Department of Electronics and Telecommunication Engineering

University of Moratuwa

BM4151 - Biosignal Processing

MATLAB Assignment 2

Wiener and Adaptive Filtering

Name	**Index**
H. D. M. Prematilaka	180497C

This report is submitted in partial fulfillment of the requirements for the module BM 4151 – Biosignal Processing.

Date of Submission: December 22, 2022

Table of Contents

List of Figures

List of Tables

1 Wiener Filtering (On Stationary Signals)

The following notation will be used during the data construction.

Ideal Signal	$y_i(n)$	idealECG.mat
Sampling Frequency	f_s	500 Hz
White Gaussian Noise	$\eta_{wg}(n)$	SNR is 10dB w.r.t. $y_i(n)$
Sinusoid Noise	$\eta_{50}(n)$	$\eta_{50}(n) = 0.2sin(100\pi n)$
Input signal	$x(n) = y_i(n) + \eta(n)$	$\eta(n) = \eta_{wg}(n) + \eta_{50}(n)$

Table 1: Data Construction Notation

1.1 Discrete-time Domain Implementation of the Wiener Filter

When deriving the optimum Wiener filter, the following assumptions are made.

- The signal and noise are independent
- The signal and noise are stationary
- The relevant signal is known
- The characteristics of the noise are known

The main objective of the filtering process is to minimize the mean square error between the desired output signal and the estimated output signal.

Let the estimated signal be $\hat{y}(n)$, the known desired signal be $y(n)$ (stationary process with mean zero) and the error be $e(n)$.

$$\hat{y}(n) = \sum_{k=0}^{M-1} w_k x(n-k) \qquad (order = M-1)$$

$$\hat{y}(n) = \boldsymbol{w}^T \boldsymbol{x}(n) = \boldsymbol{x}^T(n)\boldsymbol{w}$$

$$e(n) = y(n) - \hat{y}(n)$$

$$J(\boldsymbol{w}) = E\left[e^2(n)\right]$$

$$J(\boldsymbol{w}) = E\left[(y(n) - \hat{y}(n))^2\right]$$

$$= E[(y(n) - \boldsymbol{w}^T\boldsymbol{x}(n))(y(n) - \boldsymbol{x}^T(n)\boldsymbol{w})]$$

$$= \underbrace{E[y^2(n)]}_{\sigma_y^2(n) \because E[y(n)]=0} - \boldsymbol{w}^T \underbrace{E[\boldsymbol{x}(n)y(n)]}_{\Theta_{xy}} - E[y(n)\boldsymbol{x}^T(n)]\boldsymbol{w} + \boldsymbol{w}^T \underbrace{E[\boldsymbol{x}(n)\boldsymbol{x}^T(n)]}_{\Phi_x}\boldsymbol{w}$$

$$J(\boldsymbol{w}) = \sigma_y^2(n) - \boldsymbol{w}^T\Theta_{xy} - \Theta_{xy}^T\boldsymbol{w} + \boldsymbol{w}^T\Phi_x\boldsymbol{w}$$

$$\frac{dJ(\boldsymbol{w})}{d\boldsymbol{w}} = -\Theta_{xy} - \Theta_{xy} + 2\Phi_x\boldsymbol{w} \qquad (1)$$

To find the minimum,

$$\frac{dJ(\boldsymbol{w})}{d\boldsymbol{w}} = -\Theta_{xy} - \Theta_{xy} + 2\Phi_x\boldsymbol{w}_0 = 0$$

$$\Phi_x\boldsymbol{w}_0 = \Theta_{xy}$$

$$\boldsymbol{w}_0 = \Phi_x^{-1}\Theta_{xy} \qquad (2)$$

The Eq. 2 represents the Wiener Hopf equation. Further, the input signal x(n) contains the actual signal y(n) and the noise signal $\eta(n)$

$$
\begin{aligned}
x(n) &= y(n) + \eta(n) \\
\boldsymbol{X}(n) &= \boldsymbol{Y}(n) + \boldsymbol{N}(n) \\
E[\boldsymbol{Y}(n)\boldsymbol{N}^T(n)] &= E[\boldsymbol{N}(n)\boldsymbol{Y}^T(n)] = 0 \\
\Phi_X &= E[\boldsymbol{X}(n)\boldsymbol{X}^T(n)] \\
&= E[(\boldsymbol{Y}(n) + \boldsymbol{N}(n))\,(\boldsymbol{Y}(n) + \boldsymbol{N}(n))^T] \\
&= \underbrace{E[\boldsymbol{Y}(n)\boldsymbol{Y}^T(n)]}_{\Phi_Y} + \underbrace{E[\boldsymbol{Y}(n)\boldsymbol{N}^T(n)]}_{0} + \underbrace{E[\boldsymbol{N}(n)\boldsymbol{Y}^T(n)]}_{0} + \underbrace{E[\boldsymbol{N}(n)\boldsymbol{N}^T(n)]}_{\Phi_N} \\
\Phi_X &= \Phi_Y + \Phi_N \qquad (3) \\
\Theta_{\boldsymbol{X}y} &= E[\boldsymbol{X}(n)y(n)] \\
&= E[(\boldsymbol{Y}(n) + \boldsymbol{N}(n))y(n)] \\
&= \underbrace{E[\boldsymbol{Y}(n)y(n)]}_{\Theta_{\boldsymbol{Y}y}} + \underbrace{E[\boldsymbol{N}(n)y(n)]}_{0} \\
\Theta_{\boldsymbol{X}y} &= \Theta_{\boldsymbol{Y}y} \qquad (4)
\end{aligned}
$$

By substituting Eq. 3 and Eq. 4 in Eq.2, we can further simplify the Wiener Hopf Equation.

$$
\begin{aligned}
\boldsymbol{w}_0 &= \Phi_x^{-1}\Theta_{xy} \\
\boldsymbol{w}_0 &= (\Phi_Y + \Phi_N)^{-1}\Theta_{\boldsymbol{Y}y}
\end{aligned}
$$

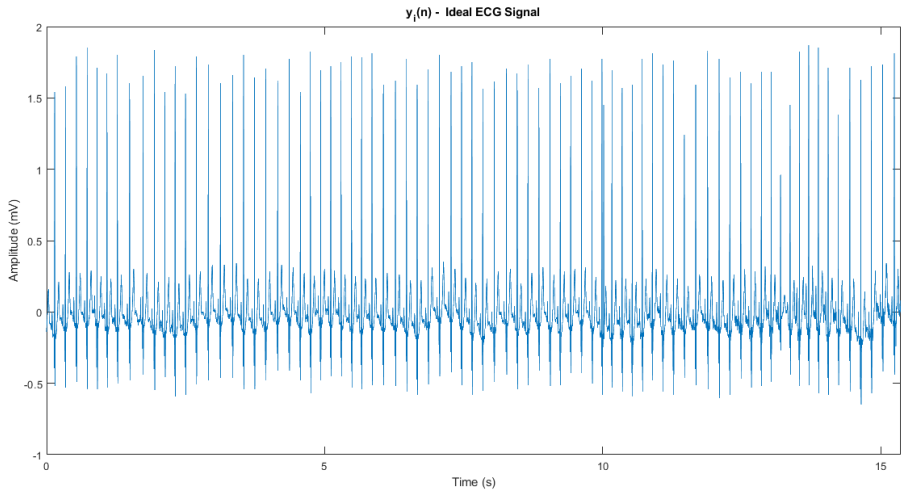

Figure 1: Ideal ECG Signal

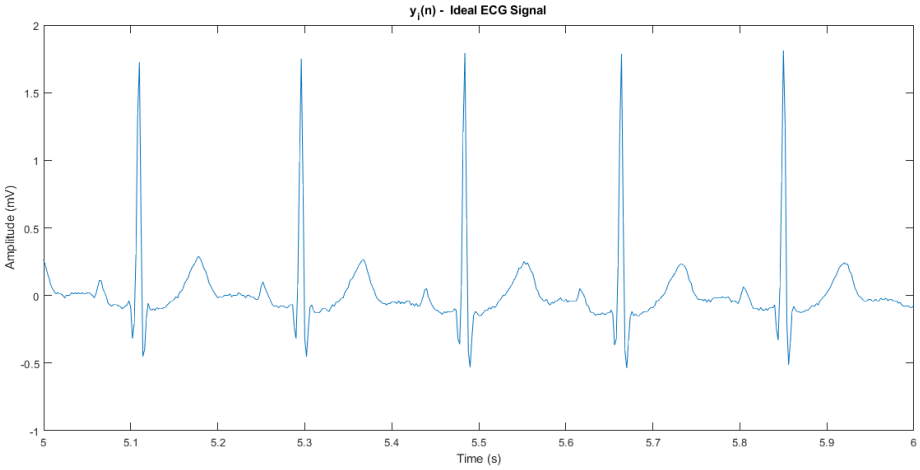

Figure 2: Ideal ECG Signal (Zoomed In)

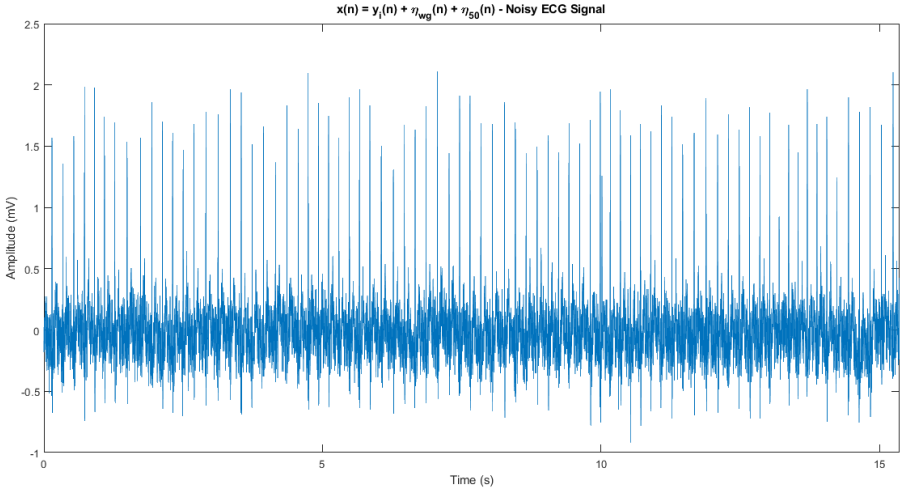

Figure 3: Noisy ECG Signal

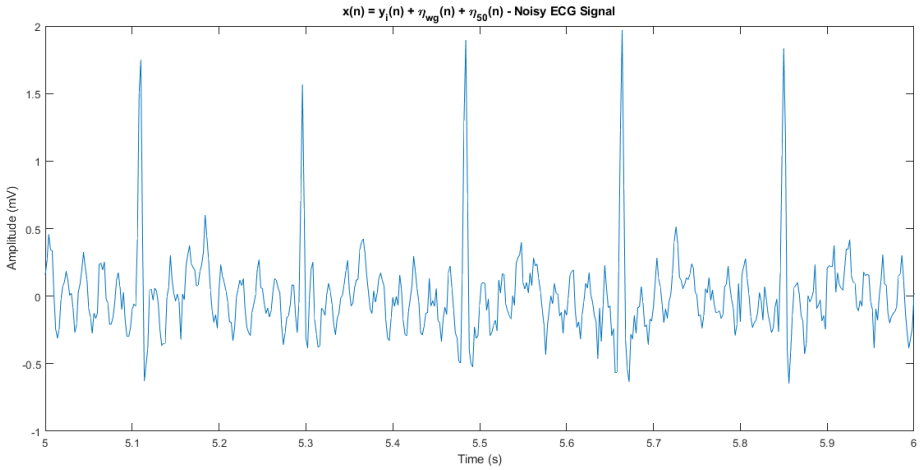

Figure 4: Noisy ECG Signal (Zoomed In)

1.1.1 Part 1

As instructed, a single beat of the desired signal and an arbitrary isoelectric segment from the noisy ECG signal were handpicked after observing the corresponding waves. Noisy isoelectric segment is extracted starting from the T wave till the beginning of the P wave of the next ECG beat. The selected wavelets are shown below.

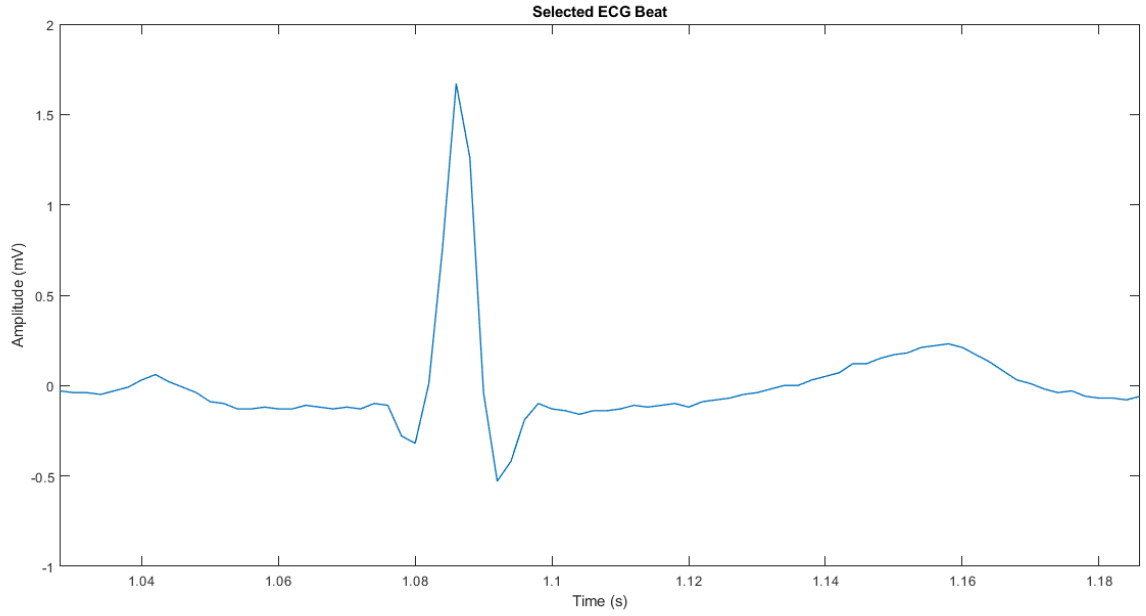

Figure 5: Selected ECG Beat

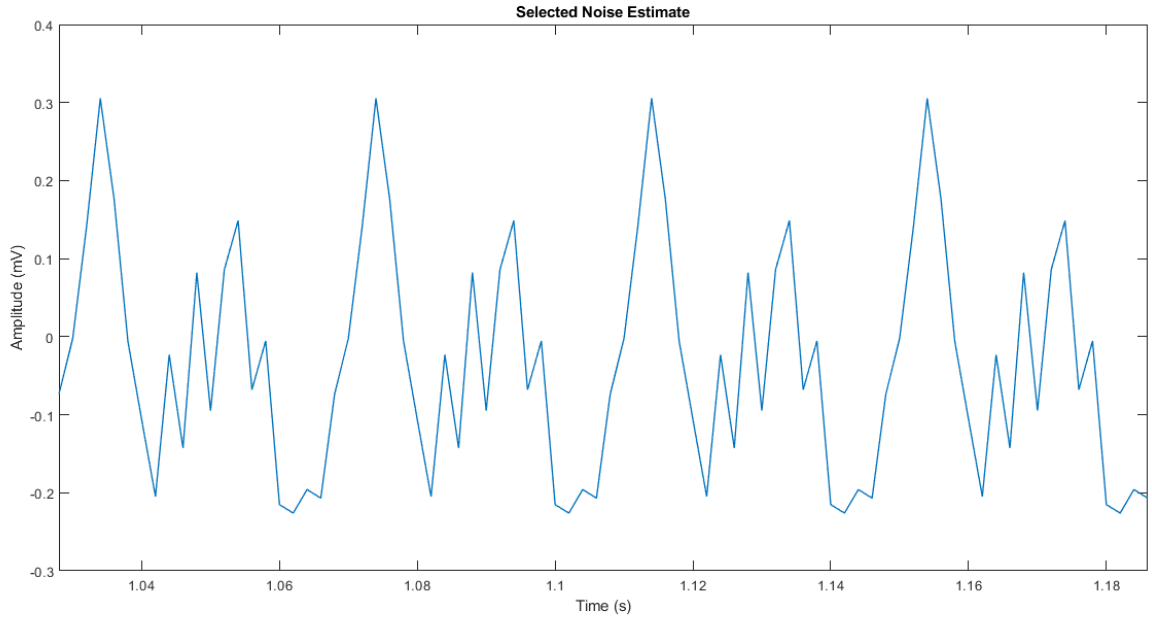

Figure 6: Noisy Isoelectric Segment

1.1.1.1 Finding Optimum Weights for an Arbitrary Order

An order of 10 was used as the arbitrary order for the Wiener Filter. After computations, the following weights were obtained as the optimum weights for the Wiener Filter of order 10.

w_0	0.6257
w_1	0.3082
w_2	-0.0446
w_3	-0.1276
w_4	0.0052
w_5	0.1350
w_6	0.1138
w_7	0.0535
w_8	0.0044
w_9	0.0085

Table 2: Optimum Wiener Weights for Order 10

The following diagram indicates the original ECG signal, the noisy template and the Wiener-filtered noisy signal segment.

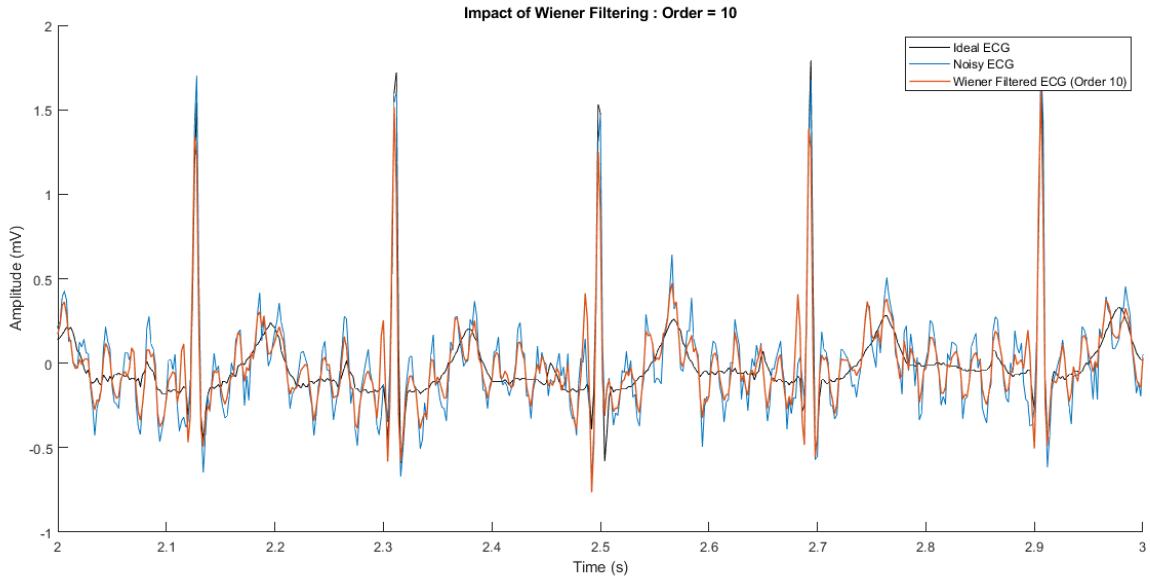

Figure 7: Impact of Wiener Filtering (Order 10)

1.1.1.2 Finding the Optimum Filter Order

For a given order, we could find the optimum weights of the filter. However, that doesn't guarantee that we have chosen the optimum order that would result in the minimum mean squared error. Accordingly, an iterative loop was carried out to figure out the optimum order of the Wiener in this context.

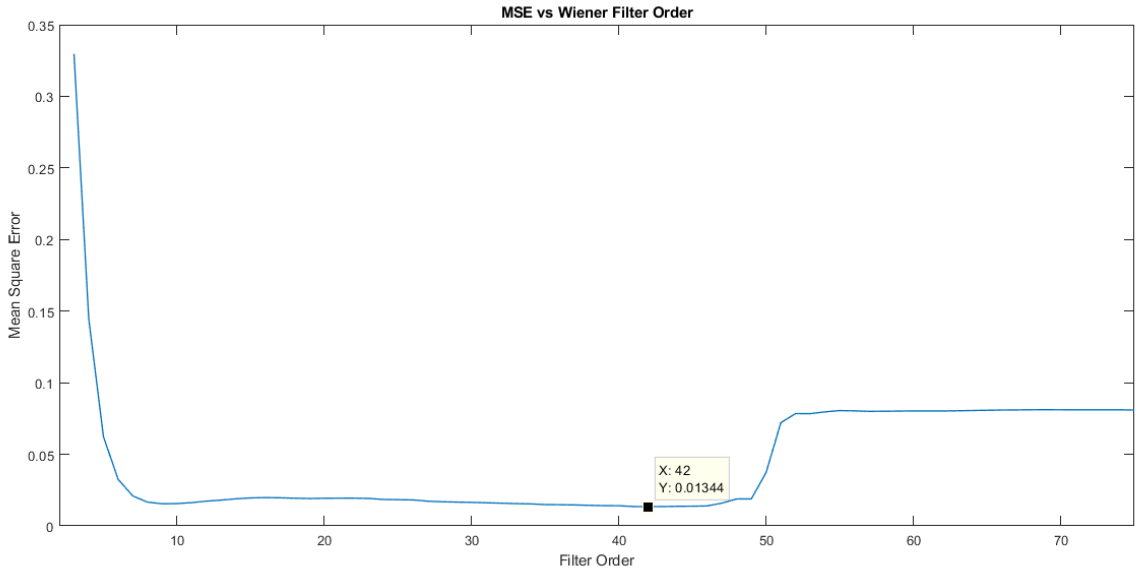

Figure 8: MSE vs Wiener Filter Order

As we can see in Fig. 8, the optimum filter order is **42**. The weights corresponding to the aforementioned Wiener filter are summarized in the following table.

w_0	0.6791	w_{21}	0.0154
w_1	0.2207	w_{22}	0.0327
w_2	-0.0893	w_{23}	0.0277
w_3	-0.1291	w_{24}	0.0118
w_4	-0.0173	w_{25}	0.0102
w_5	0.0378	w_{26}	0.0172
w_6	0.0781	w_{27}	0.0065
w_7	-0.0011	w_{28}	0.0140
w_8	-0.0539	w_{29}	0.0141
w_9	-0.0492	w_{30}	0.0214
w_{10}	-0.0190	w_{31}	0.0302
w_{11}	-0.0325	w_{32}	0.0278
w_{12}	-0.0197	w_{33}	0.0305
w_{13}	-0.0217	w_{34}	0.0348
w_{14}	0.0311	w_{35}	0.0344
w_{15}	0.0190	w_{36}	0.0391
w_{16}	-0.0087	w_{37}	0.0323
w_{17}	-0.0178	w_{38}	0.0256
w_{18}	0.0047	w_{39}	0.0162
w_{19}	0.0265	w_{40}	0.0219
w_{20}	-0.0439	w_{41}	-0.0141

Table 3: Optimum Wiener Weights for Order 42

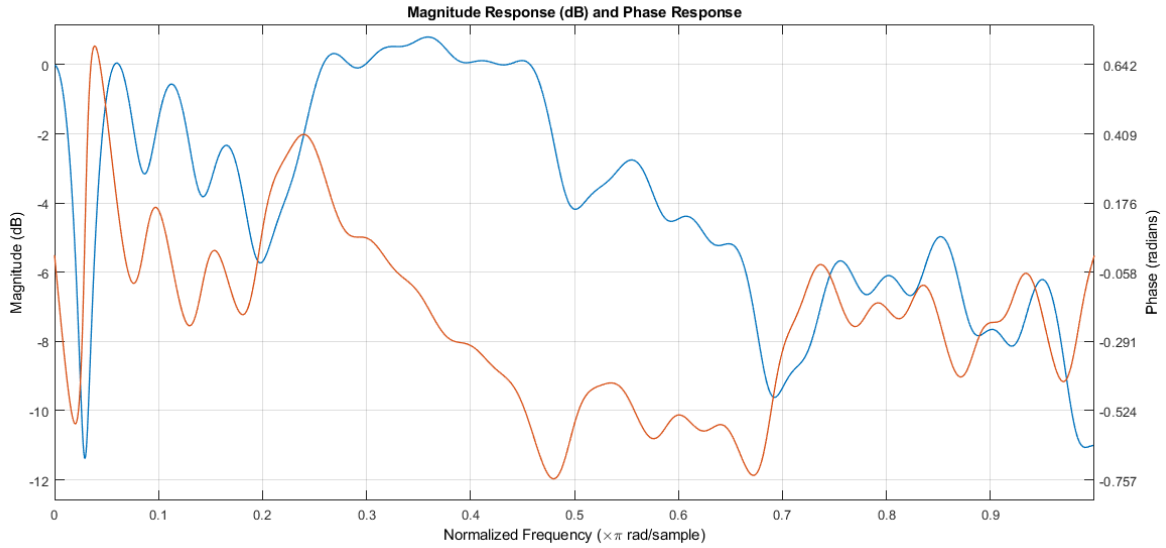

Figure 9: Magnitude and Phase Response of the Optimum Wiener Filter (Order 42)

1.1.1.3 Filtering $\hat{y}(n)$ using the Optimum Wiener Filter

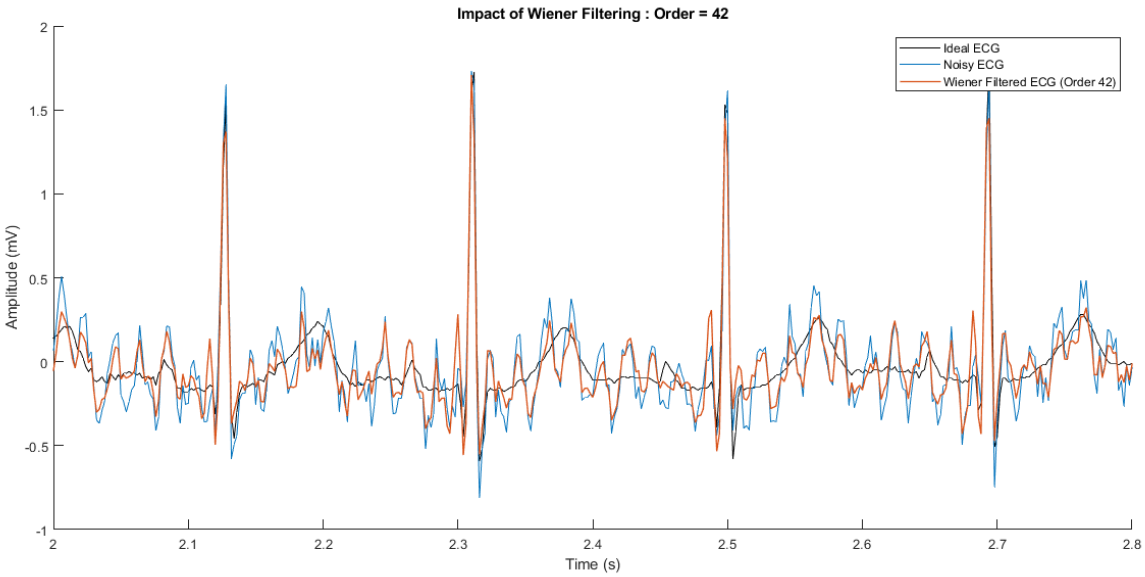

Figure 10: Impact of Wiener Filtering (Optimum Order 42)

1.1.1.4 Plotting the Spectra of $y_i(n)$, $\eta(n)$, $x(n)$ and $\hat{y}(n)$

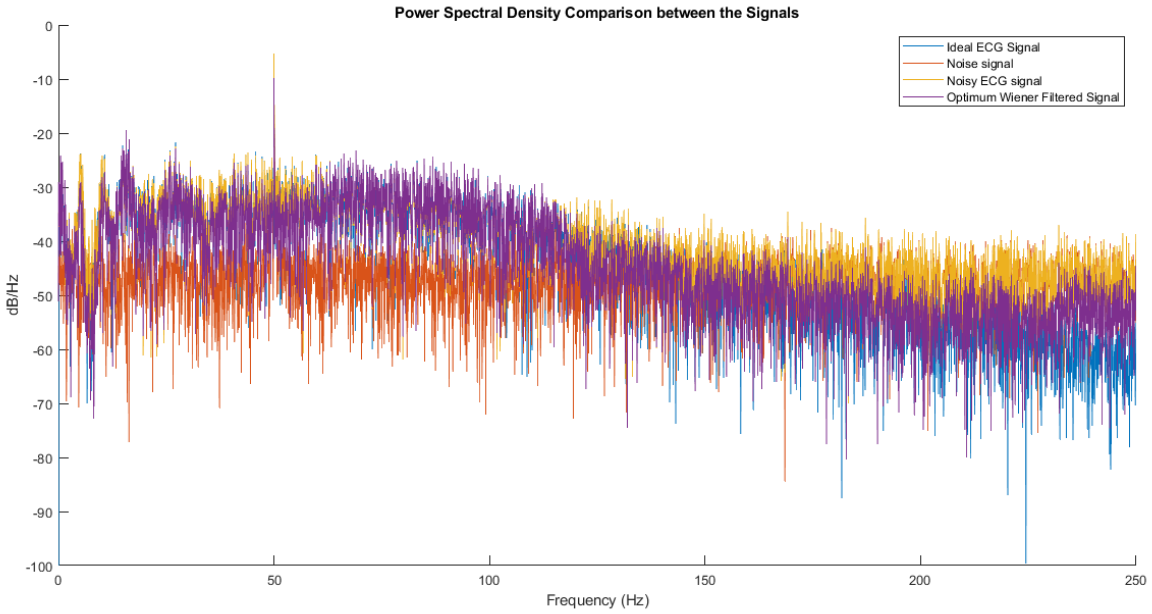

Figure 11: Comparison between the PSDs of the Signals

1.1.1.5 Interpreting the Spectra and the Magnitude Response of the Filter

The magnitude response diagram which is given in Fig. 9 clearly indicates that it treats specific frequency components in different ways in contrast to the conventional filters which typically treat a band of frequencies in the same manner. This is a positive point as the filter is adapted to preserve as much of the ideal ECG signal's information as possible.

When we compare the PSDs (Fig. 11), we can clearly observe that the optimum Wiener filter had suppressed the higher frequency noise components to a considerable extent. However, it has not managed to eliminate the spike at the 50 Hz division.

1.1.2 Part 2

As instructed, the desired signal was linearly modelled using a few samples from the previous part. In addition, an arbitrary isoelectric segment from the noisy ECG signal was handpicked after observing the corresponding waves (For convenience, noise segment used in part 1 was taken as a reference). The noisy isoelectric segment is extracted starting from the T wave till the beginning of the P wave of the next ECG beat. The selected wavelets are shown below.

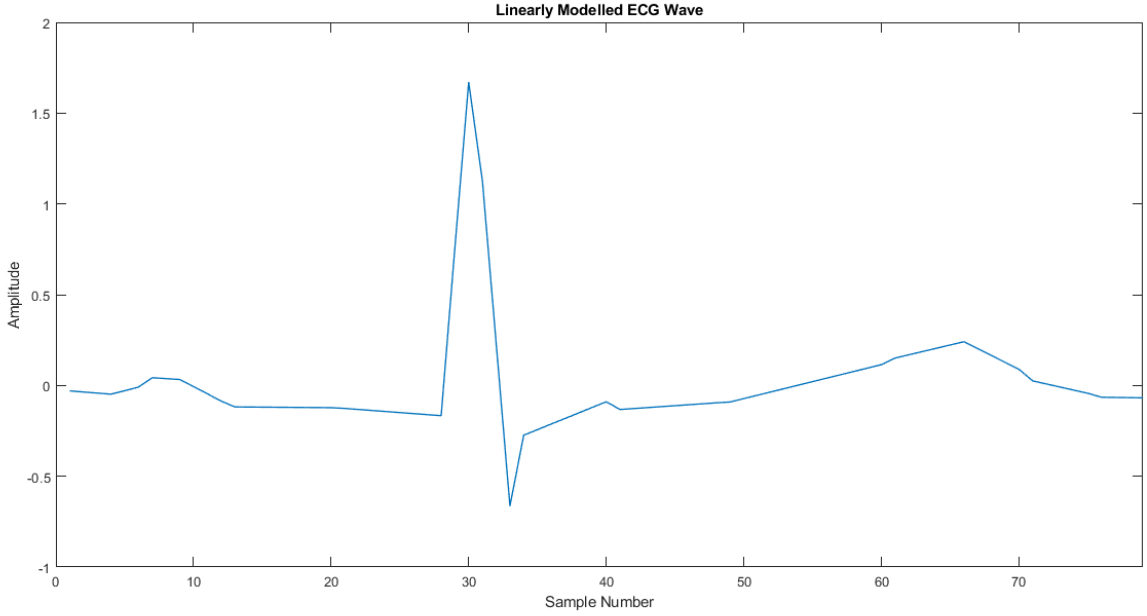

Figure 12: Linearly Modelled ECG Beat

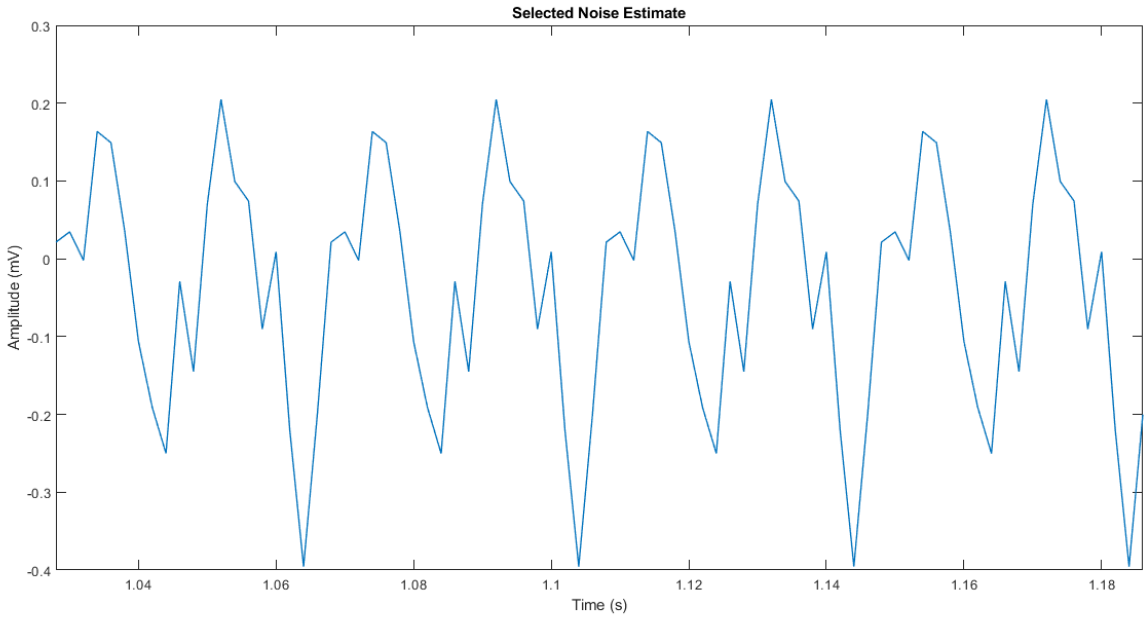

Figure 13: Noisy Isoelectric Segment

1.1.2.1 Finding Optimum Weights for an Arbitrary Order

An order of 16 was used as the arbitrary order for the Wiener Filter. After computations, the following weights were obtained as the optimum weights for the Wiener Filter of order 16.

w_0	0.6865
w_1	0.2375
w_2	-0.1020
w_3	-0.1906
w_4	0.0036
w_5	0.0990
w_6	0.0624
w_7	0.0141
w_8	-0.0153
w_9	-0.0153
w_{10}	0.0277
w_{11}	0.0227
w_{12}	0.0133
w_{13}	0.0174
w_{14}	0.0271
w_{15}	-0.0506

Table 4: Optimum Wiener Weights for Order 16

The following diagram indicates the original ECG signal, the noisy template and the Wiener-filtered noisy signal segment.

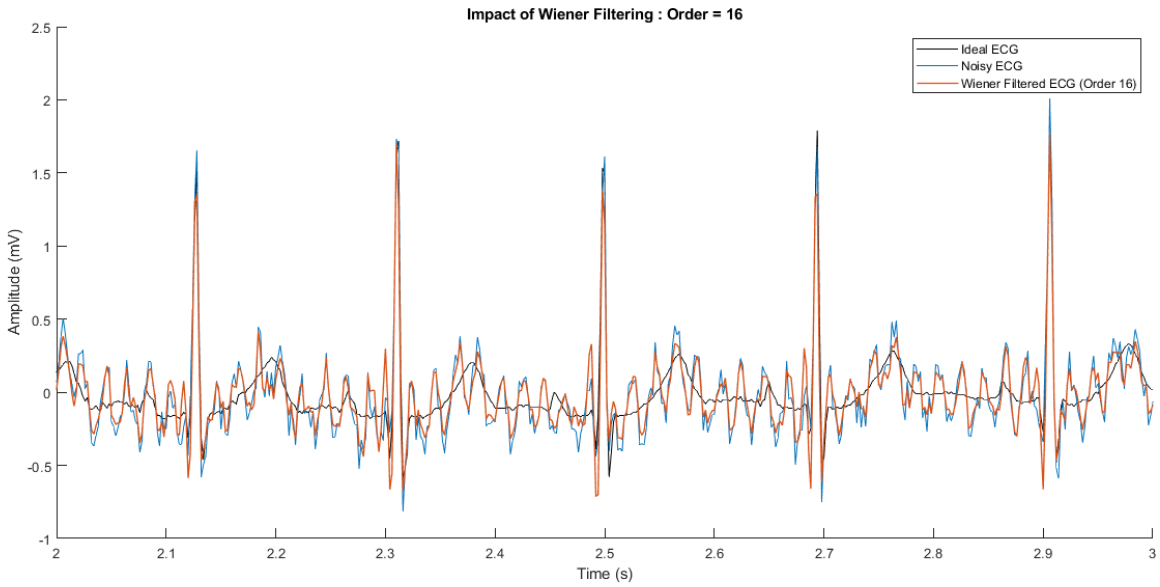

Figure 14: Impact of Wiener Filtering (Order 16)

1.1.2.2 Finding the Optimum Filter Order

The same procedure as in part 1 was carried out and the Wiener filtering was looped over a range of selected order values

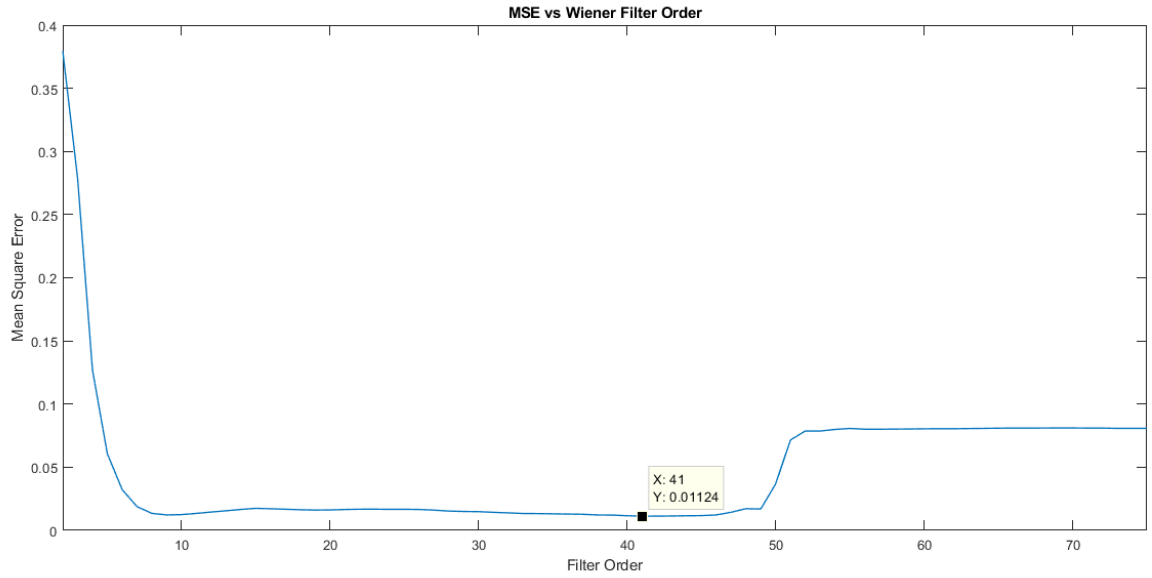

Figure 15: MSE vs Wiener Filter Order

As we can see in Fig. 15, the optimum filter order is **41**. In the previous part, the optimum order was 42 and in this case, it is 41. The weights corresponding to the optimum Wiener filter are summarized in the following table.

w_0	0.7043	w_{21}	0.0181
w_1	0.1864	w_{22}	0.0331
w_2	-0.0826	w_{23}	0.0266
w_3	-0.1404	w_{24}	0.0120
w_4	0.0209	w_{25}	0.0161
w_5	0.0497	w_{26}	0.0147
w_6	0.0288	w_{27}	0.0142
w_7	-0.0080	w_{28}	0.0060
w_8	-0.0396	w_{29}	0.0134
w_9	-0.0528	w_{30}	0.0208
w_{10}	-0.0136	w_{31}	0.0271
w_{11}	-0.0291	w_{32}	0.0325
w_{12}	0.0266	w_{33}	0.0235
w_{13}	-0.0115	w_{34}	0.03336
w_{14}	0.0327	w_{35}	0.0362
w_{15}	0.0161	w_{36}	0.0398
w_{16}	-0.0002	w_{37}	0.0283
w_{17}	-0.0152	w_{38}	0.0258
w_{18}	0.0056	w_{39}	0.0205
w_{19}	0.0219	w_{40}	0.0081
w_{20}	-0.0417	-	-

Table 5: Optimum Wiener Weights for Order 41

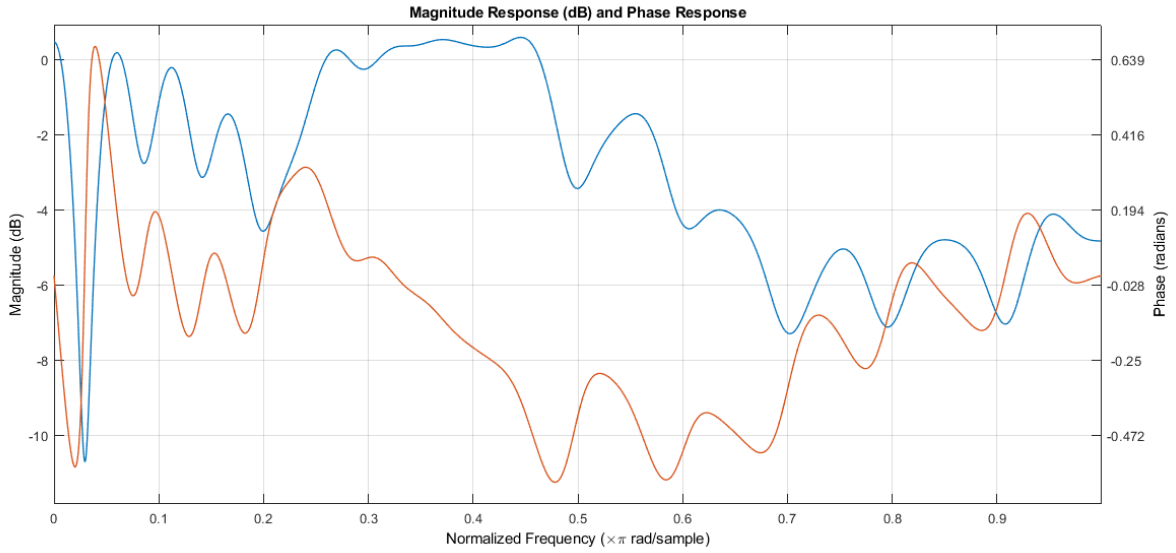

Figure 16: Magnitude and Phase Response of the Optimum Wiener Filter (Order 41)

1.1.2.3 Filtering $\hat{y}(n)$ using the Optimum Wiener Filter

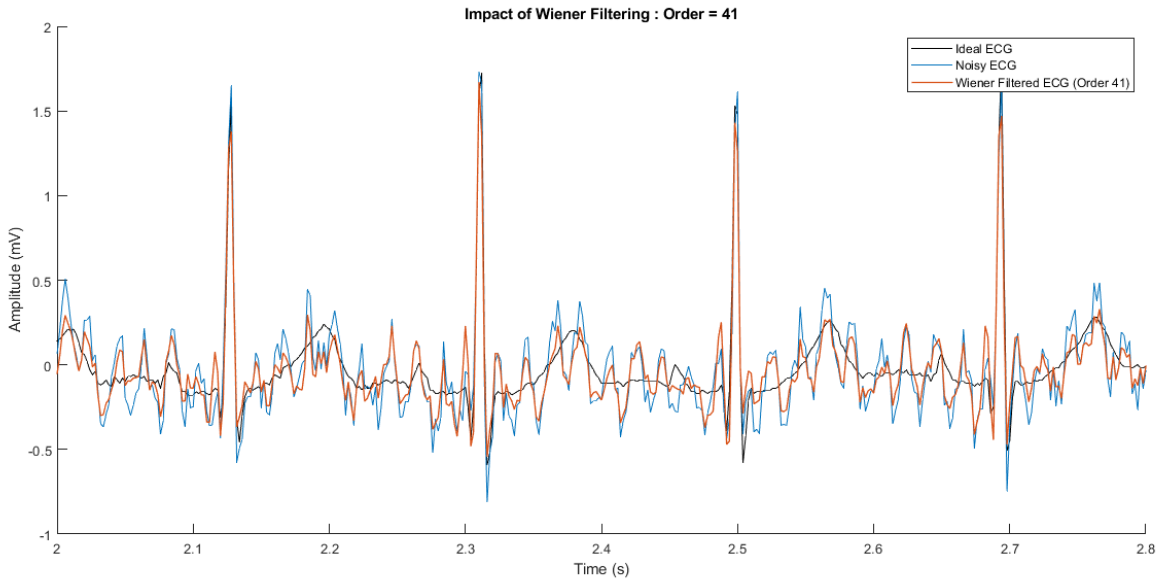

Figure 17: Impact of Wiener Filtering (Optimum Order 41)

1.1.2.4 Plotting the Spectra of $y_i(n)$, $\eta(n)$, $x(n)$ and $\hat{y}(n)$

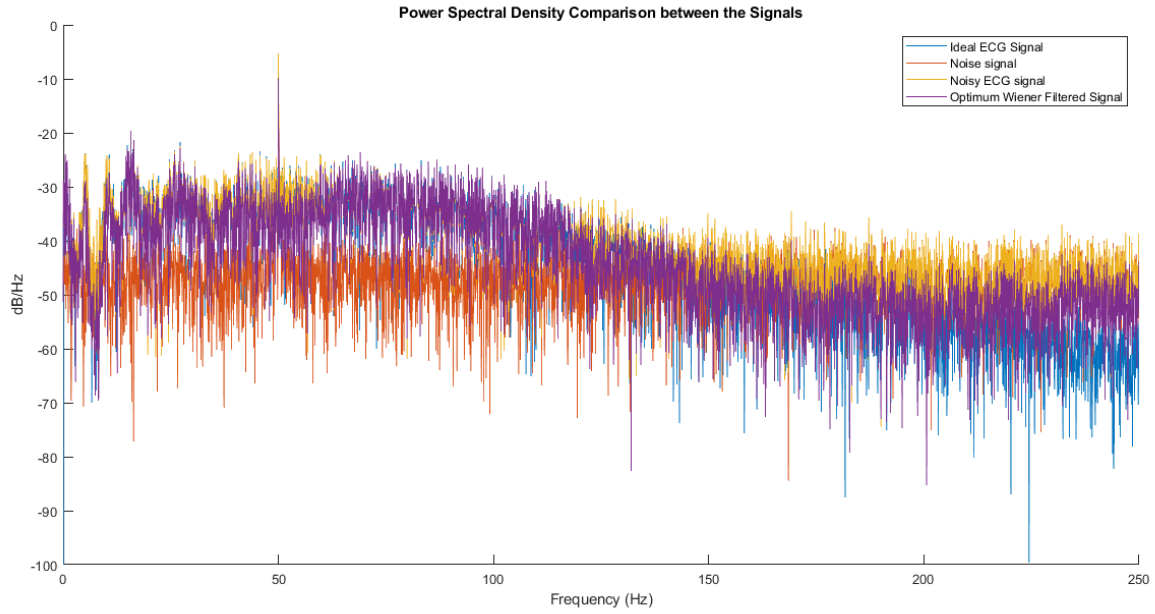

Figure 18: Comparison between the PSDs of the Signals

1.1.2.5 Interpreting the Spectra and the Magnitude Response of the Filter

The order of the optimum Wiener filter for part 2 was 41; hence, the optimum filter's behaviour is quite similar to that of part 1 which had an optimum order of 42. The magnitude response diagram given in Fig. 16 clearly suppresses certain snippets of frequencies more than the nearby frequencies.

When we compare the PSDs (Fig. 18), we can clearly observe that the optimum Wiener filter had suppressed the higher frequency noise components to a reasonable level. However, its performance is nearly identical to that of the optimum Wiener filter obtained in part 1 and accordingly, it has not managed to eliminate the spike at the 50 Hz division as well.

1.2 Frequency Domain Implementation of the Wiener Filter

Let's define the Fourier transforms of the signals and window to be $\mathcal{F}[x(n)] = X(f)$, $\mathcal{F}[y(n)] = Y(f)$, $\mathcal{F}[\hat{y}(n)] = \hat{Y}(f)$ and $\mathcal{F}[w(n)] = W(f)$.

$$\begin{aligned}
\hat{Y}(f) &= W(f)X(f) \\
e(f) &= Y(f) - \hat{Y}(f) \\
e(f) &= Y(f) - W(f)X(f) \\
E[|e(f)|^2] &= E[(Y(f) - W(f)X(f))(Y(f) - W(f)X(f))^*]
\end{aligned}$$

Define $S_{XY}(f)$ to be the cross-spectral density between the input signal and the desired signal and $S_{XX}(f)$ be the power spectral density of the input signal.

$$\begin{aligned}
\frac{\partial E[|e(f)|^2]}{\partial W(f)} &= 2W(f)S_{XX}(f) - 2S_{XY}(f) = 0 \\
W(f) &= \frac{S_{XY(f)}}{S_{XX}(f)} \qquad (5)
\end{aligned}$$

When $X(n) = Y(n) + N(n)$,

$$S_{XX}(f) = S_{YY}(f) + S_{NN}(f) \qquad (6)$$

$$S_{XY}(f) = S_{YY}(f) \qquad (7)$$

By substituting Eq. 6 and Eq. 7 in Eq.5,

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

1.2.1 Implementing the Wiener Filter in Frequency Domain

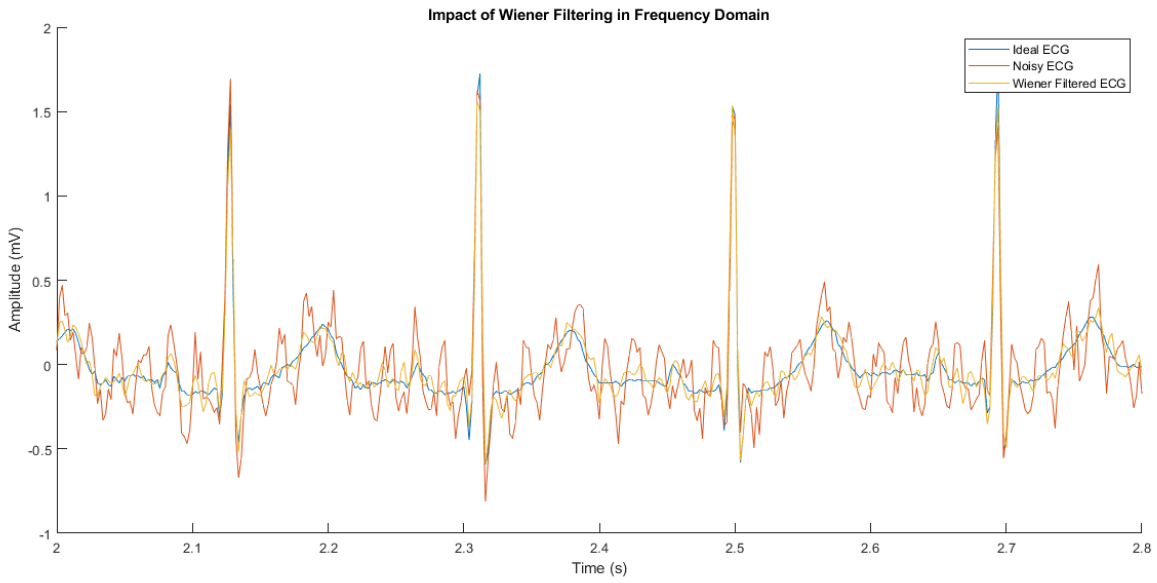

Figure 19: Impact of Wiener Filtering in Frequency Domain

As we can see from Fig. 19, the frequency domain Wiener filtering has managed to recover the signal better than the corresponding discrete-time domain Wiener filters. We can solidify our premise using the PSDs of the signals.

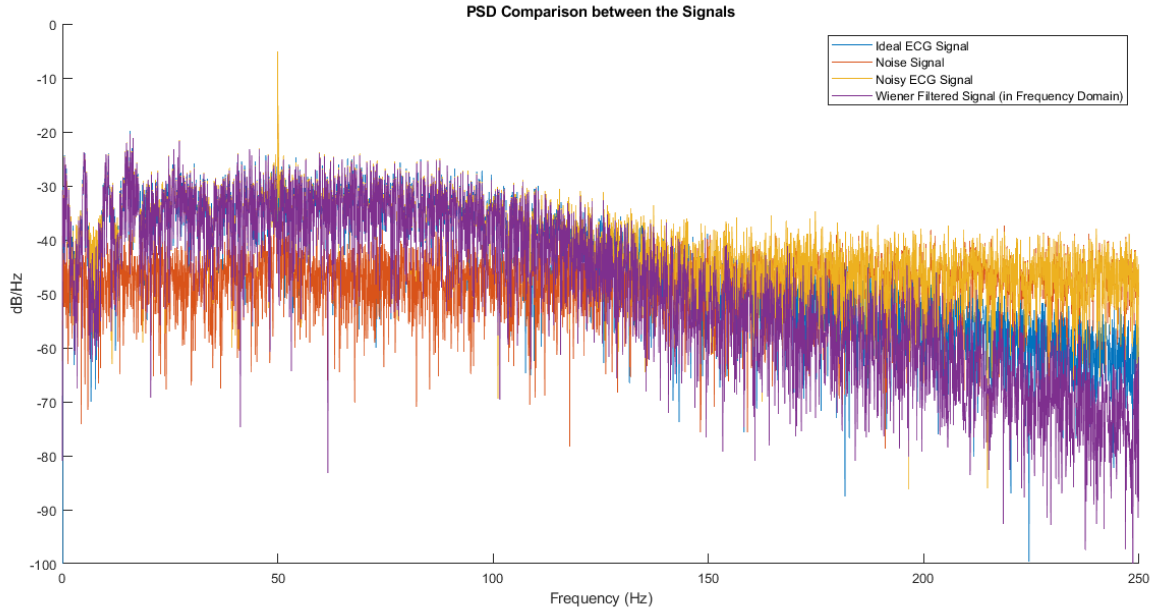

Figure 20: Comparison between the PSDs of the Signals

It is evident from Fig. 20 that the Wiener filter in the frequency domain has managed to suppress the noise spike at 50Hz which was not removed when the noisy signal was filtered using the Wiener filters in the time domain.

1.2.2 Comparing Frequency Domain Filtered Signal with Time Domain Filtered Signals

The following table summarizes the mean squared errors between the filtered signals and the desired signal.

Type of the Filtered Signal	MSE
Wiener Filtered Signal in Time Domain (Order 42)	0.0134
Wiener Filtered Signal in Time Domain (Order 41)	0.0112
Wiener Filtered Signal in Frequency Domain	0.0034

Table 6: Comparison of MSEs

Clearly, the frequency domain implementation results in better filtering. Moreover, frequency domain implementation is stable and reliable as the time domain implementation can lead to singular matrix inversion or nearly singular matrix inversions which could yield non-desirable outputs.

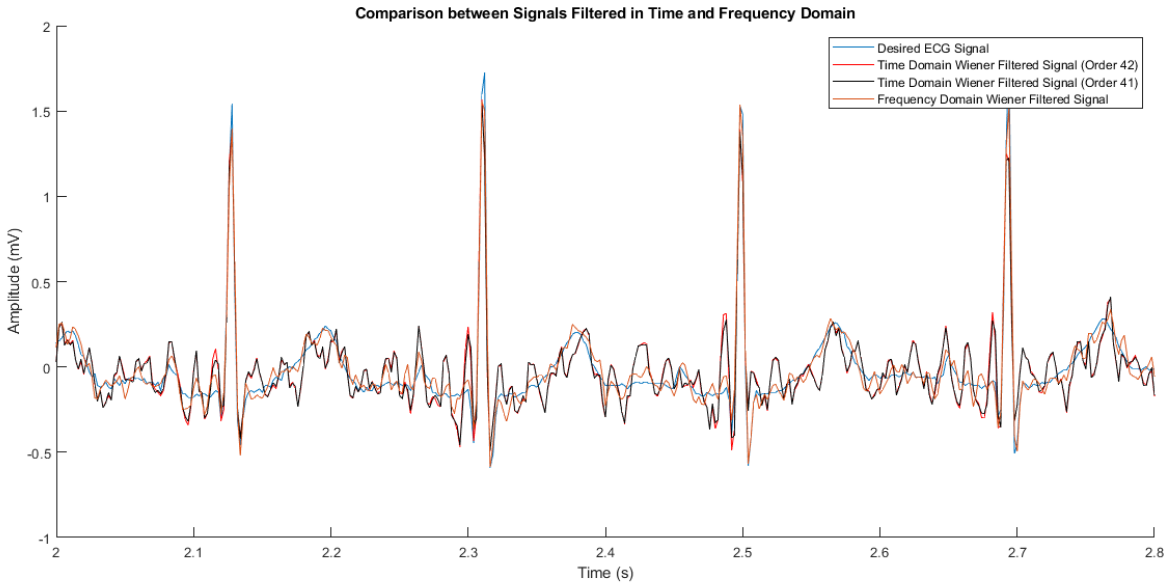

Figure 21: Comparison of Wiener Filtering in Time and Frequency Domain

1.3 Effect of Non-stationary Noise on Wiener Filtering

One of the assumptions used when applying Wiener filtering is that the noise signal is stationary. That is, its characteristics don't change over the duration of concern. In this section, we will evaluate the performance of a Wiener filter in the presence of a non-stationary noise signal.

1.3.1 Applying Wiener Filter on the New Noisy ECG Signal

Ideal Signal	$y_i(n)$	idealECG.mat
Sampling Frequency	f_s	500 Hz
White Gaussian Noise	$\eta_{wg}(n)$	SNR is 10dB w.r.t. $y_i(n)$
Sinusoid Noise 1	$\eta_{50}(n)$	$\eta_{50}(n) = 0.2sin(100\pi n)$
Sinusoid Noise 2	$\eta_{100}(n)$	$\eta_{100}(n) = 0.3sin(200\pi n)$
Input signal	$x(n) = y_i(n) + \eta(n)$	$\eta(n) = \eta_{wg}(n) + \eta_{50}(n)$ when $0 \leq n < \frac{T}{2}$
Input signal	$x(n) = y_i(n) + \eta(n)$	$\eta(n) = \eta_{wg}(n) + \eta_{100}(n)$ when $\frac{T}{2} \leq n \leq T$

Table 7: Data Construction Notation - Non-Stationary Noise

Since the performance of the frequency domain Wiener filter was superior, the ECG signal corrupted with the non-stationary noise was filtered using the Wiener filter in the frequency domain. The plots of the filtered signals are provided in the subsequent section and the implementation could be found in the MATLAB script.

1.3.2 Plotting and Interpreting the Filtered Signal

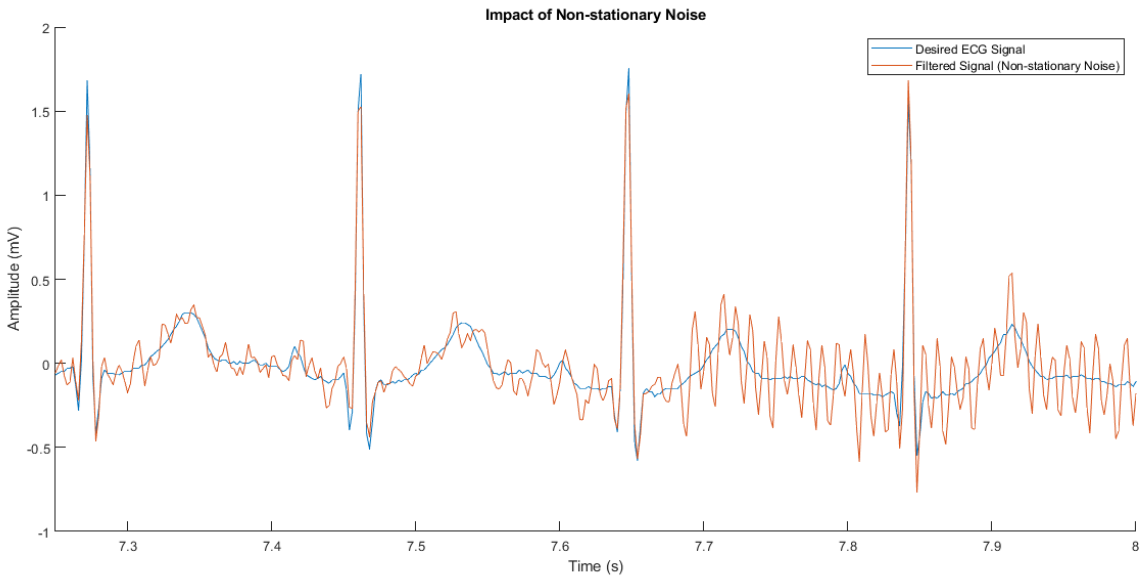

Figure 22: Impact of Wiener Filtering on a Signal Corrupted with Non-stationary Noise

From the above figure, we can observe that the Wiener filter has managed to suppress the noise components up to the mid-duration of the signal. However, it has failed to adapt to the change in the frequency of the noise and we can prove that behaviour from the prominent noise jitter observed in the second half of the filtered signal.

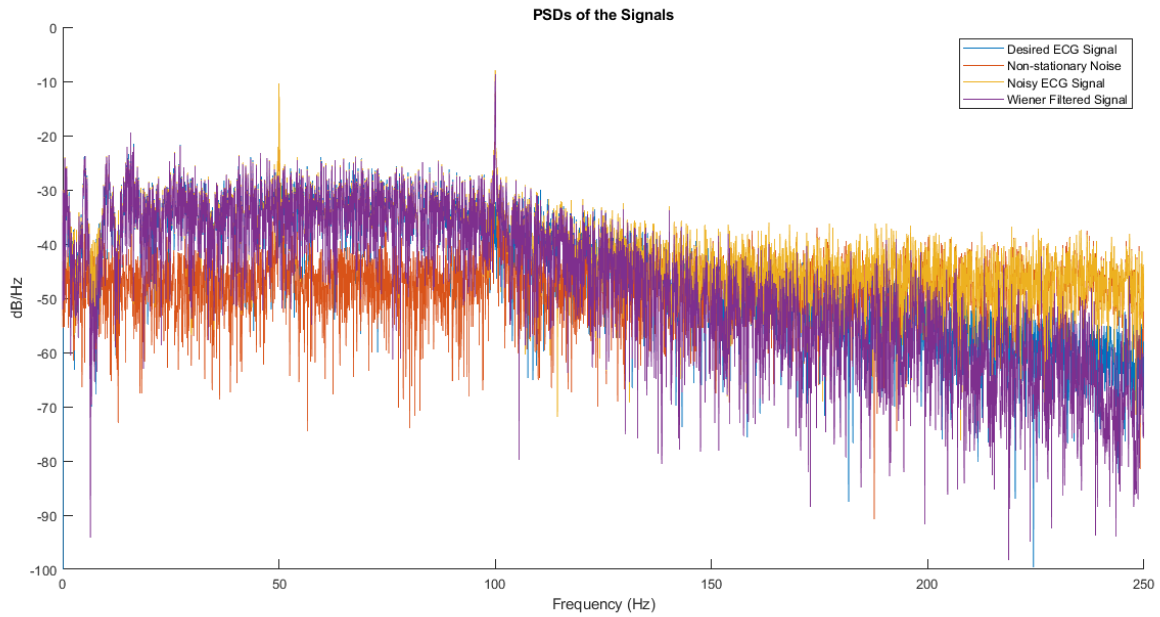

Figure 23: Comparison between the PSDs of the Signals (with Non-stationary Noise)

The PSD plot verifies that the Wiener filter has failed to suppress the 100Hz component albeit the impact of the Wiener filter on the signal is quite similar to the previous case. However, the Wiener filter would not be the best option in the presence of non-stationary noise signals.

2 Adaptive Filtering

The following notation will be used during the data construction.

Sawtooth Waveform	$y_i(n)$	Width is 0.5
Sampling Frequency	f_s	500 Hz
Number of Samples	N	5000
White Gaussian Noise	$\eta_{wg}(n)$	SNR is a constant w.r.t. $y_i(n)$
Sinusoid Noise 1	$\eta_{50}(n)$	$\eta_{50}(n) = 0.2sin(100\pi n)$
Sinusoid Noise 2	$\eta_{100}(n)$	$\eta_{100}(n) = 0.3sin(200\pi n)$
Noise signal	$\eta(n) = \eta_{50}(n)$	when $0 \leq n < \frac{T}{2}$
Noise signal	$\eta(n) = \eta_{100}(n)$	when $\frac{T}{2} \leq n \leq T$
Reference signal	$r(n)$	$a(\eta_{wg}(n) + \eta_{50}(n)(nf_{50} + \phi_1) + \eta_{100}(n)(nf_{100} + \phi_2))$

Table 8: Data Construction for Adaptive Filtering

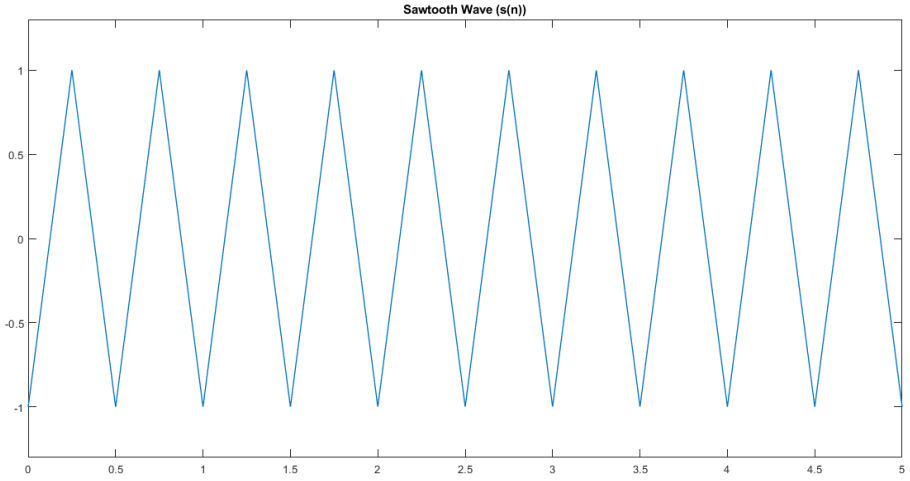

Figure 24: Noise-less Sawtooth Signal

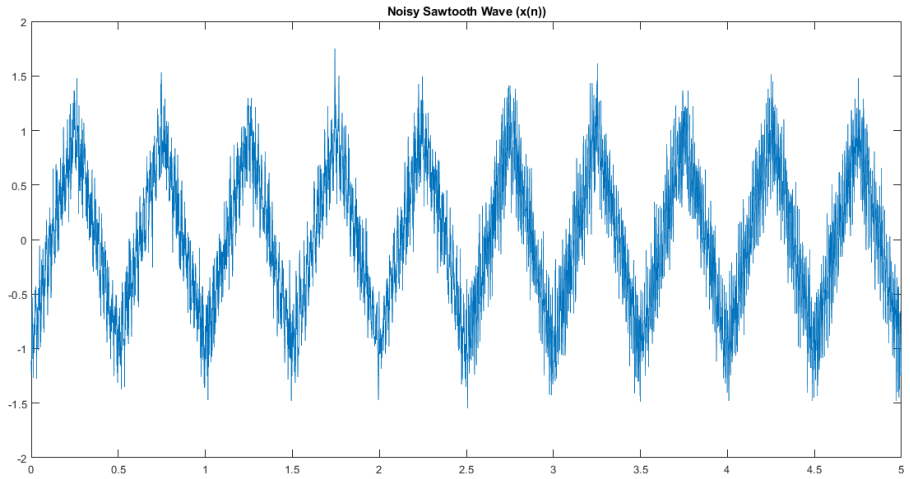

Figure 25: Noisy Sawtooth Signal

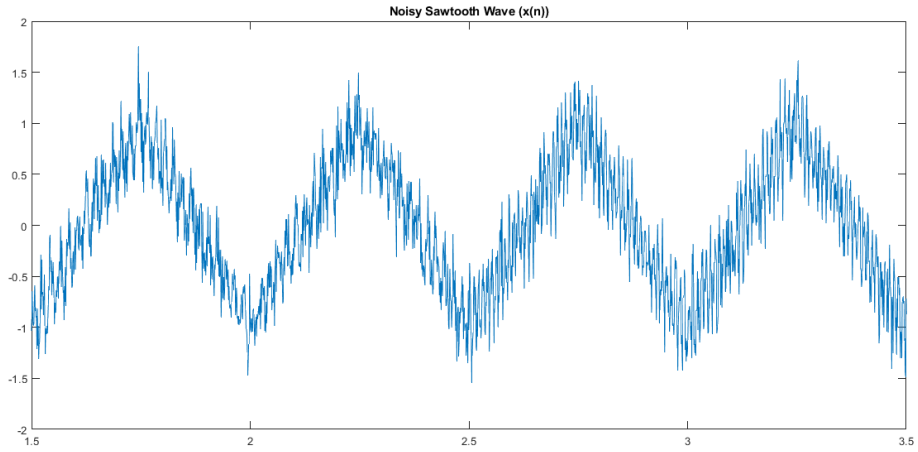

Figure 26: Noisy Sawtooth Signal (Zoomed in)

2.1 Least Mean Square (LMS) Method

$$
\begin{aligned}
y(n) &= \sum_{k=0}^{M-1} w_k r(n-k) && (order = M-1)\\
y(n) &= \boldsymbol{w}^T(n)\boldsymbol{R}(n)\\
e(n) &= x(n) - y(n)\\
J(\boldsymbol{w}(n)) &= E\left[e^2(n)\right]\\
J(\boldsymbol{w}(n)) &= E\left[(x(n) - y(n))^2\right]\\
&= E[(x(n) - \boldsymbol{w}^T(n)\boldsymbol{R}(n))(x(n) - \boldsymbol{R}^T(n)\boldsymbol{w}(n)]\\
J(\boldsymbol{w}(n)) &= x^2(n) - 2x(n)\boldsymbol{w}^T(n)\boldsymbol{R}(n) + \boldsymbol{w}^T(n)\boldsymbol{R}(n)\boldsymbol{R}^T(n)\boldsymbol{w}(n)
\end{aligned}
$$

Then, let's take the partial derivatives.

$$
\begin{aligned}
\nabla(n) &= \frac{\partial J(\boldsymbol{w}(n))}{\partial \boldsymbol{w}(n)}\\
&= \frac{\partial(x^2(n) - 2x(n)\boldsymbol{w}^T(n)\boldsymbol{R}(n) + \boldsymbol{w}^T(n)\boldsymbol{R}(n)\boldsymbol{R}^T(n)\boldsymbol{w}(n))}{\partial \boldsymbol{w}(n)}\\
&= -2x(n)\boldsymbol{R}(n) + 2\boldsymbol{w}^T(n)\boldsymbol{R}(n)\boldsymbol{R}^T(n)\\
\nabla(n) &= -2e(n)\boldsymbol{R}^T(n)
\end{aligned}
$$

Then, let's apply the above result in the steepest descent equation.

$$
\begin{aligned}
\boldsymbol{w}(n+1) &= \boldsymbol{w}(n) - \mu\nabla(n)\\
\boldsymbol{w}(n+1) &= \boldsymbol{w}(n) + 2\mu e(n)\boldsymbol{R}^T(n)
\end{aligned}
$$

2.1.1 Exploring the Rate of Adaptation

The reference signal was assigned the following signal.

$$r(n) = 3\left(\eta_{wg}(n) + \eta_{50}(n)\left(nf_{50} + \frac{\pi}{3}\right) + \eta_{100}(n)\left(nf_{100} + \frac{\pi}{6}\right)\right)$$

Then, the MSE values were calculated over a range of orders and μ to find the optimum parameter values. The following surf plot was obtained consequently.

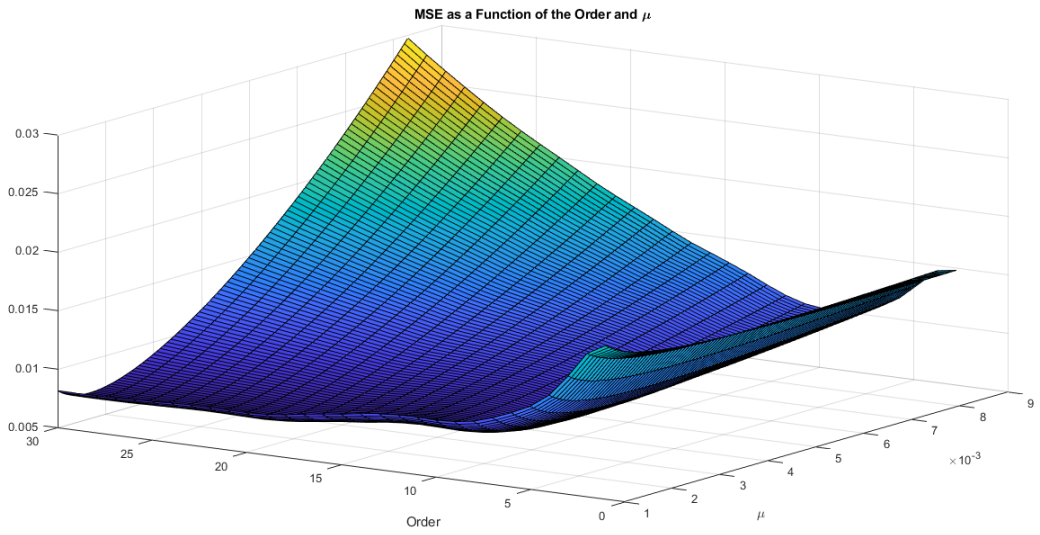

Figure 27: Variation of MSE against M and μ - LMS Method

Accordingly, the parameters that minimize the MSE and the minimum MSE are,

Order (M)	**20**
Step Size (μ)	**0.001949**
Minimum Error	**0.0070289**

Table 9: Optimum Parameter Values and the Corresponding MSE - LMS Method

For the aforementioned parameters, the signal shown below could be obtained once the noisy signal is filtered using the adaptive filter (LMS method).

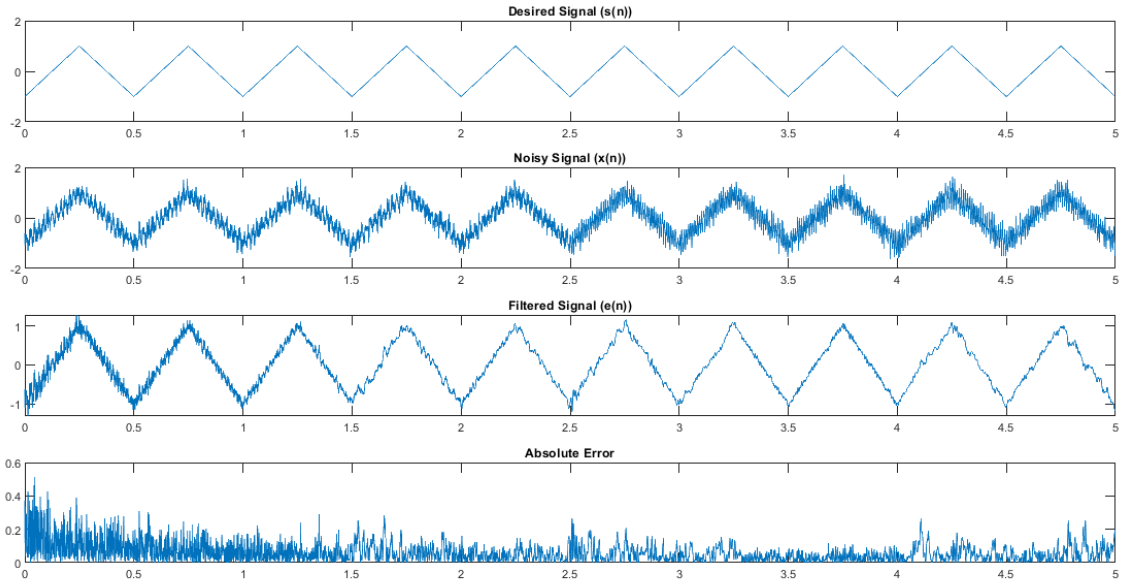

Figure 28: Adaptive Filtered Signal under the Optimum Parameters - LMS Method

2.2 Recursive Least Squares (RLS) Method

The objective of RLS is to minimize the performance index which is the weighted sum of the previous squared errors by finding the corresponding weights while satisfying the Wiener Hopf equation.

$$\begin{aligned}
\xi(n) &= \sum_{i=1}^{n} \lambda^{n-i} e^2(i) \\
\boldsymbol{w}(n) &= \boldsymbol{\Phi_R}^{-1}(n)\boldsymbol{\Theta_{Rx}}(n) \\
\boldsymbol{\Phi_R}(n) &= \sum_{i=1}^{n} \lambda^{n-i} \boldsymbol{R}(i)\boldsymbol{R}^T(i) \\
\boldsymbol{\Theta_{Rx}}(n) &= \sum_{i=1}^{n} \lambda^{n-i} \boldsymbol{R}(i)x(i)
\end{aligned}$$

Let's derive a recursive expression by equating i to n.

$$\begin{aligned}
\boldsymbol{\Phi_R}(n) &= \sum_{i=1}^{n-1} \lambda^{n-i} \boldsymbol{R}(i)\boldsymbol{R}^T(i) + \boldsymbol{R}(n)\boldsymbol{R}^T(n) \\
&= \lambda \sum_{i=1}^{n-1} \lambda^{n-i-1} \boldsymbol{R}(i)\boldsymbol{R}^T(i) + \boldsymbol{R}(n)\boldsymbol{R}^T(n) \\
\boldsymbol{\Phi_R}(n) &= \lambda \boldsymbol{\Phi_R}(n-1) + \boldsymbol{R}(n)\boldsymbol{R}^T(n) \qquad (8)
\end{aligned}$$

$$\begin{aligned}
\boldsymbol{\Theta_{Rx}}(n) &= \sum_{i=1}^{n-1} \lambda^{n-i} \boldsymbol{R}(i)x(i) + \boldsymbol{R}(n)x(n) \\
&= \lambda \sum_{i=1}^{n-1} \lambda^{n-i-1} \boldsymbol{R}(i)x(i) + \boldsymbol{R}(n)x(n) \\
\boldsymbol{\Theta_{Rx}}(n) &= \lambda \boldsymbol{\Theta_{Rx}}(n-1) + \boldsymbol{R}(n)x(n) \qquad (9)
\end{aligned}$$

By applying ABCD lemma to inverse $\boldsymbol{\Phi_R}(n)$.

$$\boldsymbol{\Phi_R}^{-1}(n) = \lambda^{-1}\boldsymbol{\Phi_R}^{-1}(n-1) - \frac{\lambda^{-1}\boldsymbol{\Phi_R}^{-1}(n-1)\boldsymbol{R}(n)\boldsymbol{R}^T(n)\lambda^{-1}\boldsymbol{\Phi_R}^{-1}(n-1)}{\boldsymbol{R}^T(n)\lambda^{-1}\boldsymbol{\Phi_R}^{-1}(n-1)\boldsymbol{R}(n) + 1}$$

Take $\boldsymbol{P}(n) = \boldsymbol{\Phi_R}^{-1}(n)$.

$$\boldsymbol{K}(n) = \frac{\lambda^{-1}\boldsymbol{P}(n-1)\boldsymbol{R}(n)}{1 + \lambda^{-1}\boldsymbol{R}^T(n)\boldsymbol{P}(n-1)\boldsymbol{R}(n)} \qquad (10)$$

$$\boldsymbol{P}(n) = \lambda^{-1}\boldsymbol{P}(n-1) - \lambda^{-1}\boldsymbol{K}(n)\boldsymbol{R}^T(n)\boldsymbol{P}(n-1) \qquad (11)$$

$$\boldsymbol{K}(n) = \underbrace{[\lambda^{-1}\boldsymbol{P}(n-1) - \lambda^{-1}\boldsymbol{K}(n)\boldsymbol{R}^T(n)\boldsymbol{P}(n-1)]}_{\boldsymbol{P}(n)}\boldsymbol{R}(n) \qquad [\textit{Rearranging Eq.}10]$$

$$\boldsymbol{K}(n) = \boldsymbol{P}(n)\boldsymbol{R}(n) \qquad (12)$$

Substituting these results in the Wiener-Hopf equation.

$$
\begin{aligned}
\boldsymbol{w}(n) &= \boldsymbol{\Phi_R}^{-1}(n)\boldsymbol{\Theta_{Rx}}(n) \\
&= \boldsymbol{P}(n)\boldsymbol{\Theta_{Rx}}(n) \\
&= \boldsymbol{P}(n)[\lambda\boldsymbol{\Theta_{Rx}}(n-1) + \boldsymbol{R}(n)x(n)] && [\textit{From Eq. } 9] \\
&= \boldsymbol{P}(n-1)\boldsymbol{\Theta_{Rx}}(n-1) - \boldsymbol{K}(n)\boldsymbol{R}^T(n)\boldsymbol{P}(n-1)\boldsymbol{\Theta_{Rx}}(n-1) + \boldsymbol{P}(n)\boldsymbol{R}(n)x(n) && [\textit{From Eq. } 11] \\
&= \boldsymbol{\Phi_R}^{-1}(n-1)\boldsymbol{\Theta_{Rx}}(n-1) - \boldsymbol{K}(n)\boldsymbol{R}^T(n)\boldsymbol{\Phi_R}^{-1}(n)\boldsymbol{\Theta_{Rx}}(n-1) + \boldsymbol{P}(n)\boldsymbol{R}(n)x(n) \\
&= \boldsymbol{w}(n-1) - \boldsymbol{K}(n)\boldsymbol{R}^T(n)\boldsymbol{w}(n-1) + \boldsymbol{P}(n)\boldsymbol{R}(n)x(n) \\
&= \boldsymbol{w}(n-1) - \boldsymbol{K}(n)\boldsymbol{R}^T(n)\boldsymbol{w}(n-1) + \boldsymbol{K}(n)x(n) \\
\boldsymbol{w}(n) &= \boldsymbol{w}(n-1) + \boldsymbol{K}(n)\underbrace{[x(n) - \boldsymbol{R}^T(n)\boldsymbol{w}(n-1)]}_{\alpha(n)} \\
\boldsymbol{w}(n) &= \boldsymbol{w}(n-1) + \boldsymbol{K}(n)\alpha(n) && (13)
\end{aligned}
$$

In summary,

$$
\begin{aligned}
\boldsymbol{K}(n) &= \frac{\lambda^{-1}\boldsymbol{P}(n-1)\boldsymbol{R}(n)}{1 + \lambda^{-1}\boldsymbol{R}^T(n)\boldsymbol{P}(n-1)\boldsymbol{R}(n)} \\
\boldsymbol{P}(n) &= \lambda^{-1}\boldsymbol{P}(n-1) - \lambda^{-1}\boldsymbol{K}(n)\boldsymbol{R}^T(n)\boldsymbol{P}(n-1) \\
\alpha(n) &= x(n) - \boldsymbol{R}^T(n)\boldsymbol{w}(n-1) = x(n) - \boldsymbol{w}^T(n-1)\boldsymbol{R}(n) \\
\boldsymbol{w}(n) &= \boldsymbol{w}(n-1) + \boldsymbol{K}(n)\alpha(n)
\end{aligned}
$$

2.2.1 Exploring the Rate of Adaptation

Initializations:

$$
\begin{aligned}
\boldsymbol{P}(0) &= \boldsymbol{\delta}^{-1}I \\
\boldsymbol{w}(0) &= \mathbf{0}
\end{aligned}
$$

Then, the MSE values were calculated over a range of orders and λ to find the optimum parameter values. The following surf plot was obtained consequently.

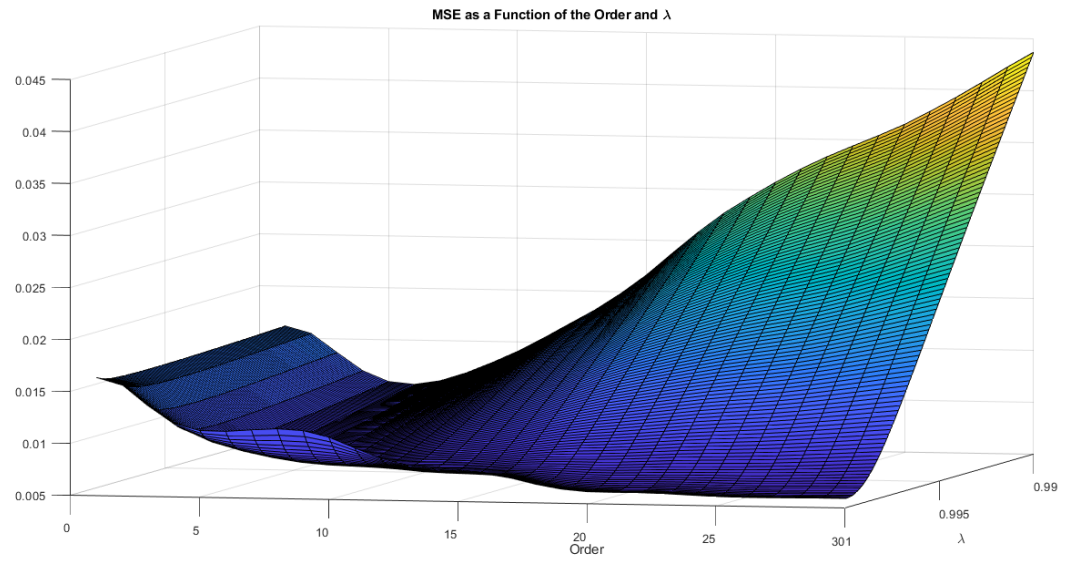

Figure 29: Variation of MSE against M and λ - RLS Method

Accordingly, the parameters that minimize the MSE and the minimum MSE are,

Order (M)	**26**
Forgetting Factor (λ)	**0.9999**
Minimum Error	**0.0058817**

Table 10: Optimum Parameter Values and the Corresponding MSE - RLS Method

For the aforementioned parameters, the signal shown below could be obtained once the noisy signal is filtered using the adaptive filter (RLS method).

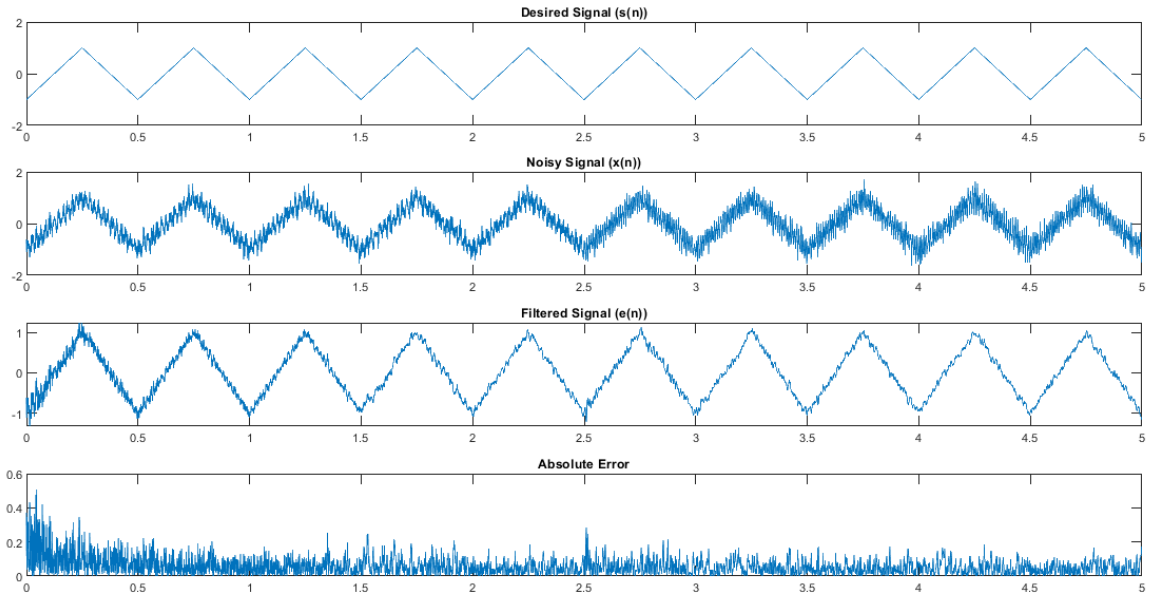

Figure 30: Adaptive Filtered Signal under the Optimum Parameters - RLS Method

2.2.2 Comparing the Performance of the LMS and RLS Algorithms

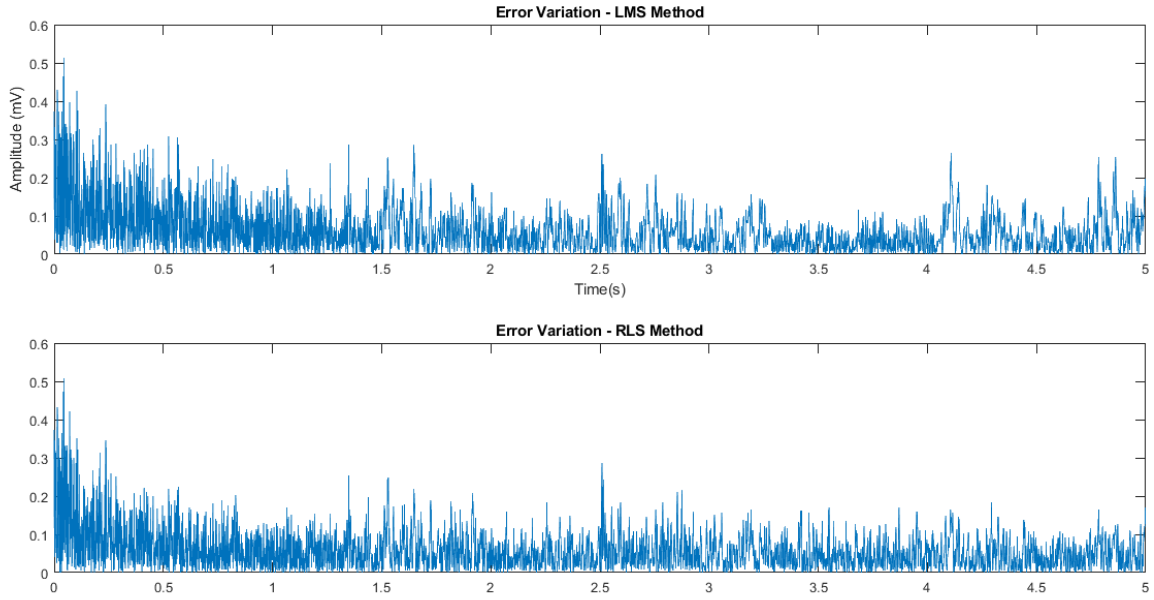

Figure 31: Comparison between LMS and RLS Algorithms

From the above figure, we can see that the RLS algorithm has adapted to the signal slightly better than the LMS algorithm. The reason why the difference is not clearly visible may be due to the specific noise component that was added. However, to verify that the RLS performs better even in this context, the errors throughout the samples were compared.

$$Percentage\ of\ samples = \frac{No.\ of\ samples\ where\ RLS\ error\ is\ less}{total\ no.\ of\ samples} \times 100\%$$

From the above calculation, it was revealed that in 55% of the samples, RLS performs better.

2.2.3 Testing LMS and RLS using $y_i(n)$ (Ideal ECG Signal)

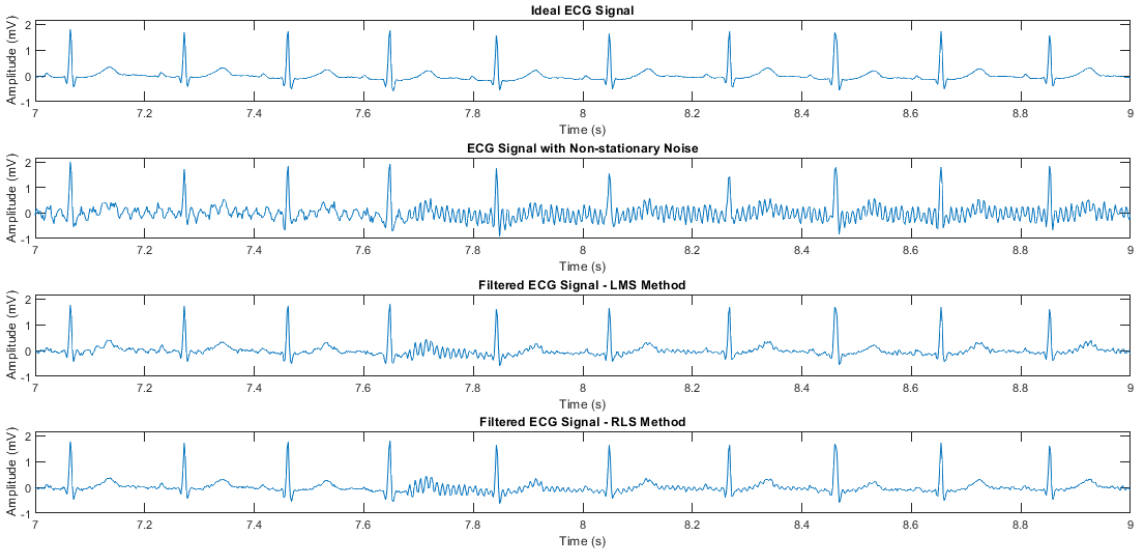

Figure 32: Impact of LMS and RLS Filtering on the ECG Signal with Non-stationary Noise

Clearly, both methods have adapted well to that change in noise characteristics. If we compare the latter portions of the two bottom plots, we can see that the RLS algorithm has adapted slightly faster than the LMS algorithm.