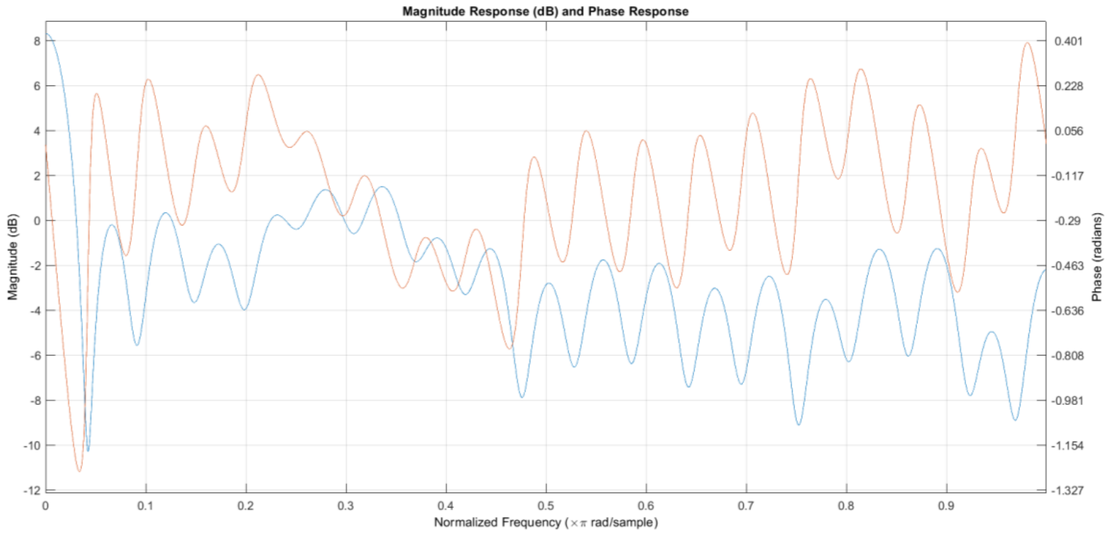

Figure 18: Magnitude Response and Phase Response of the Weiner filter(linear model)

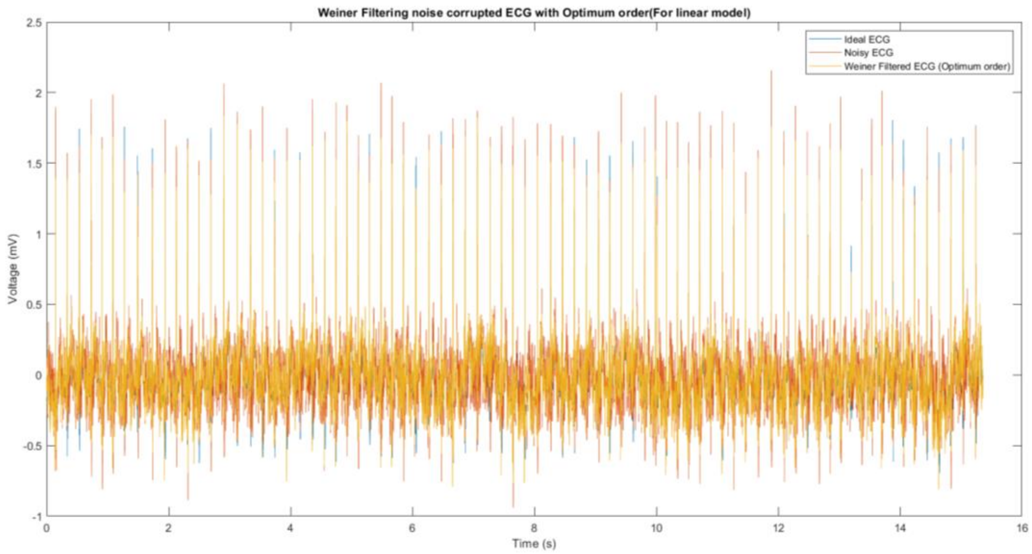

Figure 17: Linear modelled filttered signal

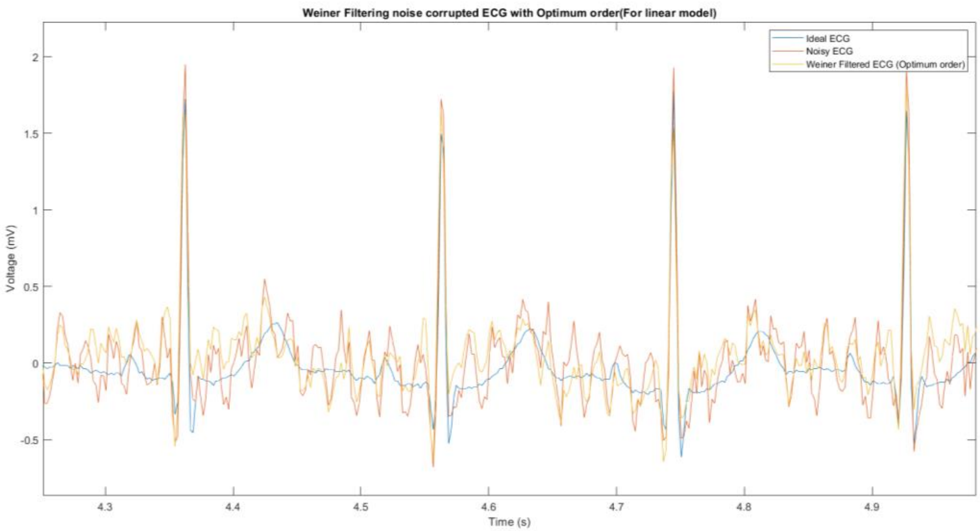

Figure 16: Linear modelled filttered signal(Zoomed)

`

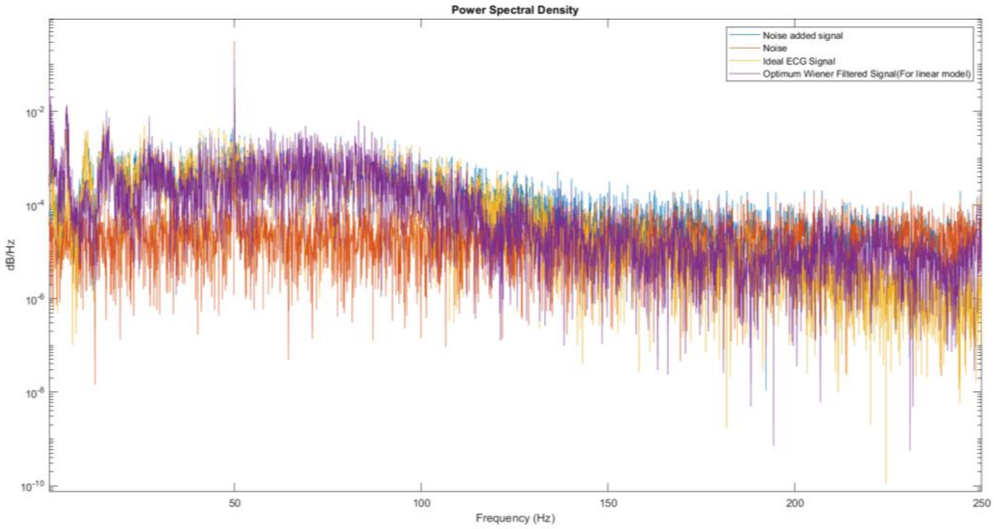

Figure 19: Power Spectral Density (Linear model)

When compared to Figure 12 (ideal ECG signal) in Figure 19(Linear model ECG signal) the high-frequency components filtering process is reduced.

1.2 Frequency domain implementation of the Wiener filter

The following equation is used for calculating the Weiner filter coefficient in the frequency domain.

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

In the above equation $S_{YY}(f)$ is the power spectral density of the $y(n)$ signal and the $S_{NN}(f)$ is the power spectral density of the noise signal. Following figure 20 shows the time domain signal representation after applying the frequency domain Wiener filter.

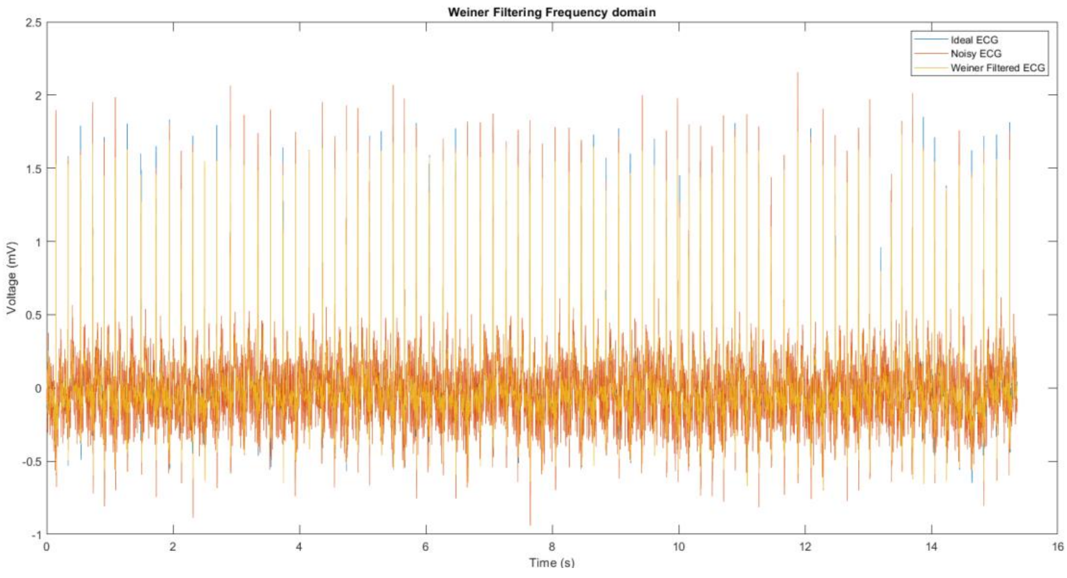

Figure 20: Frequency domain weiner filtered ECG signal

`

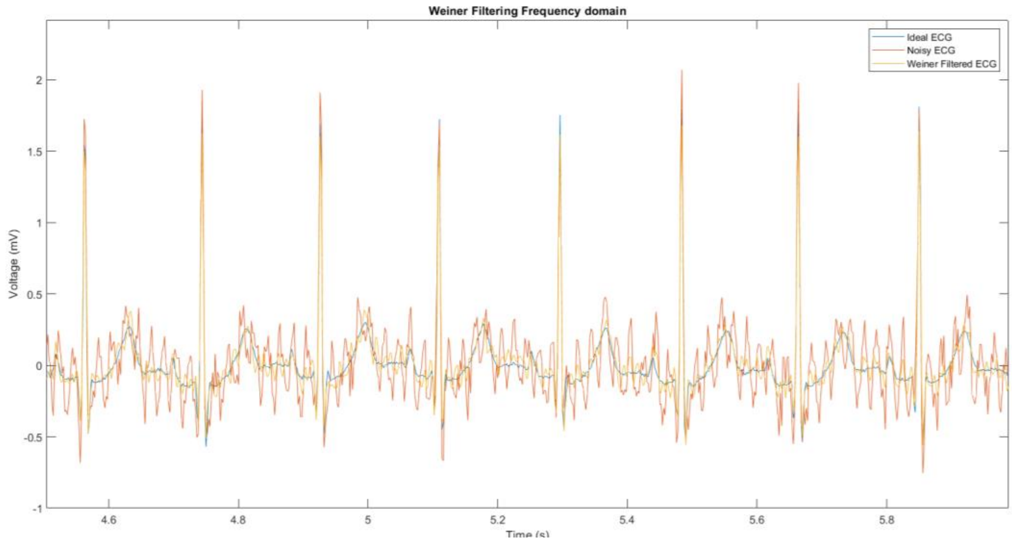

Figure 23:Frequency domain Weiner filtered ECG signal(Zoomed)

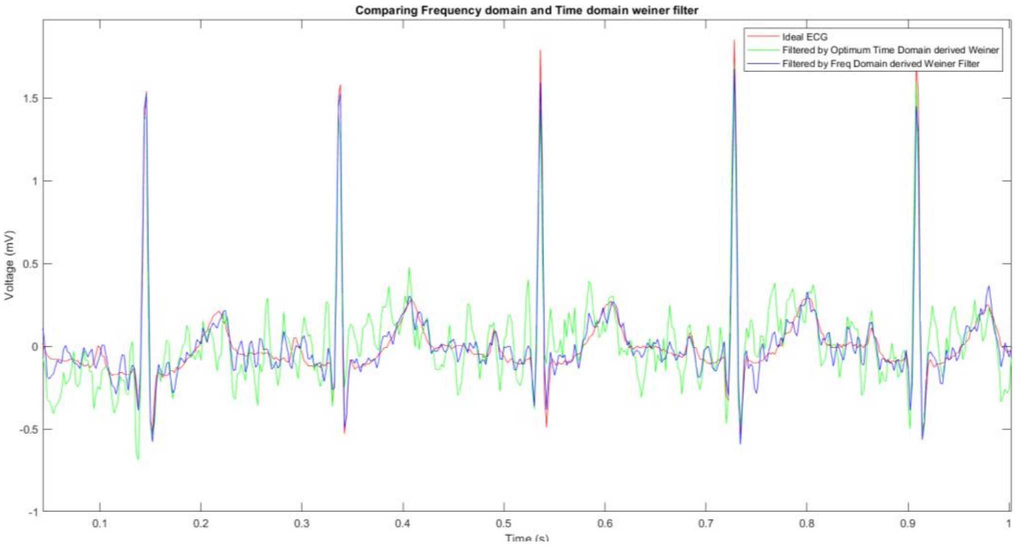

Figure 22: Comparing Frequency domain and Time domain weiner filter

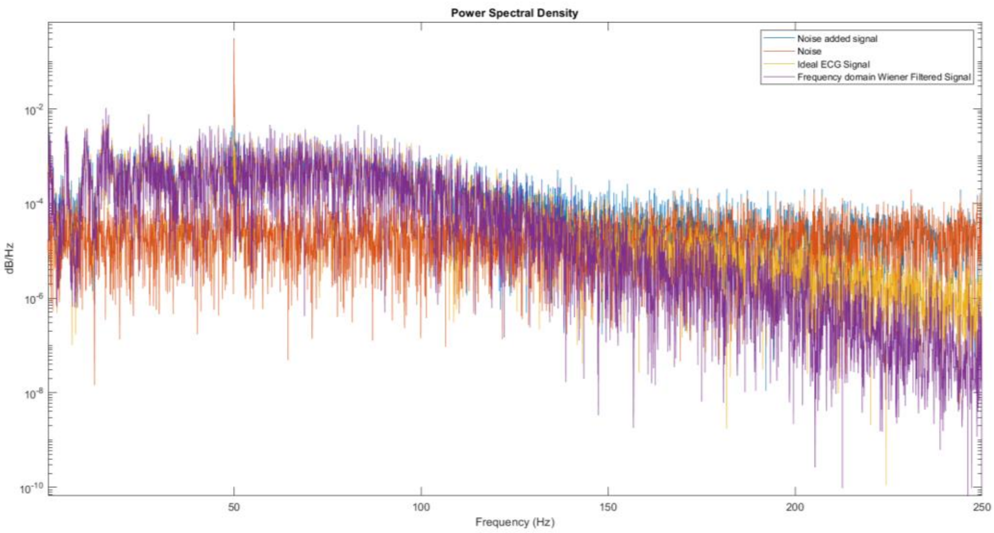

Figure 21: Power Spectral Density

`

According to the Above Power Spectral Density (PSD) graphs, figure 21 has shown the frequency domain Weiner filter was able to filter both high-frequency components and low-frequency components. The PSD of the frequency domain Winer filtered signal is nearly similar to the PSD of the ideal signal. Also, the MSE of the frequency domain (0.0033) filter is low compared to the time domain (0.0269) filter.

1.3 Effect of non-stationary noise on the Wiener filtering

In the following process non-stationary signal is added to the ideal signal. Half of the signal is added 50Hz sinusoidal noise and the other half is added 100Hz sinusoidal noise.

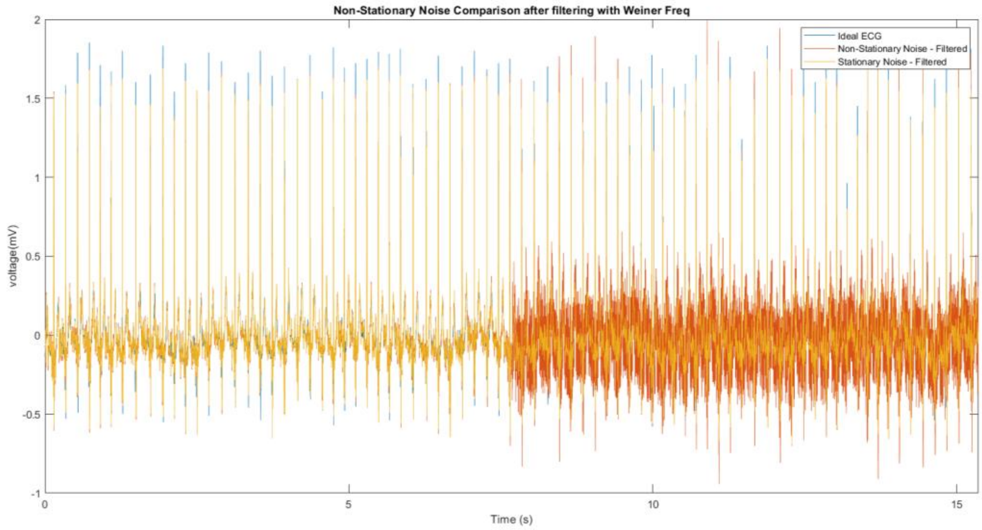

Figure 24: Non-Stationary Noise Comparison after filtering with Weiner Freq

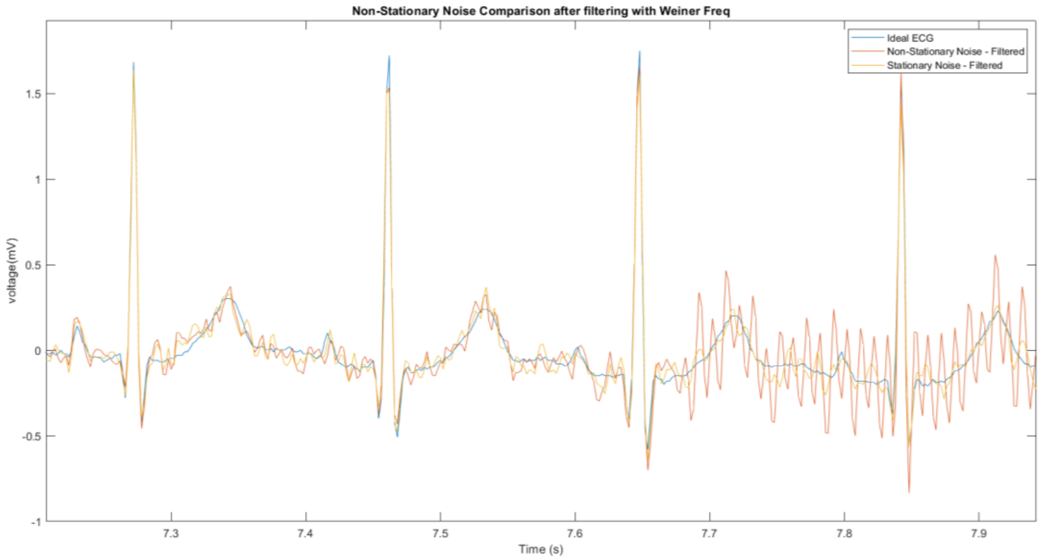

Figure 25: Non-Stationary Noise Comparison after filtering with Weiner Freq (Zoomed)

`

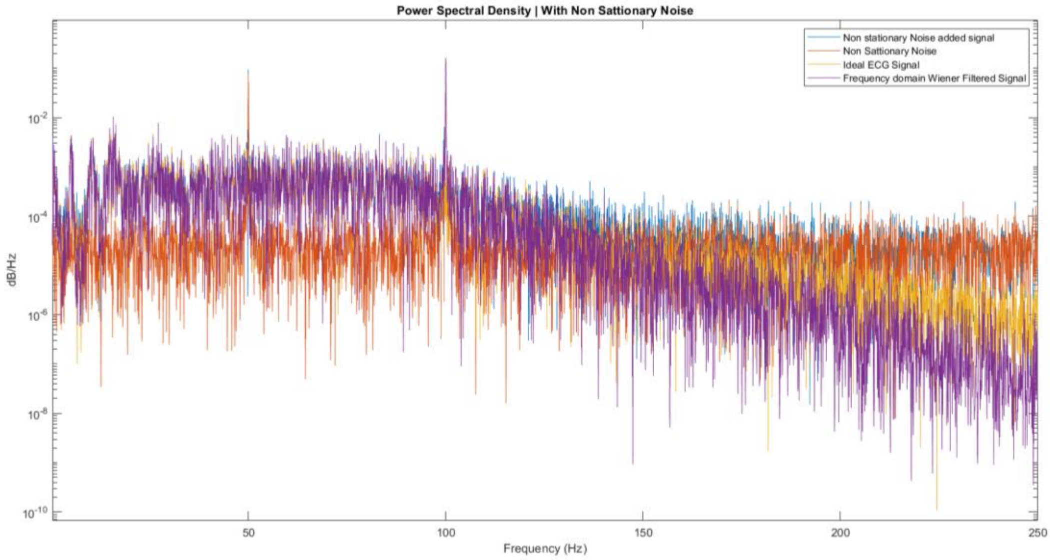

Figure 26: Power Spectral Density (Non-Stationary Noise)

For non-stationary noise, the added signal filter has failed to filter the 100Hz frequency. It shows that when the input signal is non-stationary the Weiner filters cannot be applied.

2 Adaptive filtering

Adaptive filtering is a type of signal-processing technique that involves continuously adjusting the parameters of a filter to optimize its performance based on the characteristics of the input signal. This allows the filter to adapt to changes in the signal over time, making it more effective at removing noise or isolating certain frequencies.

As the Weiner-Hopf equation required more computational power and time to calculate the inverse matrix, several different types of adaptive filter algorithms are utilized including least mean squares (LMS) and recursive least squares (RLS). These filters use algorithms that adjust the filter coefficients in real-time based on the input signal and the desired output signal.

Data Construction

Sampling frequency (f_s) = 500Hz
Number of samples (N) = 5000

$y_i(n)$: Sawtooth waveform with a width of 0.5
$\eta(n)$: Non-stationary noise: use same as in section 1.3
$r(n) = a(\eta_{wg}(n) + \eta_{50}(nf_{50} + \phi_1) + \eta_{100}(nf_{100} + \phi_2))$, where a and $\phi 1$,$\phi 2$ are arbitrary constants.

The following code snip was used to generate the signals.

```
N = 5000; % Number of points
t = linspace(0,5,N)'; % Time vector with fs = 500 Hz
s = sawtooth(2*pi*2*t(1:N,1),0.5);  % Sawtooth signal
n1 = 0.2*sin(2*pi*50*t(1:N/2,1)-phi);   % Sinusoid at 50 Hz
n2 = 0.3*sin(2* pi*100*t(N/2+1:N,1)-phi);   % Sinusoid at 100 Hz
nwg = s - awgn(s,snr,'measured'); % Gaussian white noise
```

Figure 27: Code given to generate the signals

`

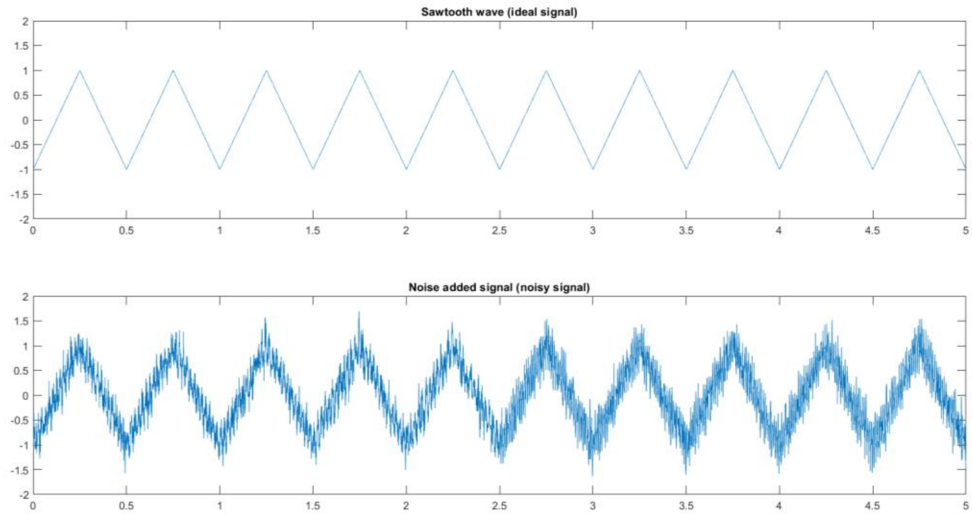

Figure 28: The provided signals

2.1 Least Mean square Algorithm

The following equation is used to obtain the filter.

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

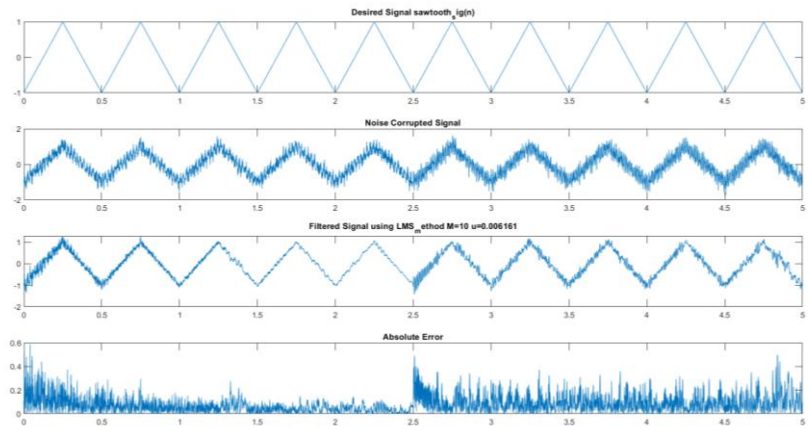

Figure 30: Adaptive filter using LMS

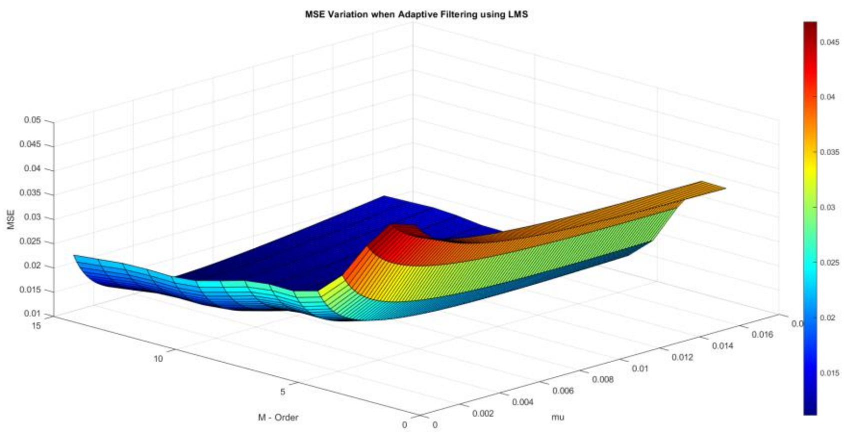

Figure 29: MSE Variation when Adaptive Filtering using LMS

`

In the implementation following arbitrary constants are applied, $a = 1.611$, $\Phi_1 = 1/6$ and $\Phi_2 = 1/2$ By changing the converging factor (μ), filter order (M) and MSE values different optimum filters can be created for a given r(n). For the Mean Square Error calculation desired signal and filtered signal are required. Based on the difference between those signals MSE is calculated.

After implementation, the following minimum MSE = 0.010678 is obtained with filter order (M) = 15 and $\mu = 0.0064686$. For small μ values, the time to do the calculations and converge to a large error is high. Also for the large μ values filter is not even converging. Therefore the error is increasing when considerably increasing the error. When increasing the filter order up to the optimum value the error is decreasing.

2.2 Recursive Least Square Algorithm

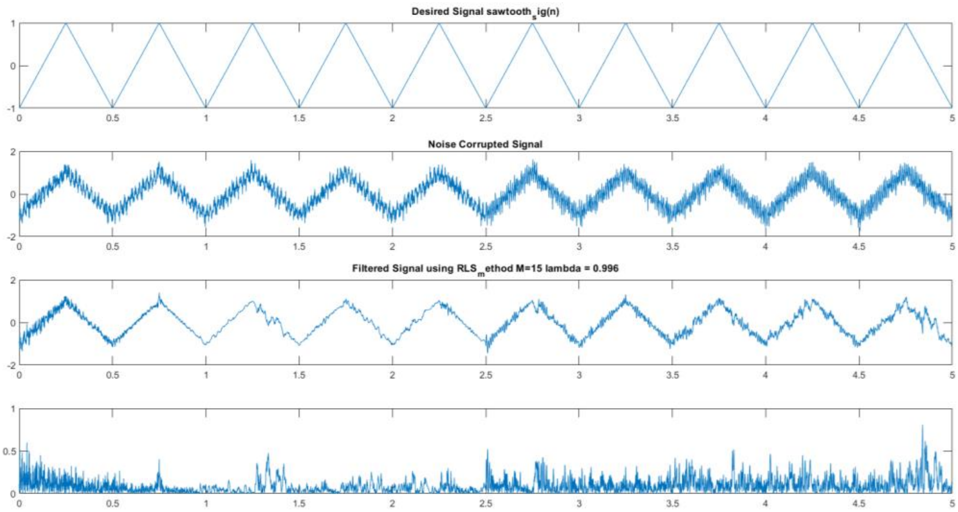

Figure 31: Adaptive filter using RLS

For the following repetitive process, initial values for matrices should be provided. Otherwise, it takes a lot more time to implement.

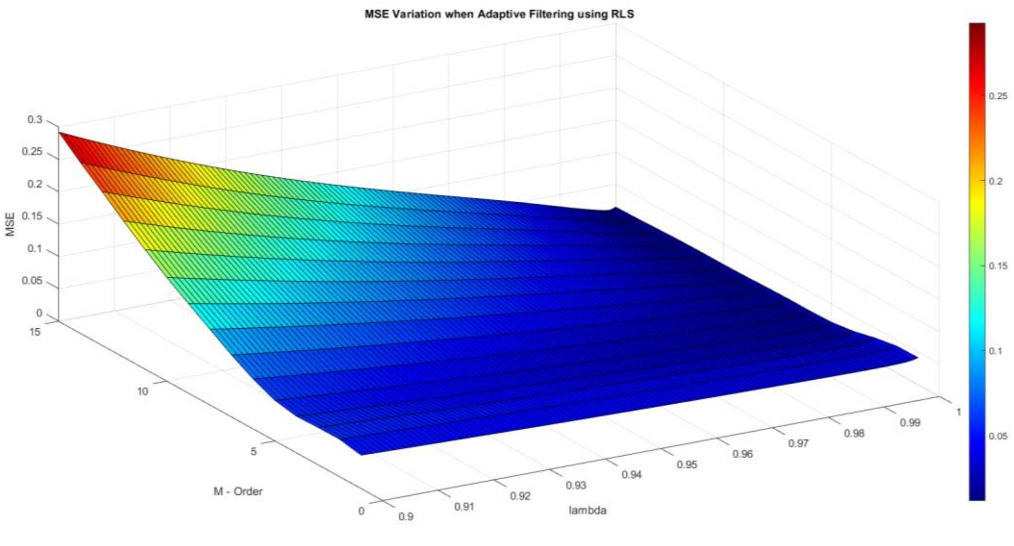

Figure 32: MSE variation vs Filter order (RLS method)

`

After implementation, the following minimum MSE = 0.011856 is obtained with filter order (M) = 10 and $\lambda = 0.9098$. The λ value is always near 1 thus previous repetition is considerable. As shown in figure 26 for larger filter orders λ should be near 1 thus error from the previous sample can be reduced.

2.2.1 Compare the performance of LMS and RLS algorithms

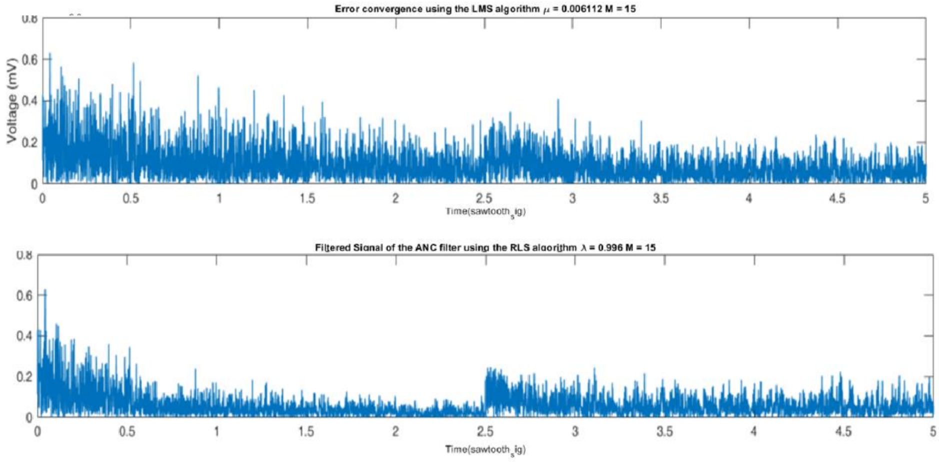

Figure 33: Performance comparison of LMS and RLS

According to above figure 33, Absolute error plots have shown that the RLS algorithm converges faster than the LMS algorithm.

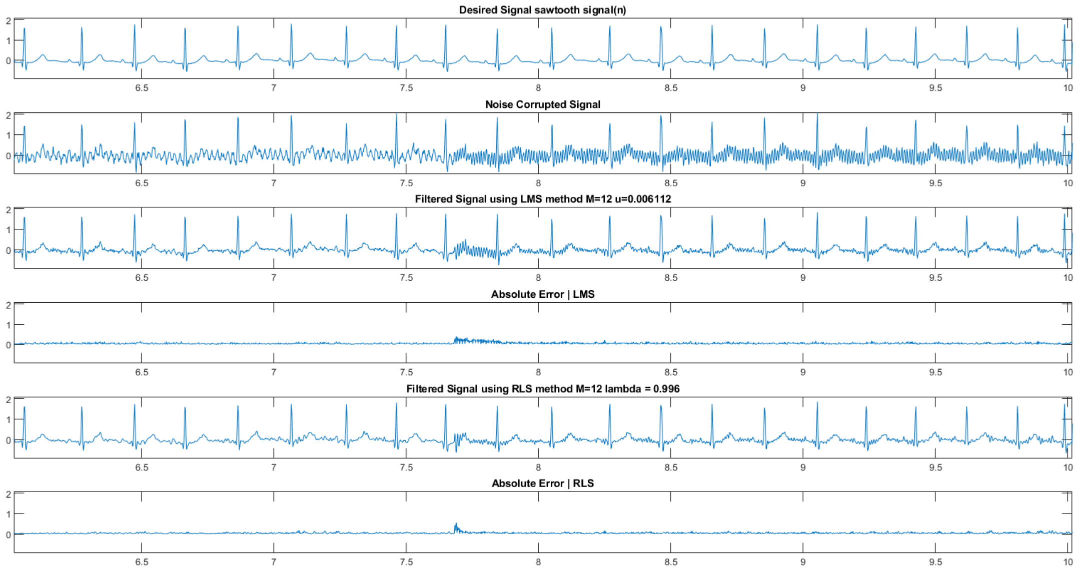

Figure 34: ECG signal filtering using LMS and RLS algorithm in Adaptive filters

According to above figure 34, as a summary of RLS and LMS, both algorithms were able to remove low-frequency components but the LMS algorithm has been unable to remove the high-frequency noise components compared to the RLS algorithm.

`