

### 3 Adaptive Signal Processing

#### 3.1 The Least Mean Square (LMS) Algorithm

a. The AR(2) process defined in the coursework has the following general form:

$$x(n) = a_1 x(n-1) + a_2 x(n-2) + \eta(n) \quad (7)$$

where  $\eta(n) \sim \mathcal{N}(0, \sigma_\eta^2)$ . Since the correlation matrix  $\mathbf{R}_x = \mathbb{E}\{\mathbf{x}(n)\mathbf{x}(n)^T\}$ , where  $\mathbf{x}(n) = [x(n-1), x(n-2)]^T$ , the entries of the matrix are:

$$\mathbf{R}_x = \begin{bmatrix} \mathbb{E}\{x(n-1)x(n-1)\} & \mathbb{E}\{x(n-1)x(n-2)\} \\ \mathbb{E}\{x(n-2)x(n-1)\} & \mathbb{E}\{x(n-2)x(n-2)\} \end{bmatrix}$$

In order to determine the entries, we multiply (7) with  $x(n-k)$  and obtain:

$$x(n)x(n-l) = a_1 x(n-1)x(n-k) + a_2 x(n-2)x(n-l) + \eta(n)x(n-k)$$

By taking expectations, we get:

$$\gamma(k) = \mathbb{E}\{x(n)x(n-k)\} = a_1 \mathbb{E}\{x(n-1)x(n-k)\} + a_2 \mathbb{E}\{x(n-2)x(n-k)\} + \mathbb{E}\{\eta(n)x(n-k)\}$$

The above equation can be simplified and 3 simultaneous equations can be generated to obtain the entries of the correlation matrix:

$$\begin{aligned} \gamma(0) &= a_1 \gamma(1) + a_2 \gamma(2) + \sigma_\eta^2 \\ \gamma(1) &= a_1 \gamma(0) + a_2 \gamma(1) \\ \gamma(2) &= a_1 \gamma(1) + a_2 \gamma(0) \end{aligned}$$

Solving the simultaneous equations with  $a_1 = 0.1$ ,  $a_2 = 0.8$  and  $\sigma_\eta^2 = 0.25$ , we obtain the ACF matrix:

$$\mathbf{R}_x = \begin{bmatrix} \mathbb{E}\{x(n-1)x(n-1)\} & \mathbb{E}\{x(n-1)x(n-2)\} \\ \mathbb{E}\{x(n-2)x(n-1)\} & \mathbb{E}\{x(n-2)x(n-2)\} \end{bmatrix} = \begin{bmatrix} \gamma(0) & \gamma(1) \\ \gamma(1) & \gamma(0) \end{bmatrix} = \begin{bmatrix} \frac{25}{27} & \frac{25}{54} \\ \frac{25}{54} & \frac{25}{27} \end{bmatrix}$$

To find the range of values for  $\mu$  for which the LMS algorithm will converge in mean, the eigenvalues of  $\mathbf{R}_x$  has to be found and the following inequality in (8) has to be satisfied.

$$0 < \mu < \frac{2}{\lambda_{max}} \quad (8)$$

If the inequality is satisfied, the LMS algorithm will converge to the Winer-Hopf solution. The maximum eigenvalue of  $\mathbf{R}_x$  is 1.3889 and thus as long as  $0 < \mu < 1.44$ , the LMS algorithm will converge in mean.

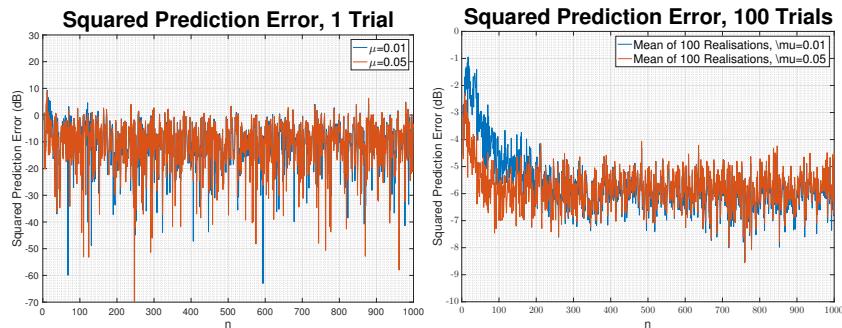


Figure 31: Analysis of Effects of Different Step Size on Convergence Speed of LMS Alogirthm

b. The LMS adaptive predictor using  $N = 1000$  samples of  $x(n)$  was implemented. Results of 1 realization and the average of 100 realizations are shown in the figure above for  $\mu = 0.01$  and  $\mu = 0.05$ . Looked at the squared

prediction error, it is clear that on average we obtain faster convergence when  $\mu = 0.05$ . This is exactly as expected as a larger value of  $\mu$  would allow the algorithm to decent down the error at a faster rate.

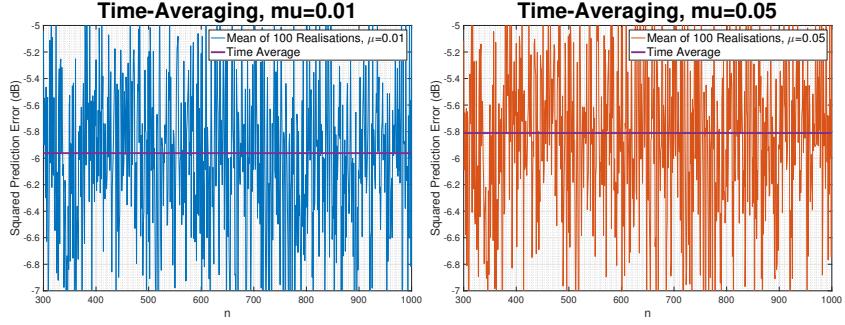


Figure 32: Time-Average of Steady State Error for  $\mu = 0.01$  and  $\mu = 0.05$

c. The figure above shows the steady-state of the ensemble-average learning curves using 100 independent trials of the experiment. Based on the 100 independent realizations, the excess mean square error, misadjustment and theoretical misadjustment are calculated and the results are shown in the table below. The empirical misadjustment are slightly higher than the theoretical values. Regardless, it is abundantly clear that having a smaller step size introduces a much smaller excess mean square error and results in a smaller misadjustment value. However, as observed above, having a smaller step size causes the rate of convergence to be slow.

mu	EMSE	Empirical Misadjustment	Theoretical Misadjustment
0.01	0.0032	0.0128	0.0093
0.05	0.124	0.0496	0.0463

d. The graphs below show how the coefficients evolve over time. For the rest of the section, most graphs will show the evolution of coefficients, averaged over 100 realizations, and thus it is good to get a feel of what the graphs look like for just 1 realization. Real world adaptive filters will more often than not only work with 1 realization of a random signal. Looking more specifically at the evolution of coefficients for  $\mu = 0.05$  for 1 realization, it is clear that speed has been traded off for steady-state error. The amount of variation around the ideal values is significant.

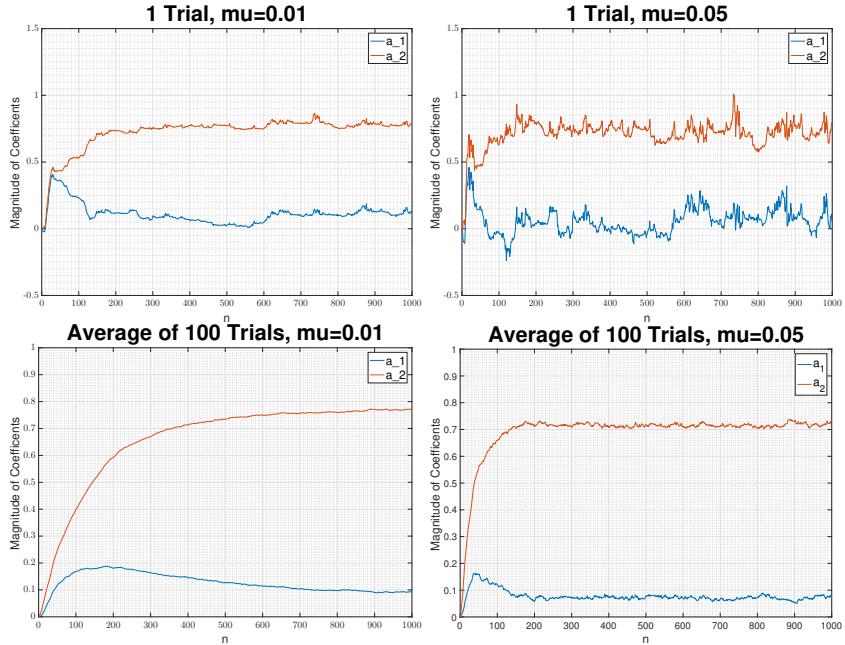


Figure 33: Evolution of Coefficients for 1 Trial and 100 Trial, for  $\mu = 0.01$  and  $\mu = 0.05$

Next, to answer the question directly, we observe Figure 34. It is clear that for both coefficients, having a small value of  $\mu$  results in the steady state value that is much closer to the ideal values of  $a_1 = 0.1$  and  $a_2 = 0.8$ . In addition, having a small value of  $\mu$  results in a much smaller steady-state variance. However, looking at

Figure 33, we find that setting  $\mu = 0.01$ , we do not approach the steady state value until the 900th iteration. In contrast, using  $\mu = 0.05$ , the algorithm converges in about 200 iterations.

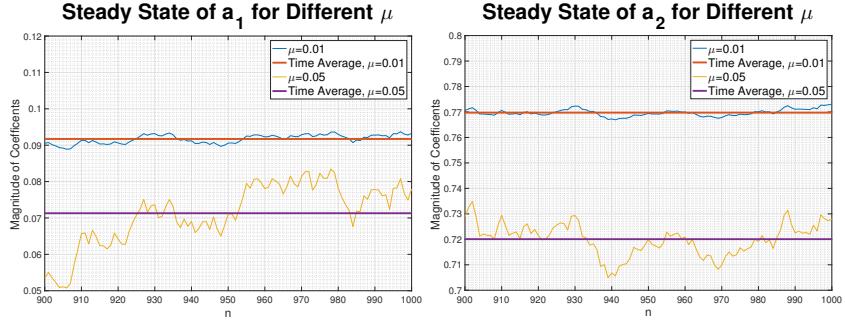


Figure 34: Steady State Values of Coefficients  $a_1$  and  $a_2$ , for  $\mu = 0.01$  and  $\mu = 0.05$

e. In order to minimize the cost function, we differentiate it and set the derivative to 0. Mathematically, we obtain:

$$J(n) = \frac{1}{2} \left( e^2(n) + \gamma \|\mathbf{w}(n)\|^2 \right) \quad (9)$$

$$\frac{\partial J}{\partial \mathbf{w}} = e(n) \frac{\partial e}{\partial \mathbf{w}} + \gamma \mathbf{w}(n) = 0$$

Since,

$$\frac{\partial e}{\partial \mathbf{w}} = \frac{\partial}{\partial \mathbf{w}} \left( \mathbf{x}(n) - \mathbf{w}^T(n) \mathbf{x}(n) \right) = -\mathbf{x}(n)$$

Then,

$$\frac{\partial J}{\partial \mathbf{w}} = -\gamma \mathbf{w}(n) + e(n)$$

Now, using the gradient decent method to reach the optimal value of  $w$ , we have to implement (10)

$$\mathbf{w}(n) = \mathbf{w}(n) - \mu \nabla_w J(n) \quad (10)$$

Substituting the derivative of the cost function, we obtain:

$$\begin{aligned} \mathbf{w}(n+1) &= \mathbf{w}(n) - \mu \left( -\gamma \mathbf{w}(n) + e(n) \right) \mathbf{x}(n) \\ \mathbf{w}(n+1) &= (1 - \mu\gamma) \mathbf{w}(n) - \mu e(n) \mathbf{x}(n) \end{aligned} \quad (11)$$

Thereby showing that the leaky LMS equation shown in the notes and derived in (11) is equivalent to minimizing the cost function shown in (9).

f. The graphs shown in Figure 35 show evolution of coefficients for different values of  $\mu$  and  $\gamma$  using the leaky LMS algorithm. The steady-state values do not converge to the true values of  $a_1$  and  $a_2$ . In fact, the larger the value of  $\gamma$ , the greater the steady-state bias. The Weiner-Hopf solution was briefly mentioned above. It gives the optimal weights calculated using the autocorrelation matrix,  $\mathbf{R}$ , and the cross correlation vector,  $\mathbf{p}$ :

$$\mathbf{w}_{opt} = \mathbf{R}^{-1} \mathbf{p}$$

The LMS, under certain bounds on the value of  $\mu$ , iteratively arrives at the Weiner-Hopf solution. The motivation of the algorithm is to prevent the inverting the autocorrelation matrix. The leaky LMS however does not

converge to the Wiener-Hopf solution. The leaky LMS minimizing a cost function that is slightly different than the cost function that the LMS algorithm minimizes and thus it converges to:

$$\lim_{k \rightarrow \infty} \mathbb{E}[\mathbf{w}_k] = (\mathbf{R} + \gamma \mathbf{I})^{-1} \mathbf{p}$$

The additional,  $\gamma \mathbf{I}$  term causes a bias that means that the leaky-LMS does not actually converge to the Wiener-Hopf solution. In fact, since  $\mathbf{p} = [\gamma(1), \gamma(2)]^T$ , we can mathematically determine the values to which the algorithm converges:

$\gamma$	$a_1$	$a_2$
0.1	0.1542	0.3919
0.5	0.1626	0.4992
0.9	0.1319	0.7076

Notice that the graphs confirm the theoretical results calculated above.

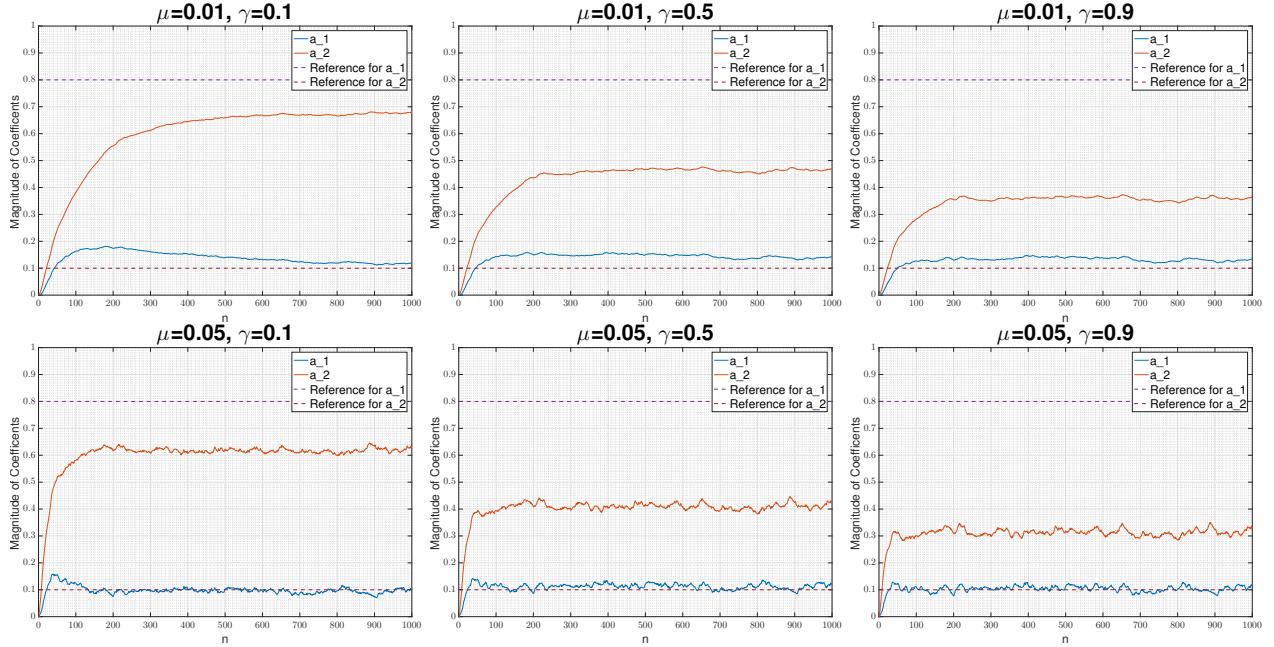


Figure 35: Effect of increasing  $\gamma$  on the Steady State Values of Coefficients  $a_1$  and  $a_2$ , for  $\mu = 0.01$  and  $\mu = 0.05$

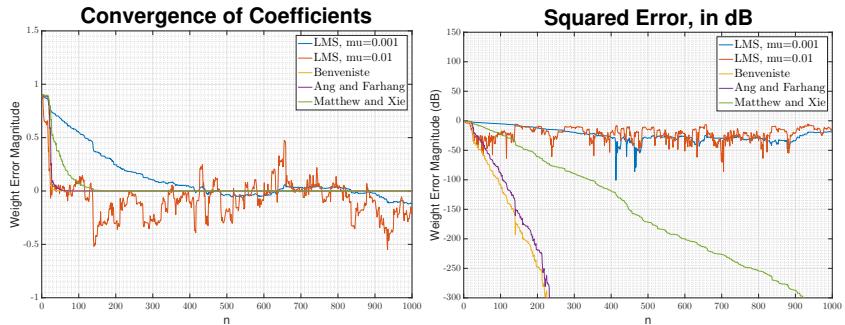


Figure 36: Comparison of Convergence Time using Adaptive Step Sizes and Standard LMS Algorithms

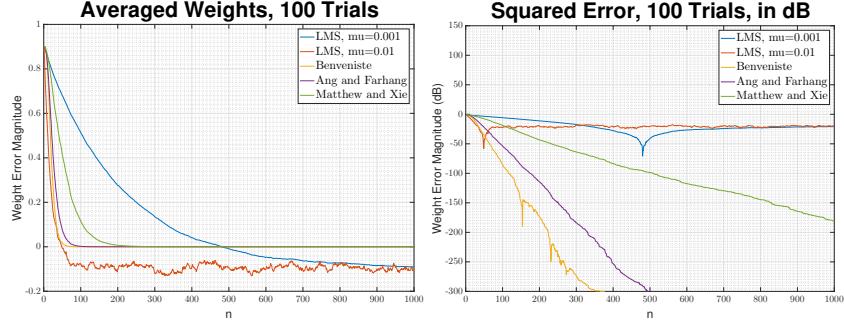


Figure 37: Studying Convergence Time by Averaging Weights over 100 Random Realisations

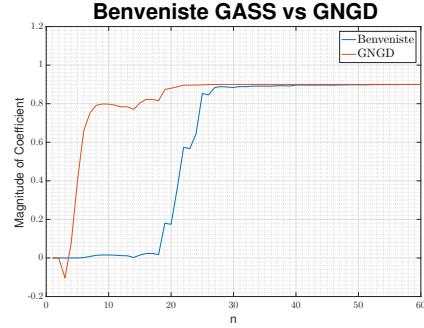


Figure 38: Comparing Convergence Speed of GNGD and Benveniste's GASS Algorithms

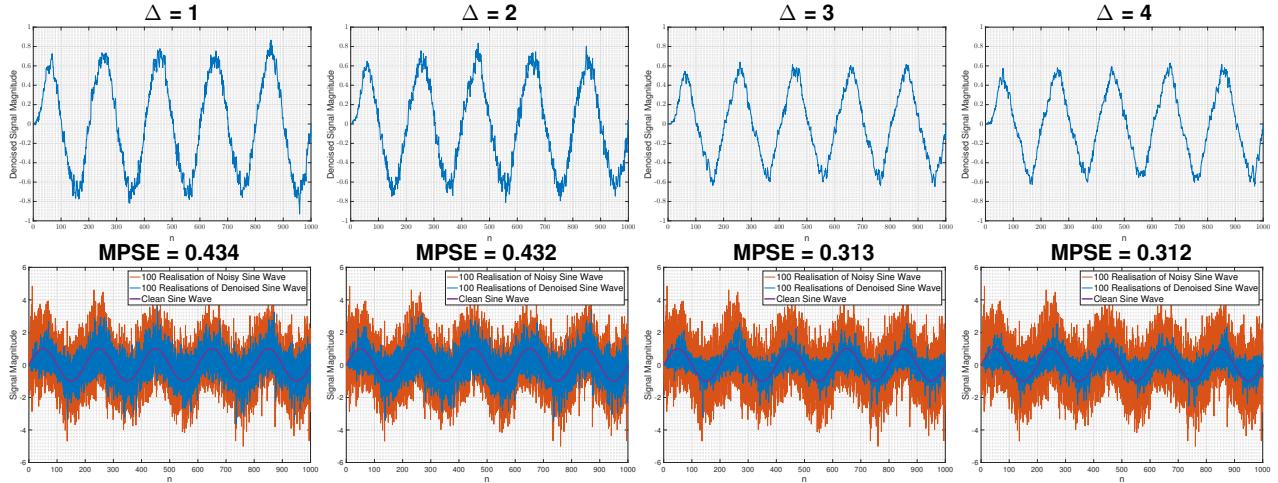


Figure 39: Determining Ideal Value for  $\Delta$  to be used in the Adaptive Line Enhancement Algorithm

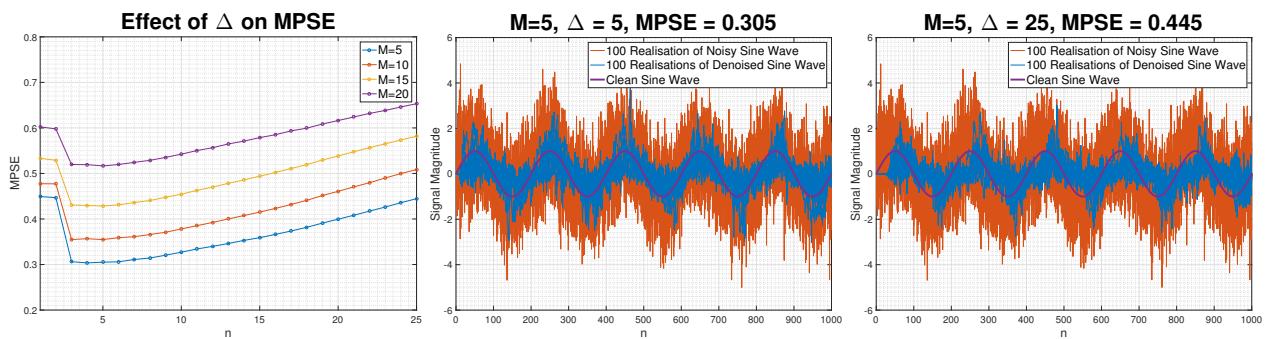


Figure 40: Studying the Effects of Increasing Delay on the MPSE

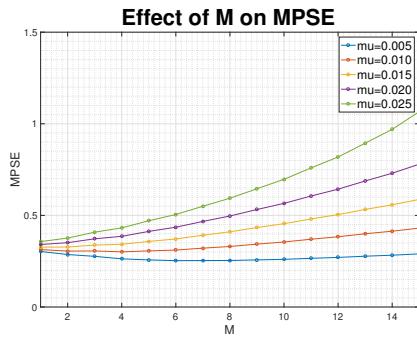


Figure 41: Studying the Effects of Increasing Model Order on the MPSE

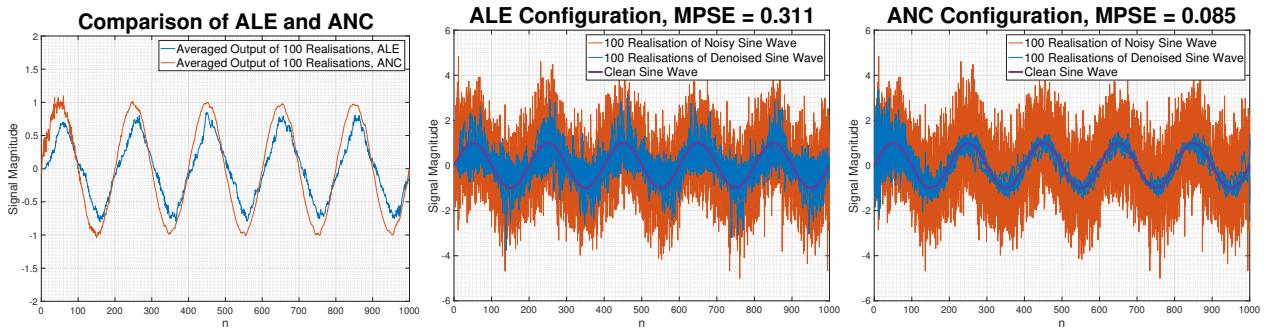


Figure 42: Comparison of ALE and ANC Configurations for Denoising Sinewave

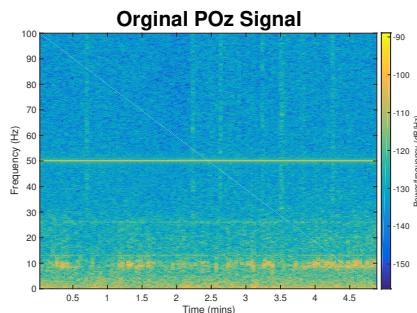


Figure 43: Original EEG Data collected from POz Location on the Scalp with Strong Component at 50 Hz

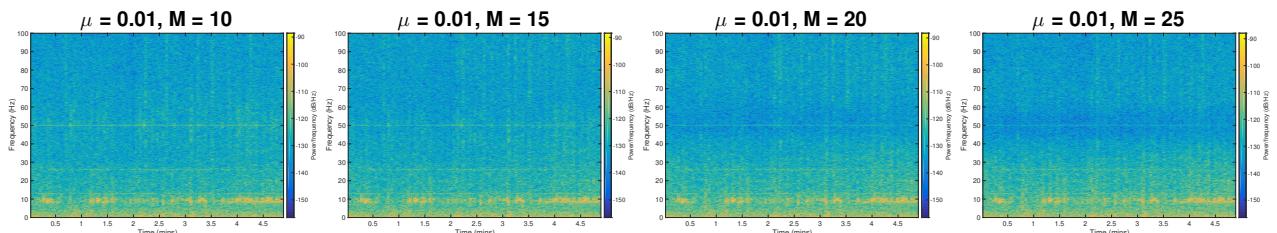


Figure 44: Effect of Increasing Model Order on the Spectrogram of EEG Data

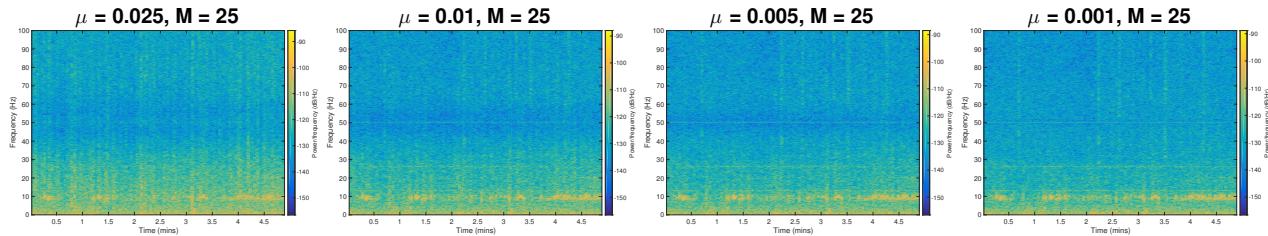


Figure 45: Effect of Varying  $\mu$  on the artifacts observed in the Spectrogram of Denoised EEG Data

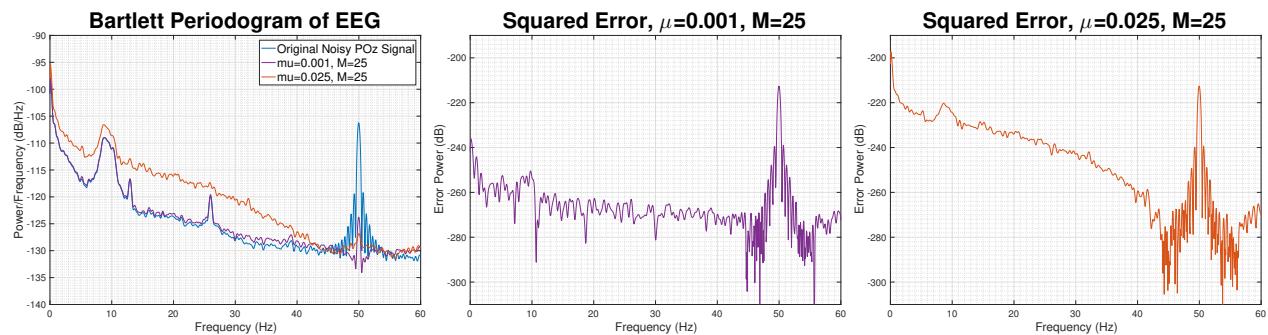


Figure 46: Bartlett Periodogram, averaged over 2 second intervals, and Squared Errors