Kumar Rohit Malhotra (5697-1748)

EEL5840
Elements of Machine Intelligence
Assignment – 3

(All the solutions have been programmed by me using MATLAB.**)**

Section 1: Finding ideal hyper parameters

- We can start by taking values for **Filter Order M** from 1 to 30, with an increment of 1. Along with that, we take **learning rates α** from 0.001 to 0.01, with an increment of 0.001.
- Using the combinations of these hyper parameters and the **training data**, we can find various of weight coefficients.
- We can organize our data to the following form for each filter order M:

$$X = \begin{bmatrix} x(M) & x(M+1) & x(M+2) & \cdots \\ x(M-1) & x(M) & x(M+1) & \cdots \\ \vdots & \vdots & \vdots & \vdots \end{bmatrix}$$

where every ith column of is an input sample for the filter equation

$$y(n) = \sum_{k=1}^{M-1} w(k)x(n-k)$$

to be simply implemented as $y = W^T X$, as the input signals i to i+M-1 would be used to predict the (i+M)th signal.

- For each combination of filter order M and learning rate α, we take the training data and train the system using the following algorithm, starting from i=1:
    1. Initialize the filter coefficients to 0 for the order M
    2. Take the output y for ith sample from X above using formula $y(i+M) = W(i)^T X(i)$.
    3. Find the error $e = d(i+M) - y(i+M)$.
    4. Use the Stochastic gradient formula to determine the new value of W using the formula : $W(i+1) = W(i) + \alpha \mathrm{e}(\mathrm{i}) X(i)$
    5. Go to step 2 unless i=N-M, i.e. till the end of the training input X above.
- These weights, along with the corresponding filter order, are applied on the **Validation data** using the formula $\boldsymbol{y} = \boldsymbol{W}^T\boldsymbol{X}$**,** by organizing the validation data in a similar fashion as given above.
- Here we calculate the Mean Square Error for the various sets of weights on the basis of validation data and the output we get from the formula above. On plotting the Mean Square Errors against the corresponding values of learning rates α and Filter Orders M, we get the following plot in figure 1:

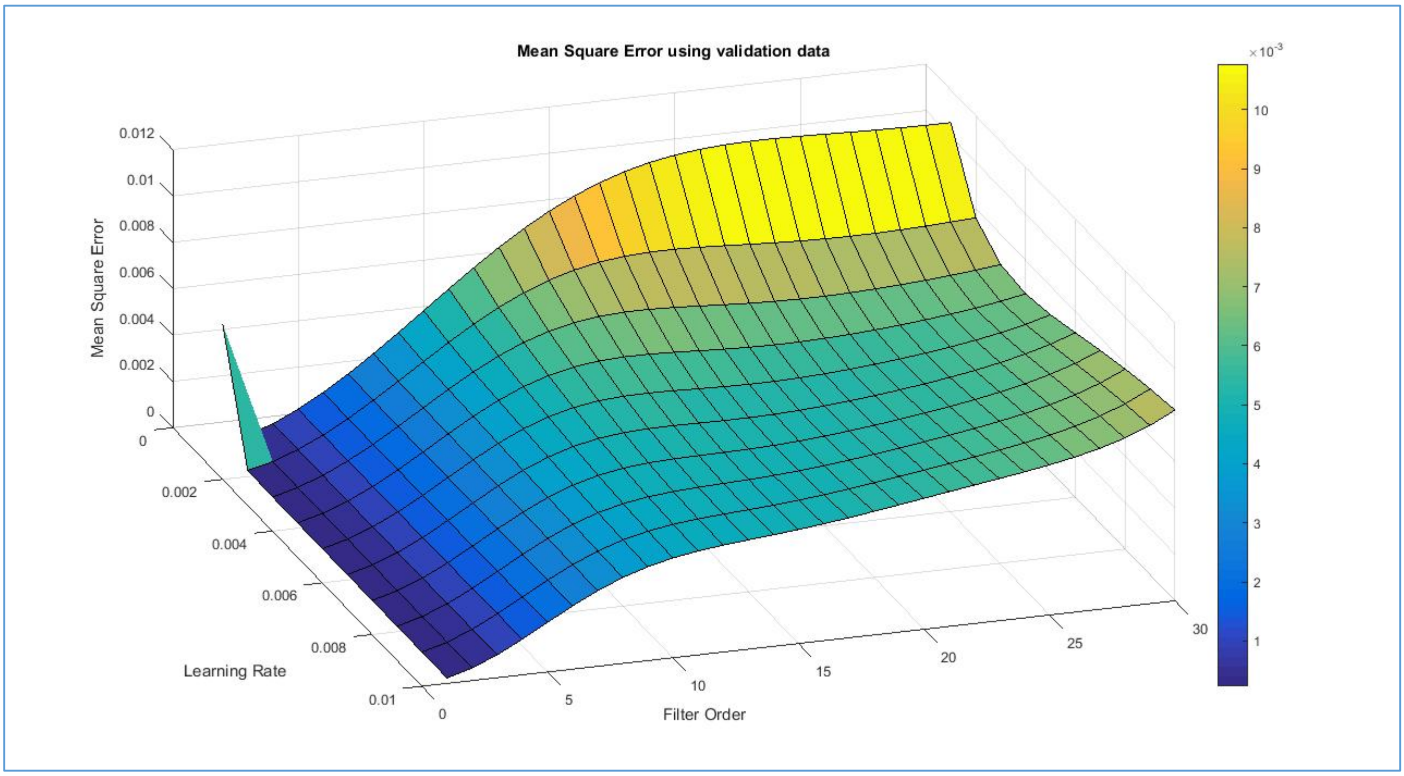

Figure 1. Mean Square Error against Learning rate and Filter order

- The minimum value of Mean Square Error is **0.00023195**, obtained for weights corresponding to Filter Order **M=1** and learning rate = **0.004.**
- As can be seen from the graph, step size is not directly related to the filter order. The step size will only decide the convergence rate of the filter coefficients.

Section 2: Plotting two smaller learning rates against the optimum learning rate

- We take the optimum learning rate and optimum filter order obtained in step 1, and plot the learning curve for **training** data for these hyper – parameters, by using the formulae in section 1 to train the system (just for checking the learning curve).
- Along with that we take two smaller learning rates, and the same filter order as above, to plot their learning curve as per error for each sample using the formulae in section 1 above.
- We obtain the following learning curves for the three learning rates:

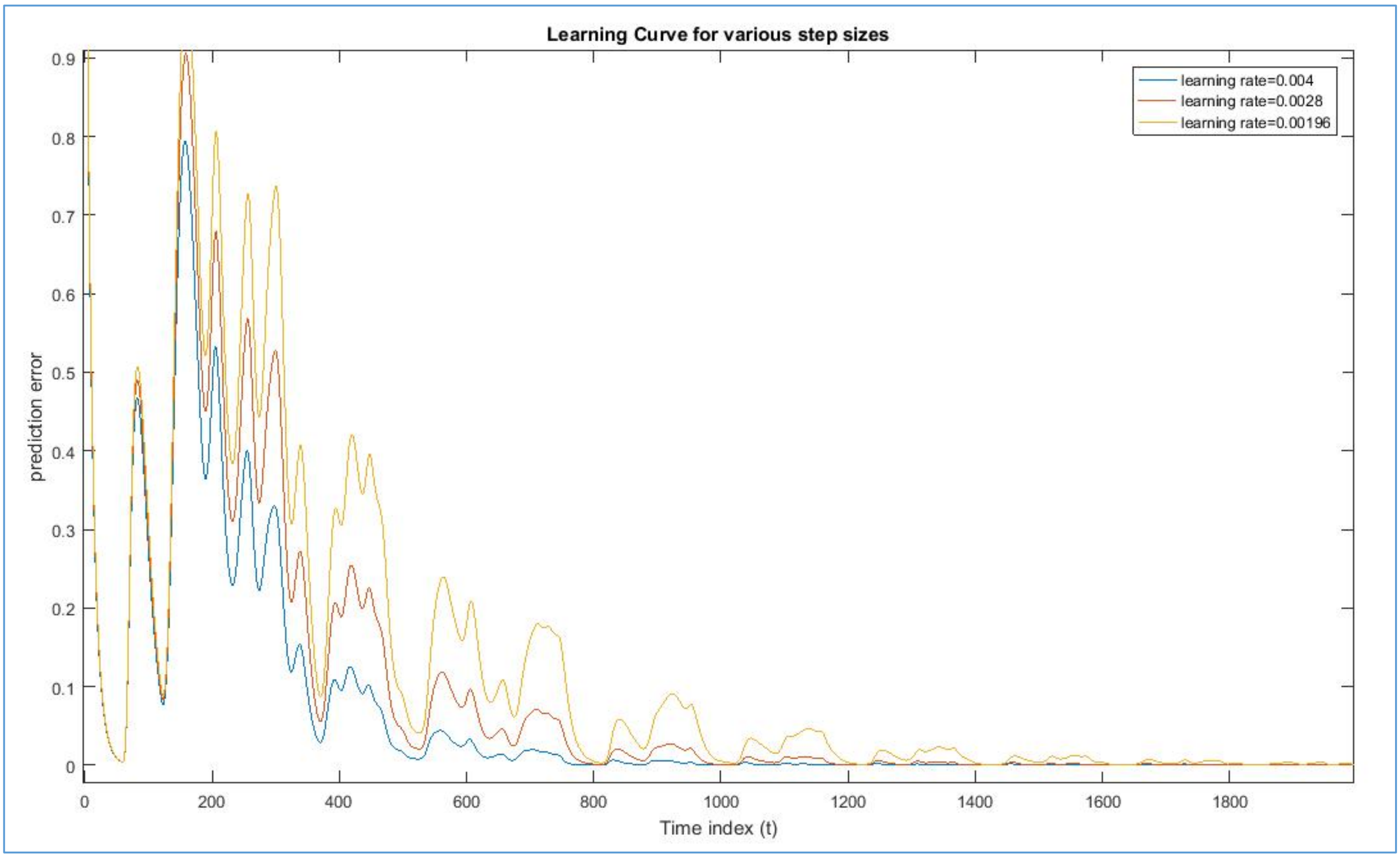

Figure 2. Learning curve for three different learning rates using same filter order

- As can be seen in the graph above, the rate of convergence for the selected optimum learning rate viz. 0.004 is faster than the two learning rates less than that. The MSE for learning rate approximates to MSE_{min} at around 800^{th} sample.

- Using the selected filter order and the corresponding filter coefficients obtained from section 1, we predict the signal using the M previous input signals for each output, and plot it against the desired signal. This is done on the **test** data. We obtain the following plot in figure 3:

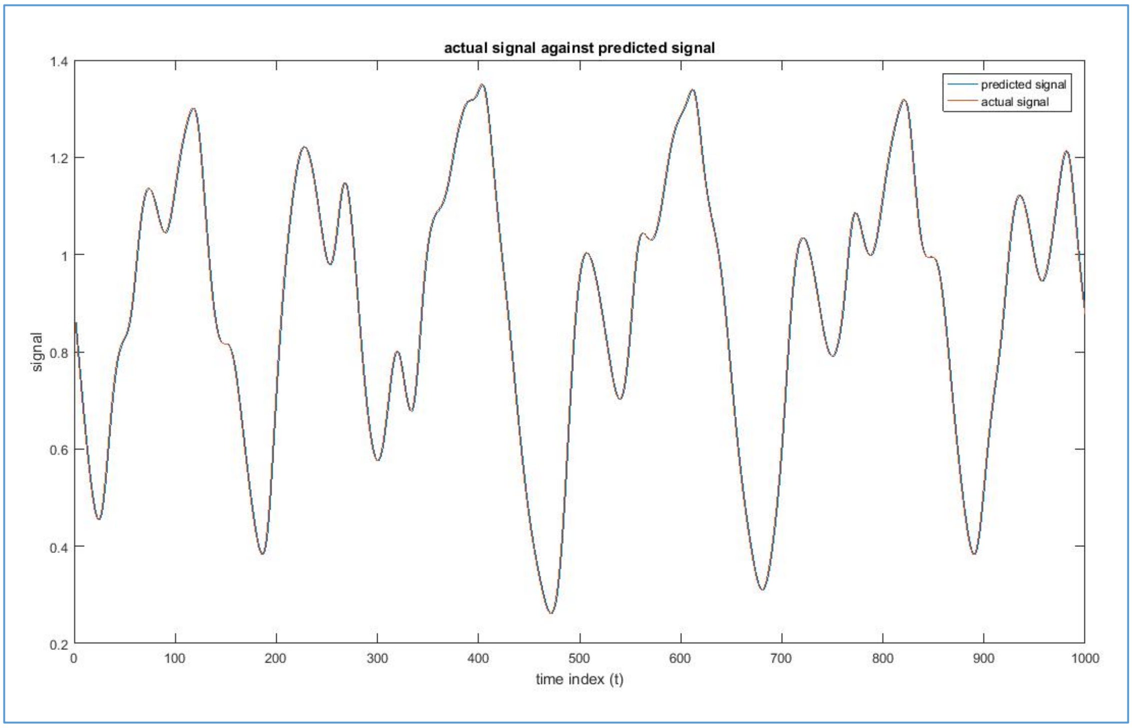

Figure 3. Actual test signal against predicted test signal

- Upon zooming in from t=140 to 220, we can see the difference in the two signals (figure 4).

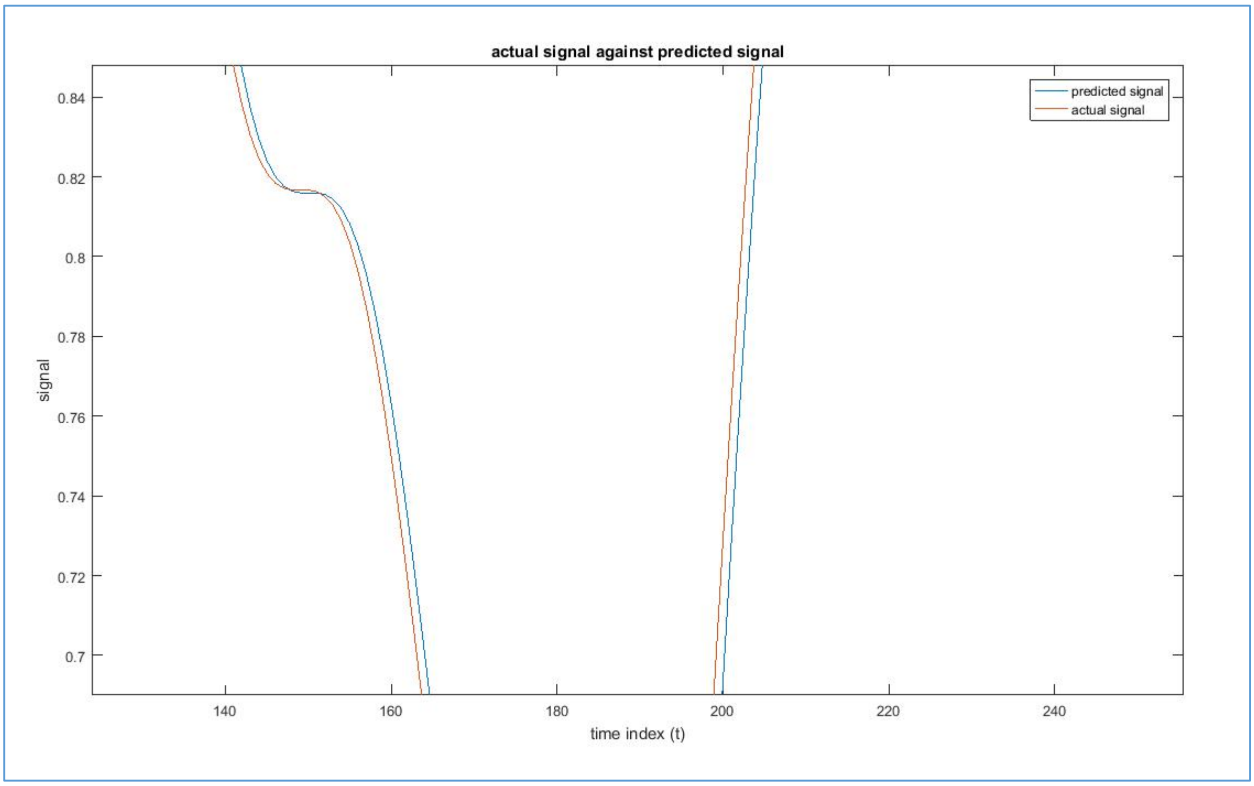

Figure 4. Zoomed in actual test signal against predicted signal

Kumar Rohit Malhotra (5697-1748)

Section 3: Repeat the problem using the normalized MSE algorithm.

- We repeat the problem using Normalized MSE, where we calculate the filter coefficients using the formula:

$$W(i+1) = W(i) + \alpha e(i) X(i)/X(i)^T X(i)$$

- The mean square error plot that we get in this case is given in figure 5.

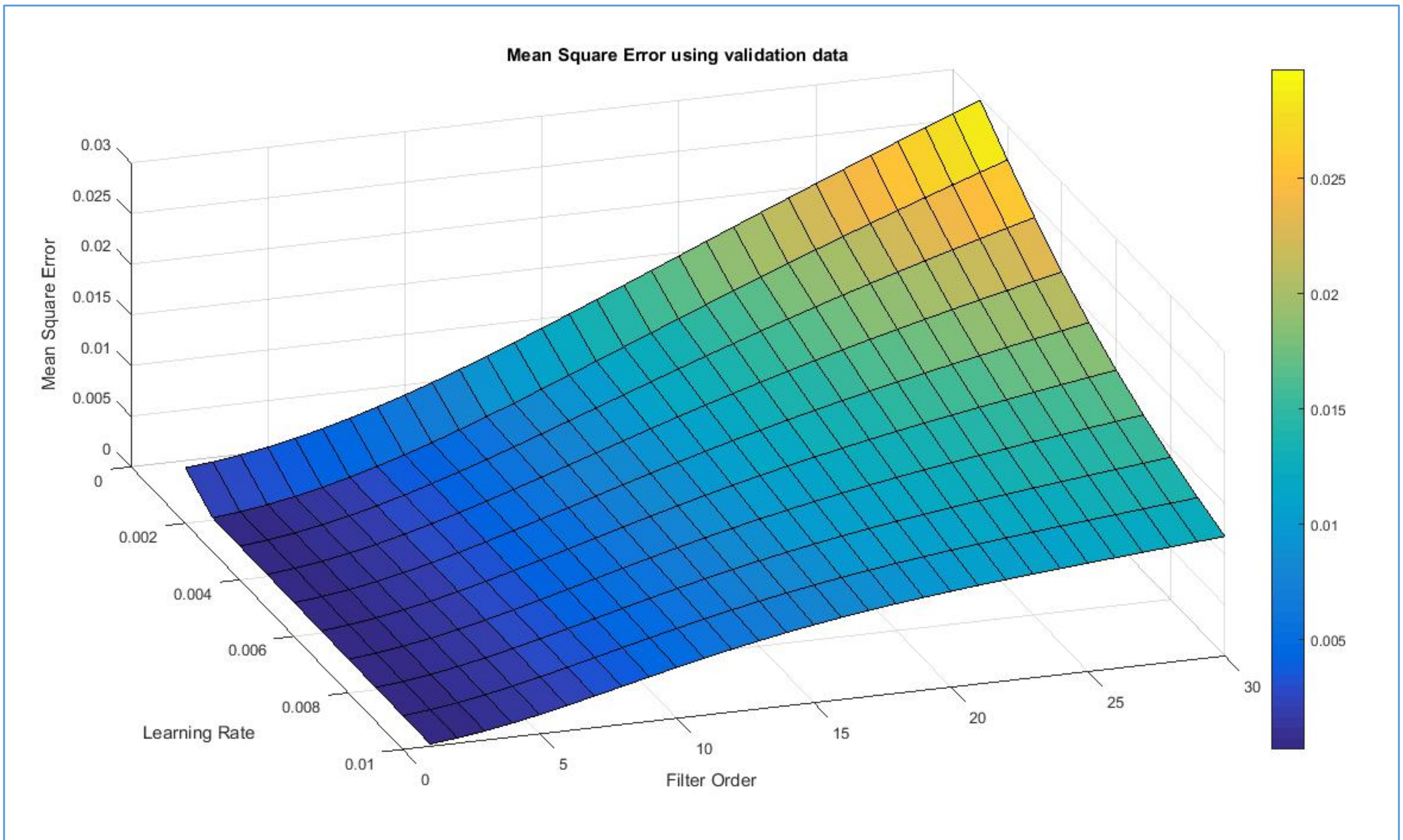

Figure 5. Mean Square Error against Learning rate and Filter order using Normalized MSE

- The minimum value of Mean Square Error is **0.00023107**, obtained for weights corresponding to Filter Order **M=1** and learning rate = **0.005.**
- In the same fashion as in LMS, we check the learning curve for 2 smaller learning rates, and see that the filter learns slower with smaller learning rates (figure 6). The MSE approximates to MSE_{min} at around 500^{th} sample.
- Once again, we predict the output from test data and plot the output signal against the desired signal (figure 7 and 8).

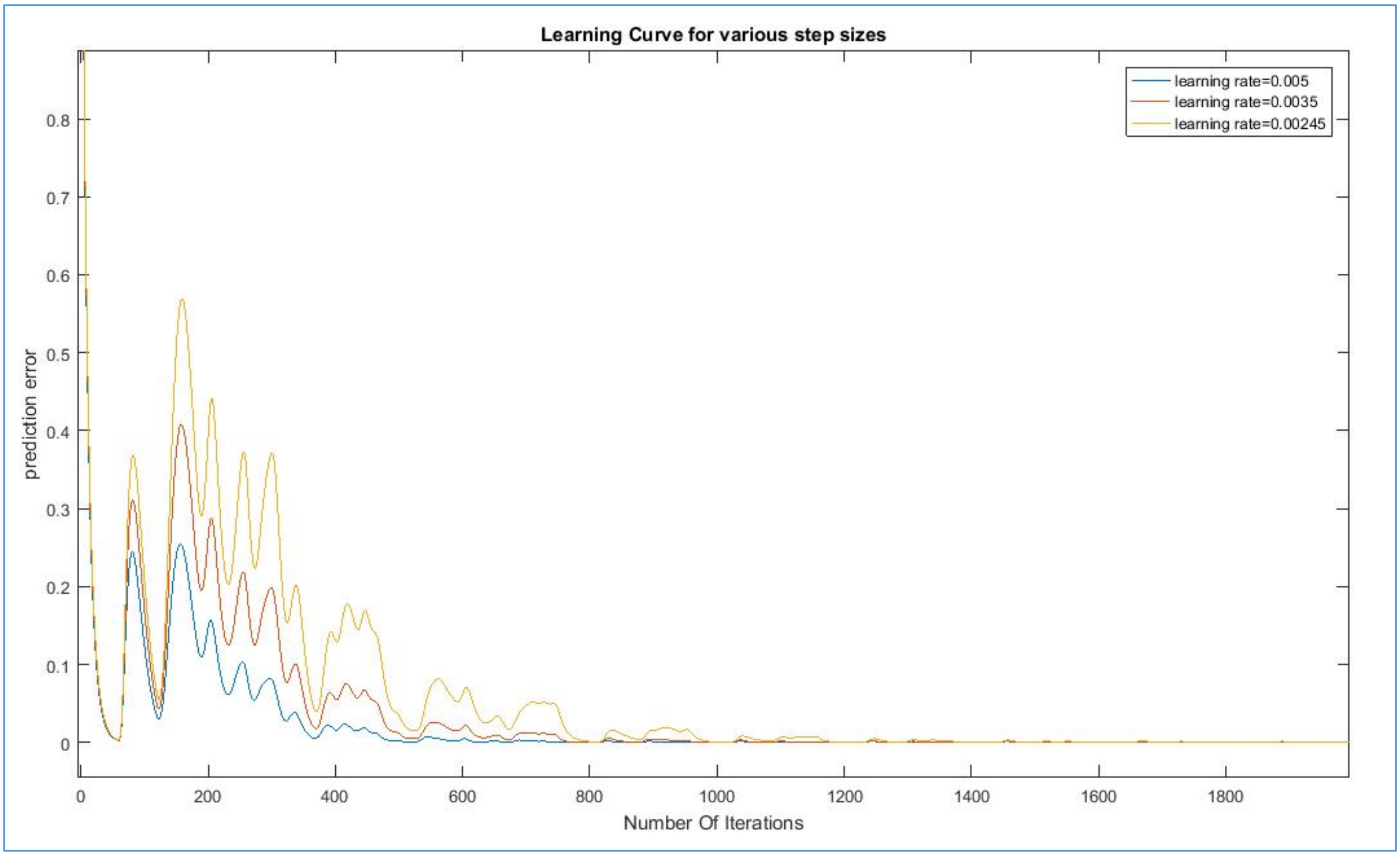

Figure 6. Learning curve for different learning rates and same filter order.

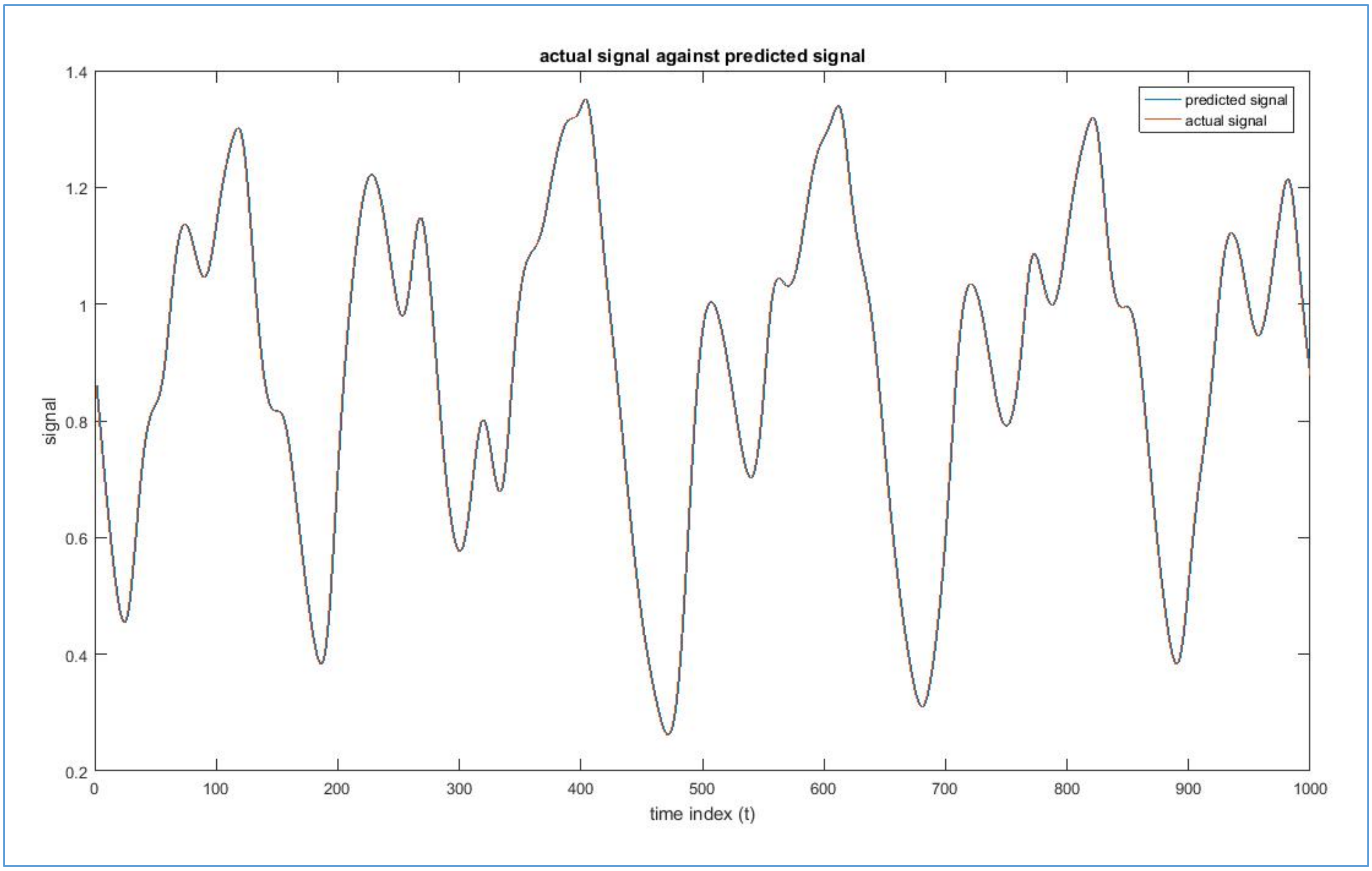

Figure 7. Actual test signal against predicted test signal

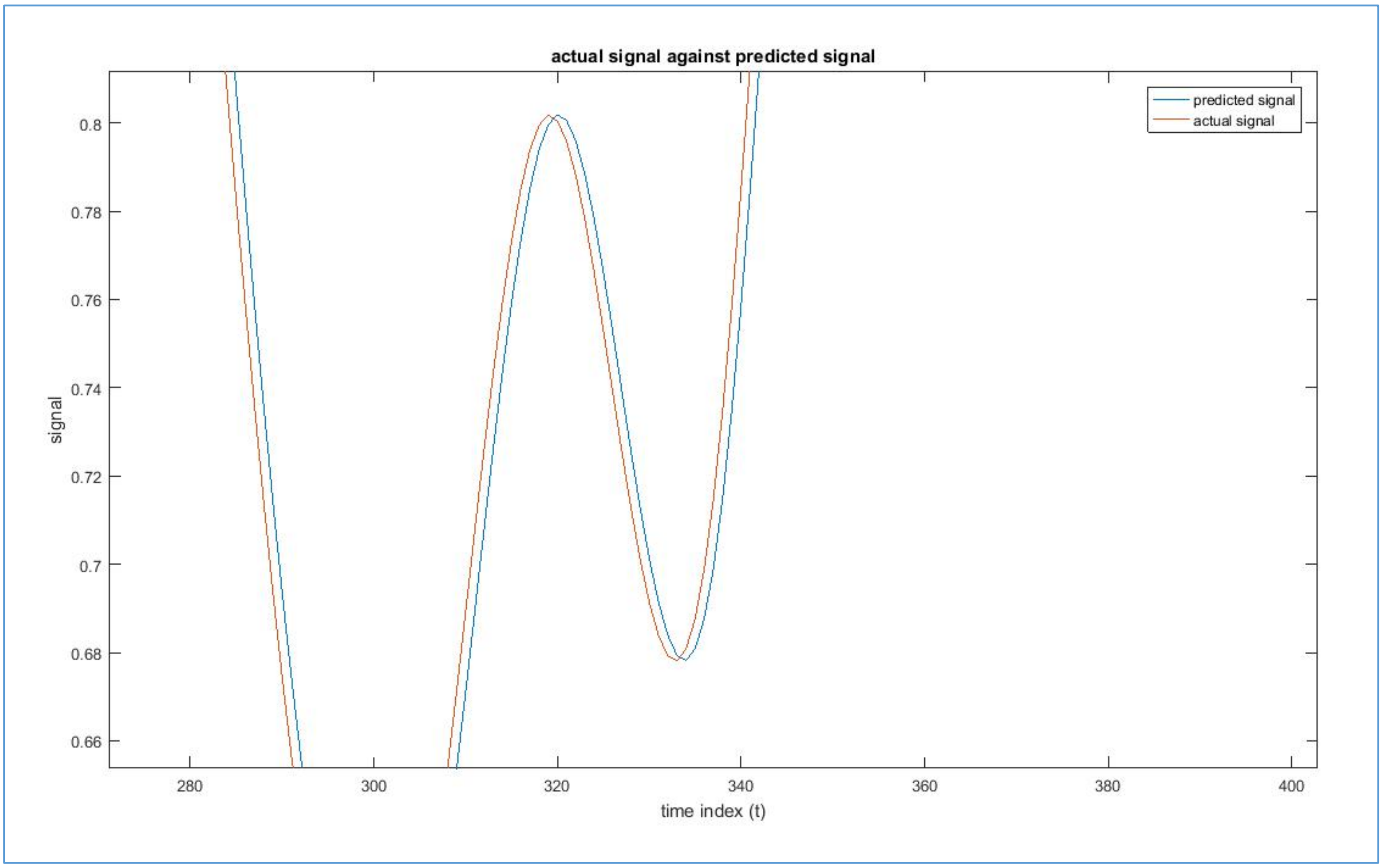

Figure 8. Zoomed in actual test signal against predicted signal

- Normalized MSE provides a better performance, in terms of convergence, in real world signals. LMS algorithm has a fixed learning rate. This makes it difficult to choose a suitable learning rate in real world signals. NLMS normalizes the learning rate with the power of the input, and hence we don't have to worry as much about selecting the learning rate. Because of this, the convergence rate is faster in case of Normalized MSE as compared to LMS, for the same learning rate α, as can be seen in figure 2 and figure 6.
- We should choose either of the two algorithms above on the basis of sample size and computational complexity trade off. If we have enough samples in training data to allow a slower convergence, we may use LMS algorithm. If we have fewer number of samples and we are not bothered much by the computational complexity of Normalized MSE, we should go ahead with Normalized MSE then.

Kumar Rohit Malhotra (5697-1748)

Section 4: Comparison of LMS and Least Squares Regression algorithm

- In homework 2, the best filter order obtained was 30 and regularization parameter 0. In LMS, the best filter order has been obtained as 1.
- The Least Squares regression algorithm provides a better performance as the Mean Square Error in that case is closer to the minimum Mean Square Error possible n and thus provides a more optimal set of filter coefficients.
- In terms of computation cost, we should choose LMS as Least Squares Regression algorithm has higher computational complexity as compared to LMS.