Warsaw University of Technology

Faculty of Electronics and Information Technology

Graduate Program (MSc)

Telecommunications

Adaptive signal processing

ESAP

Project title

Simulation Analysis and Comparison of Tracking Performance of Adaptive Algorithms

by

Kaleab Zemenay (336561)

Gashaw Yimer (336548)

Submitted to

Prof. Konrad

January 18, 2025

Warsaw, Poland

1. Introduction

Adaptive filters are widely employed in applications such as system identification, channel equalization, channel estimation, and echo cancellation. Their effectiveness is largely determined by the algorithm used to update filter weights, with the Recursive Least Squares (RLS) and Least Mean Squares (LMS) algorithms being the most prominent. Both algorithms offer unique advantages, but their performance in tracking time-varying systems remains a critical factor for comparison. Understanding their tracking behavior provides insight into their suitability for specific applications and highlights how one system adapts to changes faster or more accurately than another.

The adaptive filtering process involves not only determining the optimal filter weights but also ensuring the algorithm can track variations in these weights over time. Tracking behavior is a fundamental aspect of adaptive filters, as it dictates how well an algorithm can respond to changes in the environment or system dynamics. This project aims to analyze the tracking performance of the LMS and RLS algorithms, focusing on metrics such as convergence rate and mean square error. MATLAB simulations will be used to model these behaviors.

2. Performance Metrics

To evaluate the tracking performance of the selected algorithms, various criteria are employed, including the mean square error (MSE) of the error signal, the convergence rate, and the mean values and MSE of the filter coefficient estimates. The ideal algorithm is the one that demonstrates a faster convergence rate and lower MSE values. Each algorithm utilizes distinct parameters to achieve a low convergence rate and minimal MSE. To ensure stability, the critical parameters for the algorithms must be chosen within specified limits. We can classify this parameters into two categories that will help to assess the performance of respective algorithms.

2.1 Accuracy Metrics

a. Error Trajectory

It is the difference between the actual or desired value and the value estimated by the algorithm at each time step.

$$e(n) = d(n) - y(n)$$

$$\text{where } d(n) \text{ is the actual desired signal and}$$
$$y(n) \text{ is the estimated value of filtered signal}$$

b. Mean Squared Error

It is the average squared error over a sequence of time steps. It provides a summary measure of the magnitude of errors.

$$MSE = {}^{1}\!/_{N} \sum_{n=1}^{N} e^{2}(n)$$

$$\text{where } N \text{ is the total number of samples}$$

2.2 Convergence Metrics

a. Convergence time

The time it takes for the algorithm to minimize the error to a satisfactory level (steady state). The convergence rate can be determined from the graph depicting the MSE of the error signal as a function of the number of realizations. Each algorithm employs its own criteria to ensure convergence.

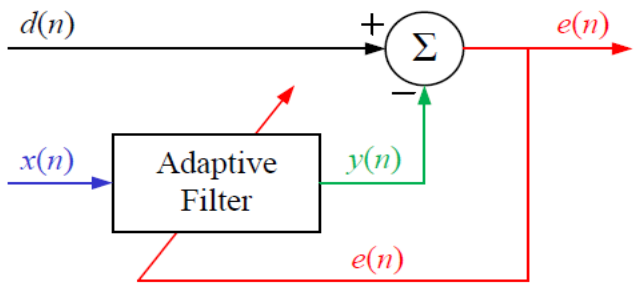
- For LMS algorithms the convergence is assured for the adaptation constant of α such that

$$0 < \alpha < \frac{2}{LE[x^{2}(n)]}$$

This stability condition (also approximate does not require the knowledge of the eigenvalues of the autocorrelation matrix ***R*** (or their calculation based on samples), but is dependent on the filter order.

3. Theoretical Background

3.1 LMS Algorithm



The system comprises an adaptive filter that processes the input signal $x(n)$ to generate the output signal $y(n)$. The filter is adjusted based on the error signal $e(n)$, which governs the filter coefficients ensuring that the output signal $y(n)$ closely matches the desired signal $d(n)$.

The LMS (Least Mean Square) algorithm is derived by minimization of the following error measure (cost function)

$$\hat{J}(n) = e^2(n) = [d(n) - \boldsymbol{f_n^T x_n}]^2$$

The algorithm is performed by using

$$y(n) = \boldsymbol{f}_n^T \boldsymbol{x}_n$$

$$e(n) = d(n) - y(n)$$

$$\boldsymbol{f}_{n+1} = \boldsymbol{f}_n + \alpha e(n) \boldsymbol{x}_n \text{ and } \boldsymbol{f}_0 = 0$$

In practice, the mean-squared error converges to a certain defined value denoted as J_∞. It results from the fact, that the values of the filter coefficients $\boldsymbol{f}$ are fluctuating, after achieving a steady-state (after adaptation), around the optimal solution $\boldsymbol{f}^*$

$$J_\infty = J_{min}(1 + \frac{\alpha}{2} tr(\boldsymbol{R}))$$

where $tr(R)$ is the trace of the autocorrelation matrix.

3.2 RLS Algorithm

Similar to the LMS algorithm the system consists of the adaptive filter acting on an input signal $x(n)$ to produce an output signal $y(n)$. The filter is updated using the error signal $e(n)$, which controls the filter coefficients f, so that the output signal $y(n)$ should approximate the desired signal $d(n)$.

The exponential Weighted Recursive Least-Squares EWRLS algorithm is derived by the minimization of the following error measure (cost function):

$$\tilde{J}_n = \sum_{l=0}^{n} \lambda^{n-1} e^2(l)$$

For $0 < \lambda \le 1$

Using the above cost function the RLS algorithm is described as follows

$$e(n|n-1) = d(n) - \boldsymbol{f}_{n-1}^{\boldsymbol{T}} \boldsymbol{x}_n$$

$$\alpha(n) = \frac{1}{\lambda + \boldsymbol{x}_n^T \boldsymbol{P}_{n-1} \boldsymbol{x}_n}$$

$$\boldsymbol{f}_n = \boldsymbol{f}_{n-1} + \alpha(n) e(n|n-1) \boldsymbol{P}_{n-1} \boldsymbol{x}_n$$

$$\boldsymbol{P}_n = \frac{1}{\lambda} [\boldsymbol{P}_{n-1} - \alpha(n) \boldsymbol{P}_{n-1} \boldsymbol{x}_n \boldsymbol{x}_n^T \boldsymbol{P}_{n-1}]$$

Where $e(n|n-1)$ is the a priori error, and $\boldsymbol{P}_n = \boldsymbol{R}_n^{-1}$ is the inverse of the correlation matrix estimate and the initialization is done using

$$\boldsymbol{f}_0 = 0 \text{ and } \boldsymbol{P}_0 = \gamma \boldsymbol{I}$$

4. Result

To track the performance of the adaptive algorithms we consider the input signal (x) as the white noise signal with standard deviation of 1 and the filter coefficients are considered as follows

$h1 = [1, 0.9, -0.7, 0.5, -0.3, 0.1]'$ and $h2 = h1 + 0.5$

The desired signal d(n) is considered as

$$d1 = step\_filter(h1, h2, step\_index, x) + sigma\_d * randn(1, N) \text{ and}$$

$$d2 = step\_filter2(h, step\_index, x) + sigma\_d * randn(1, N)$$

The first desired signal is used for the abrupt change at the value of step index where as the second one is used for the exponential change and linear change of parameter h and step index=250 is used. The parameter h is for d2 is updated using the Matlab script function

$$h = exponential\_change(x, a)$$

when the parameters change with the exponential growth a and

$$h = linear\_change(x, m)$$

when the parameters change linearly with slope m .Also

$$x = randn(1,500)$$

4.1 Abrupt Change

For sudden change of the parameter h at the value of the step index the error trajectory and the filter coefficient for the RLS and LMS algorithm is illustrated in the following plots using λ=1,0.95 and 0.9, $\alpha = 0.1 \; and \; 0.05$

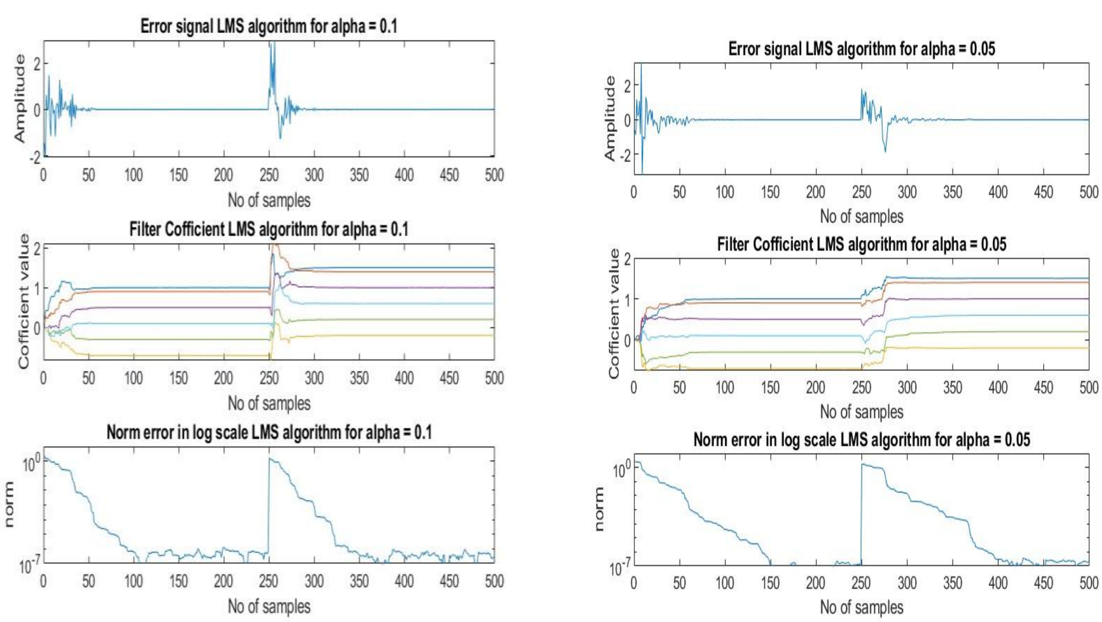


Figure 3. Error signal, filter coefficient and norm error of LMS algorithm for different α values

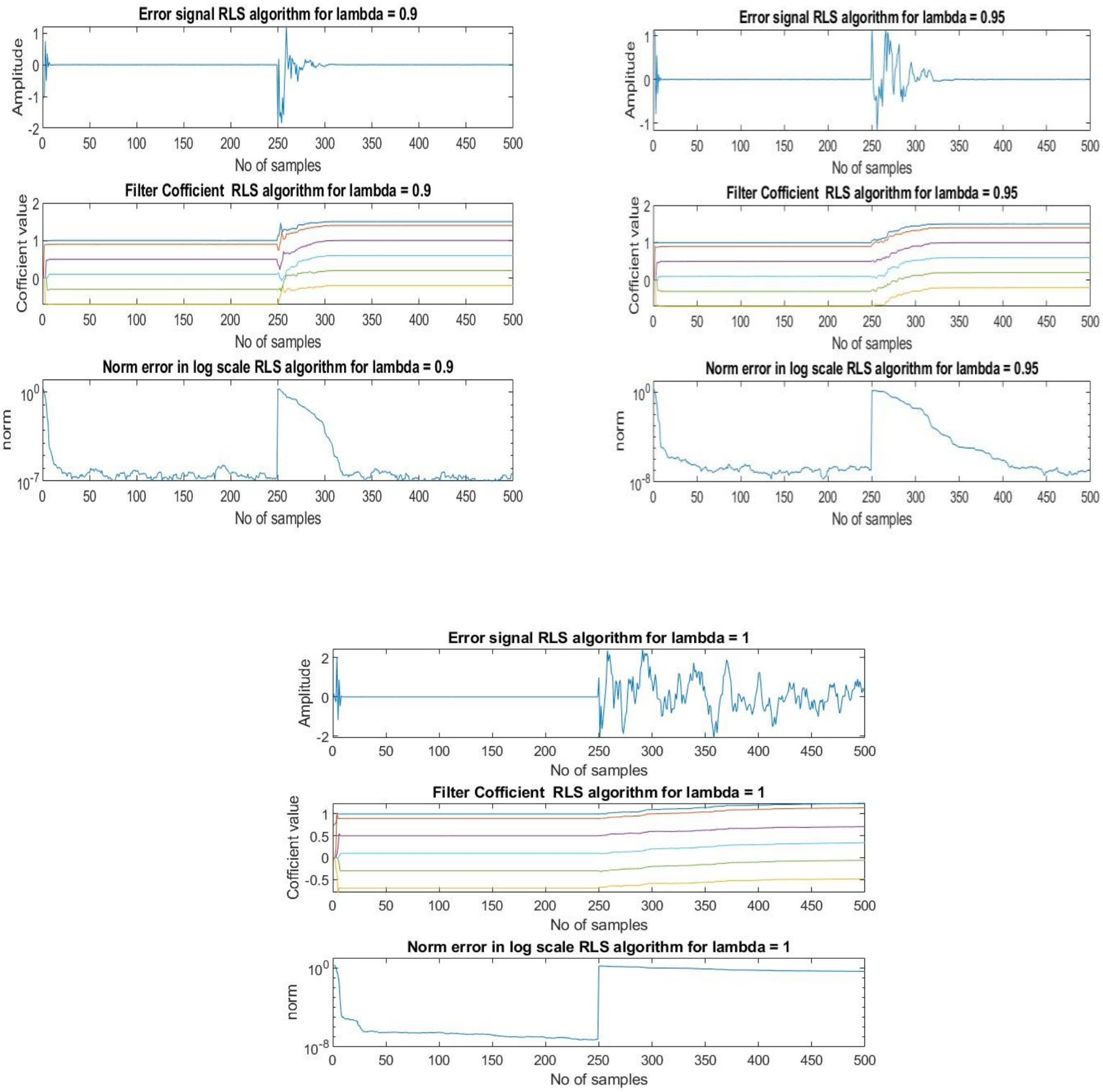


Figure 4. Error signal, filter coefficient and norm error of RLS algorithm for different λ values

From Figure 3 and 4 , we can see that for the LMS algorithm for lower value of α we can get low error in the last iterations but with slower adaptation after the step index. For RLS for lambda =0.9 we get faster adaptation but for 0.95 we get lower norm error after adaptation and finally for lambda=1 the system **can't totally track the abrupt** change in the filter coefficient

Next K=1000 different realizations have been used to investigate the behavior of each algorithm's MSE error, MSE of the filter coefficient , Mean of filter coefficients and Mean of the norm error. Note: All the MSE plots are expressed in the logarithmic scale. Again using λ=1,0.95 and 0.9, $\alpha = 0.1\ and\ 0.05$

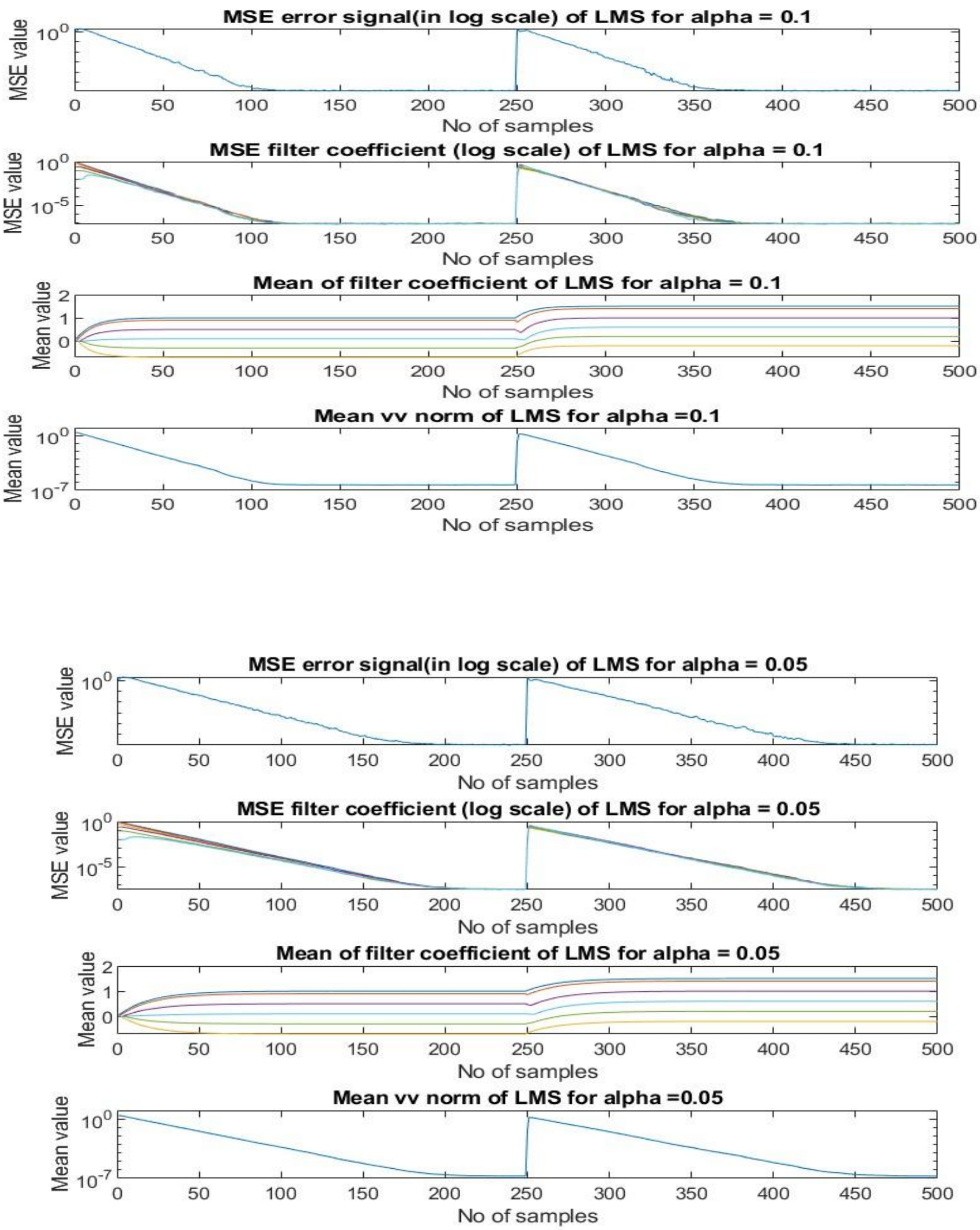
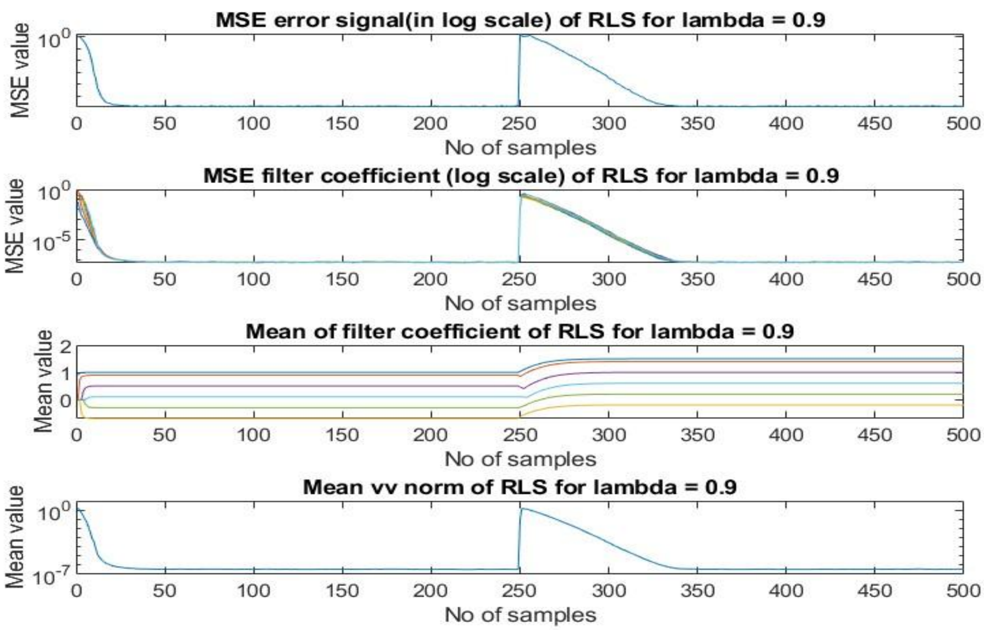
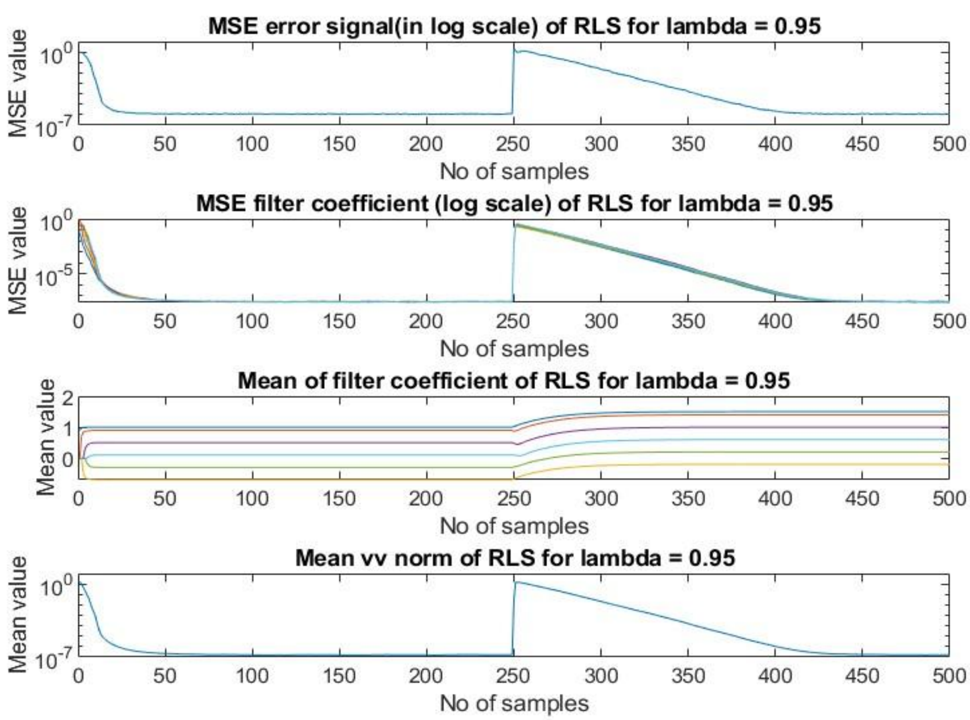


Figure 5. MSE e, MSE f, Mean of f and Mean of the norm of LMS for different α values


MSE error signal(in log scale) of RLS for lambda = 0.9
MSE value
No of samples
MSE filter coefficient (log scale) of RLS for lambda = 0.9
MSE value
No of samples
Mean of filter coefficient of RLS for lambda = 0.9
Mean value
No of samples
Mean vv norm of RLS for lambda = 0.9
Mean value
No of samples


MSE error signal(in log scale) of RLS for lambda = 0.95
MSE value
No of samples
MSE filter coefficient (log scale) of RLS for lambda = 0.95
MSE value
No of samples
Mean of filter coefficient of RLS for lambda = 0.95
Mean value
No of samples
Mean vv norm of RLS for lambda = 0.95
Mean value
No of samples

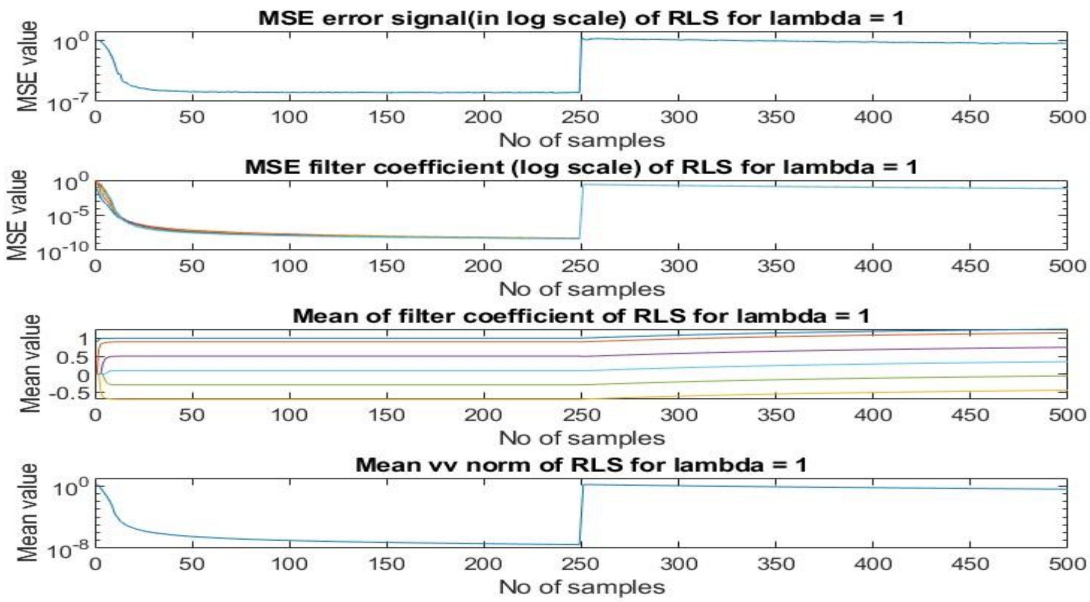


Figure 6. MSE e, MSE f, Mean of f and Mean of the norm of RLS algorithm for different λ

Table 1 Steady state result comparison (last 50 samples)

	MSE error	Mean value of norm
LSM ($\alpha = 0.05$)	1.1813×10^{-6}	1.9641×10^{-7}
LSM ($\alpha = 0.1$)	1.5195×10^{-6}	5.1460×10^{-7}
RLS ($\lambda = 0.9$)	1.3436×10^{-6}	3.5430×10^{-7}
RLS ($\boldsymbol{\lambda = 0.95}$)	$\mathbf{1.1728 \times 10^{-6}}$	$\mathbf{1.6421 \times 10^{-7}}$
RLS ($\lambda = 1$)	0.4274	0.4652

From the graphs and the above steady state (after adaptation) results we observed for LMS algorithm for both α values the system will be stable but for small alpha value we get lower MSE error values with slower adaptation rate. For the RLS algorithm $\lambda = 0.95$ value yields but MSE error values but with slower adaptation time and for $\lambda = 1$ the system totally **cannot track the change.**

In general the system RLS with lambda 0.95 has a better error performance while $\lambda = 1$ has faster tracking rate (adaptation rate).

4.2 Exponential change

Instead of abrupt change of system parameter h now a gradual change is applied and this is modeled using the exponential function

$$h = h1 + (h2 - h1)(1 - e^{-a(n-step\_index)}) \text{ for } n > step\_index$$

The value of a determines the rate of growth of the parameter. For instance, the first coefficient of h for different a values is can be seen as in the following figure

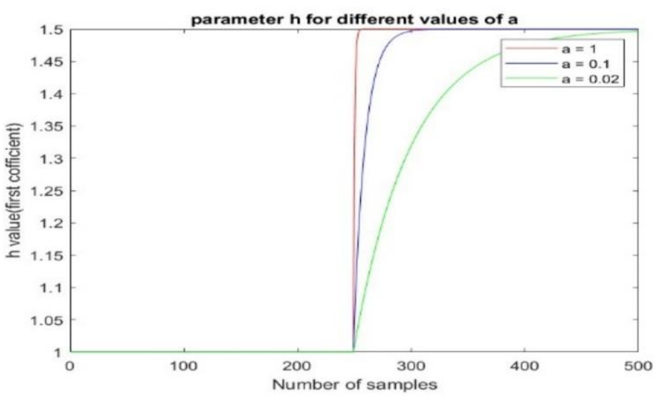


Figure 7. Parameter change for different growth rate values

Next taking these values (specially 0.1 and 0.02) the error trajectory and filter coefficient behavior of both algorithms is investigated

a. For $\boldsymbol{a = 0.02}$

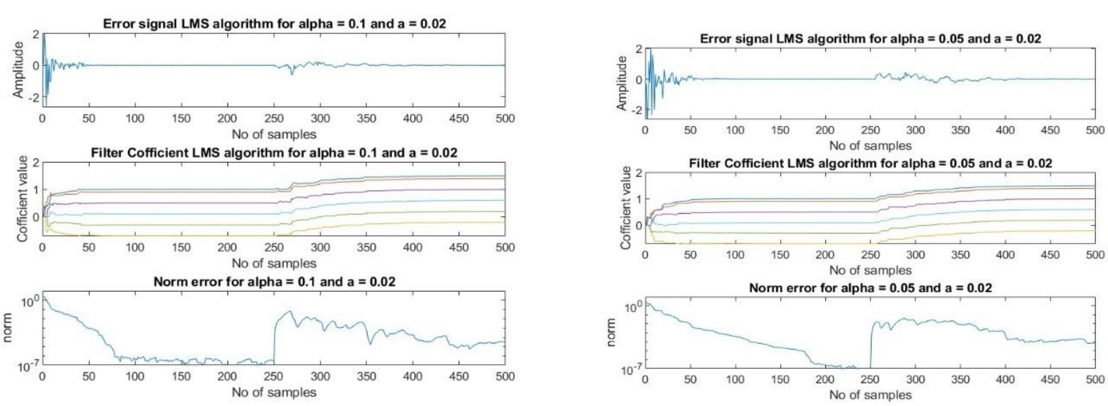


Figure 8. Error signal , filter coefficient and norm error of LMS algorithm at a =0.02 for different alphas

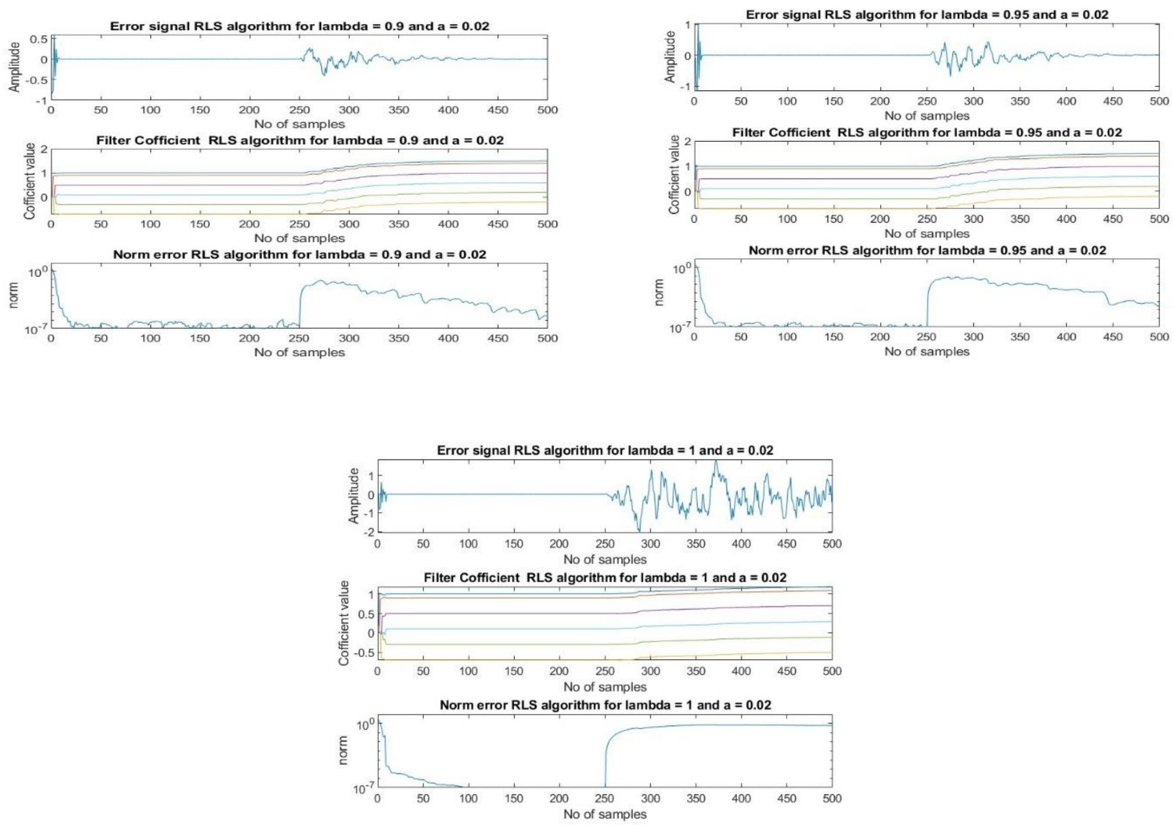
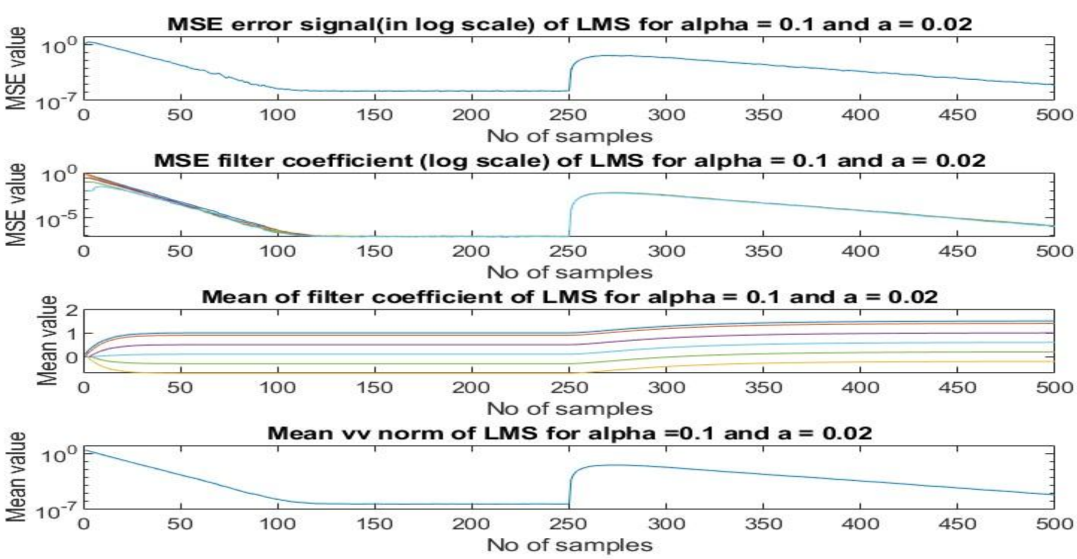


Figure 9. Error signal ,filter coefficient and error for RLS algorithm at a =0.1 for different lambdas

Next K=1000 is used, to get the full picture of the performance of the algorithms



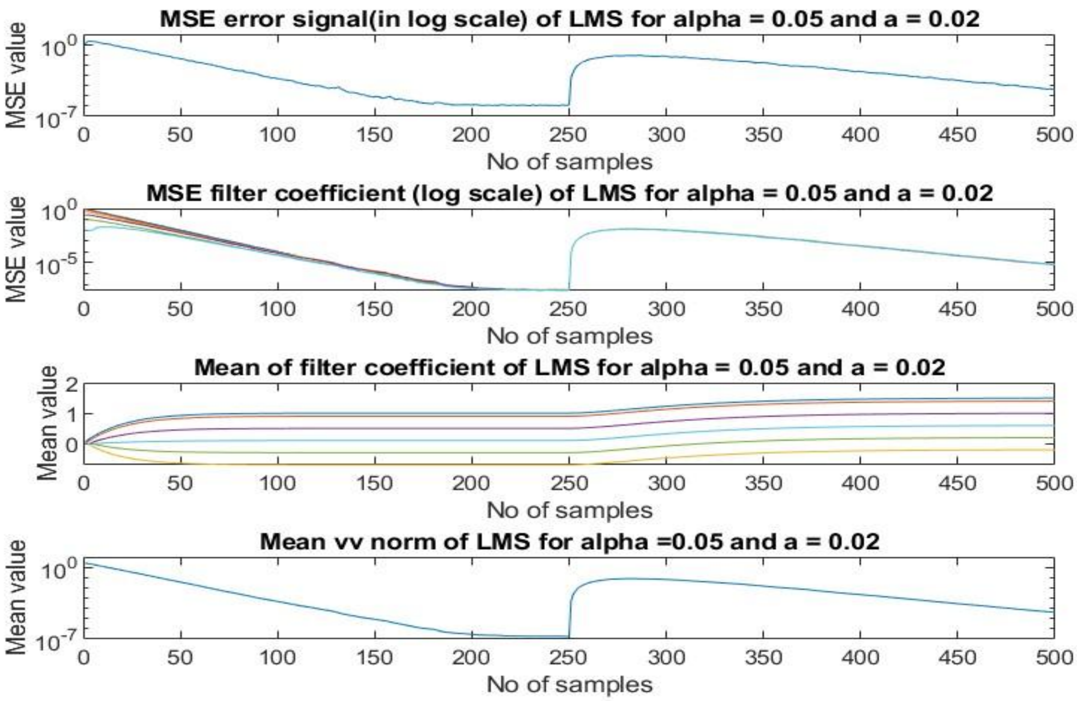
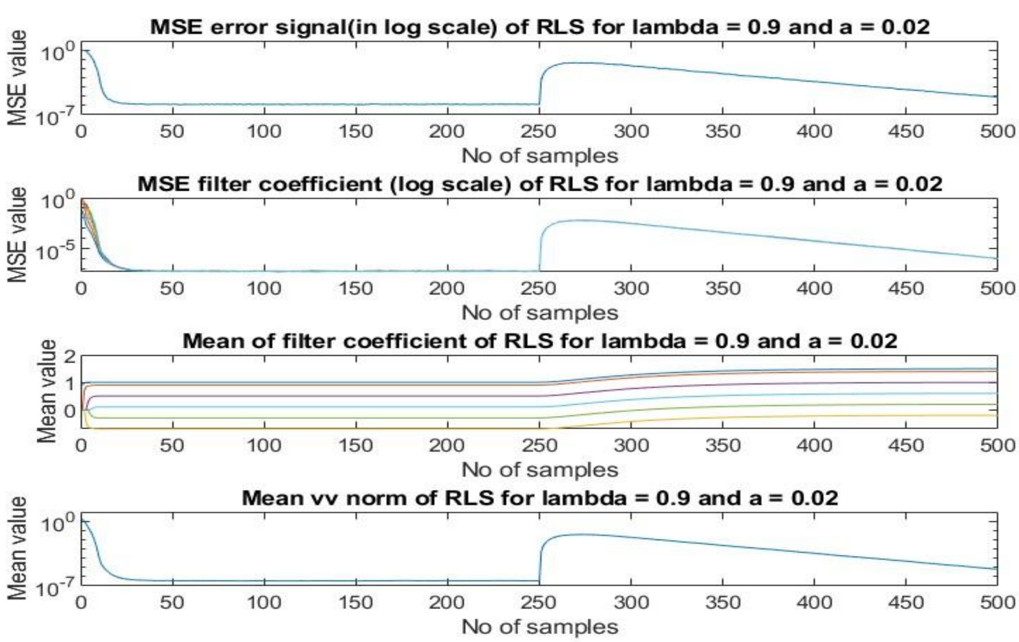


Figure 10. MSE e, MSE f, Mean of f and Mean of the norm for LMS for a=0.02 for different alphas



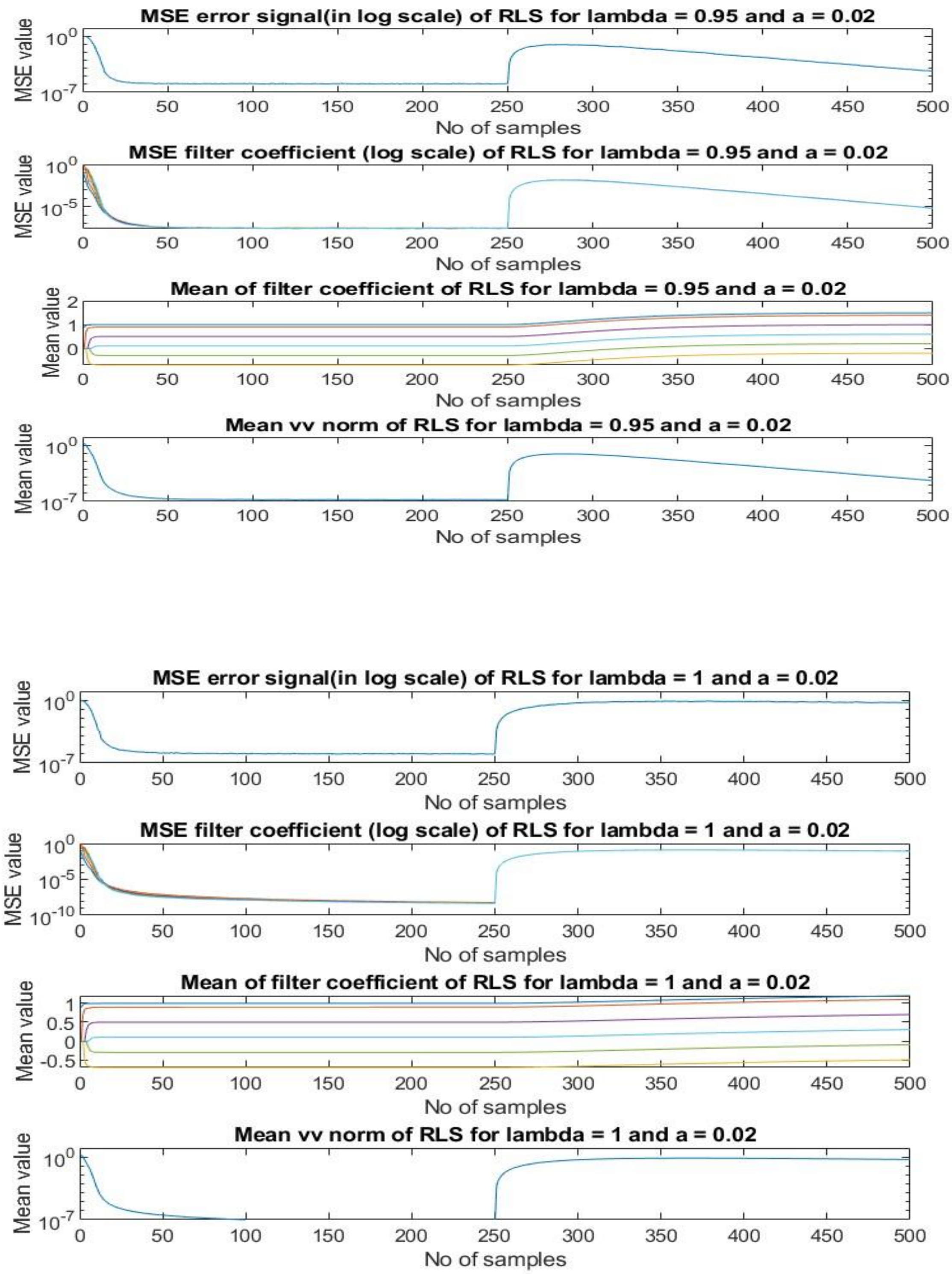


Figure 11. MSE e, MSE f, Mean of f and Mean of the norm for RLS for a=0.02 for different lambdas

Table 2 Last 50 iteration result comparison for $a = 0.02$

	MSE error	Mean value of norm
LSM ($\alpha = 0.05$)	1.3739×10^{-4}	1.3054×10^{-4}
LSM ($\alpha = 0.1$)	2.4974×10^{-5}	2.0146×10^{-5}
RLS ($\lambda = 0.9$)	2.3968×10^{-5}	1.9922×10^{-5}
RLS ($\boldsymbol{\lambda = 0.95}$)	$\mathbf{1.2814 \times 10^{-4}}$	$\mathbf{1.2059 \times 10^{-4}}$
RLS ($\lambda = 1$)	0.5959	0.5759

From the previous figures and the above table we can see that for lower alpha value the LMS algorithm needs longer time to lower the error values. Similar to the abrupt change for lambda =1 the system **isn't stable** at all and for lambda=0.95 we obtained smallest error values in the last 50 iterations.

b. For $\boldsymbol{a = 0.1}$

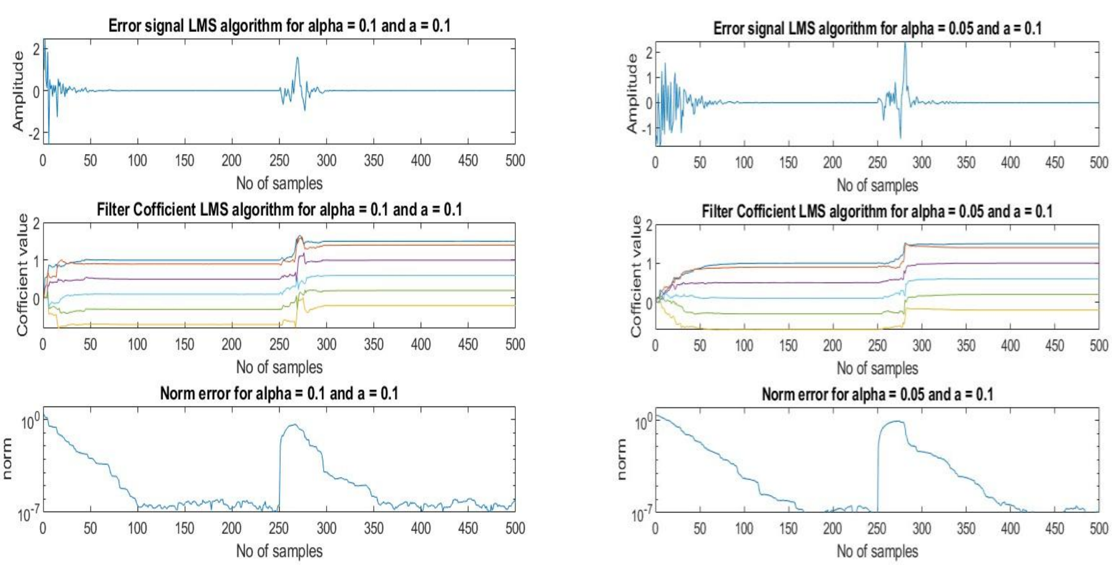


Figure 12. Error signal , filter coefficient and norm error of LMS algorithm at a =0.1 for different alphas

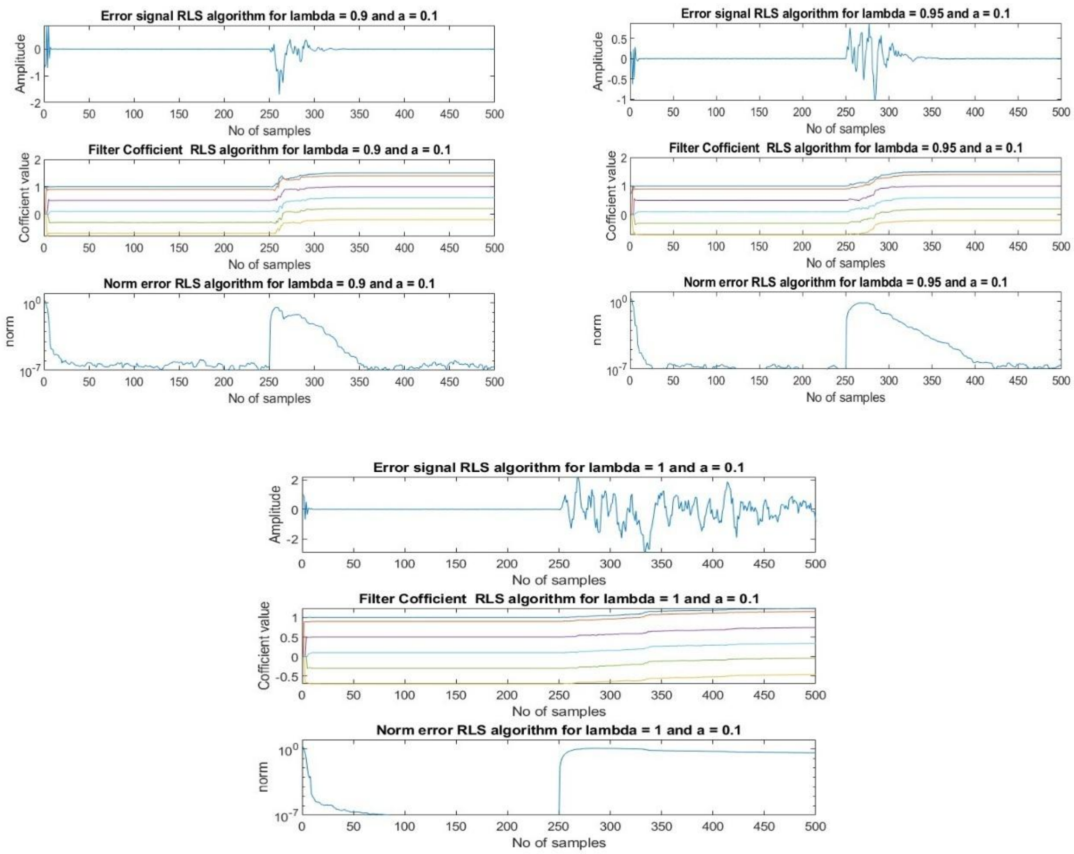
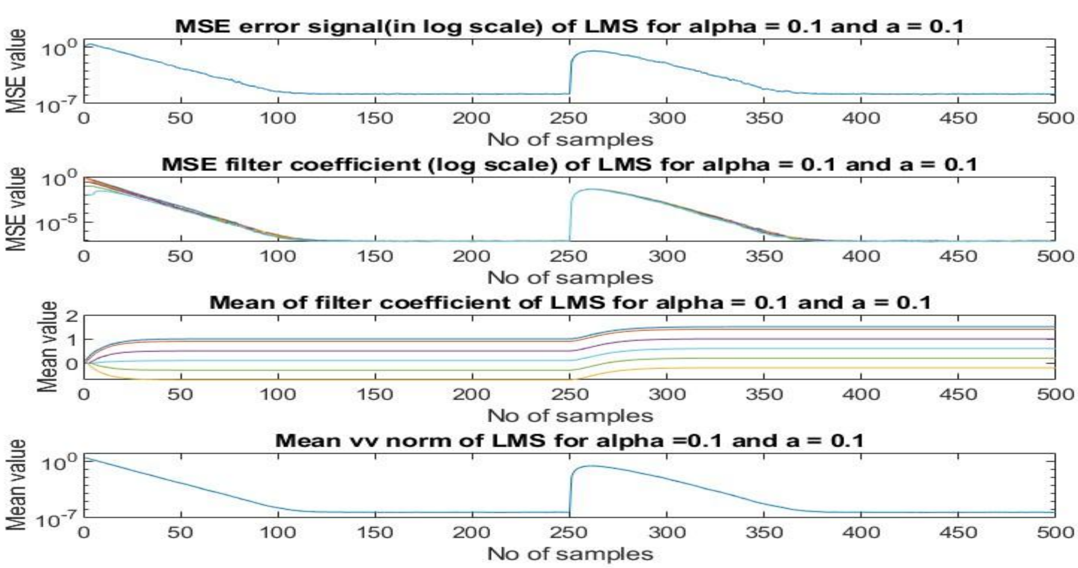


Figure 13. Error signal , filter coefficient and norm error of RLS algorithm at a =0.1 for different lambdas



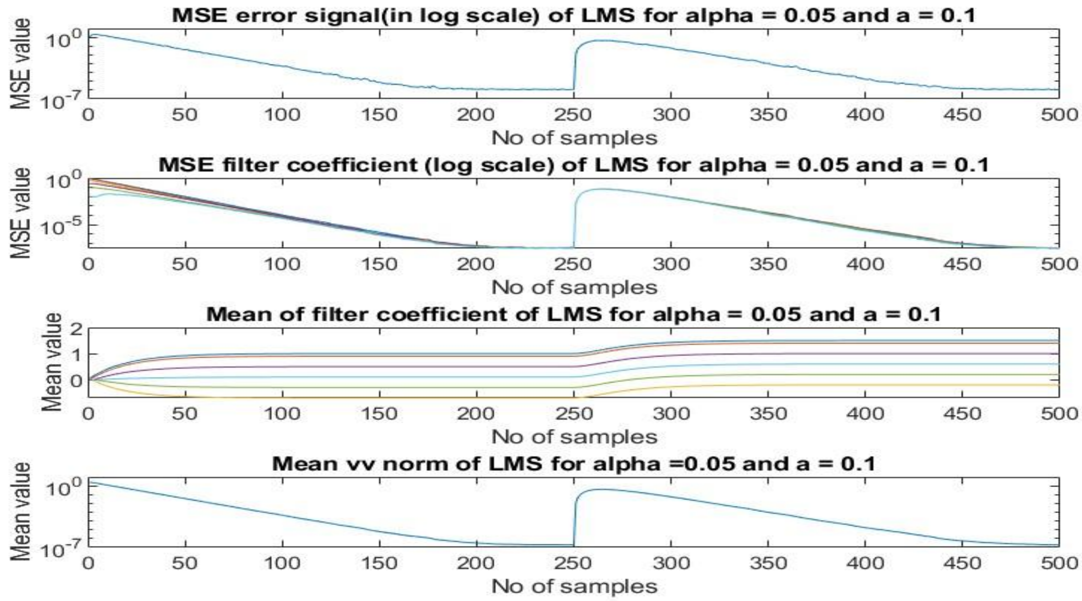
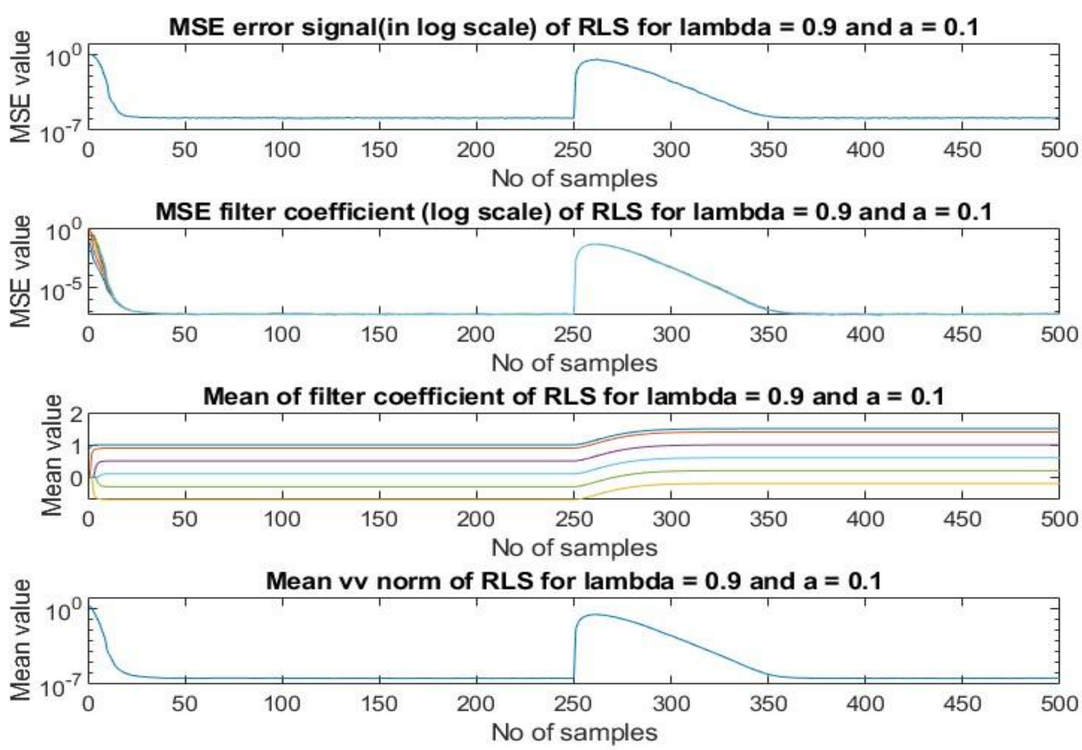


Figure 14. MSE e, MSE f, Mean of f and Mean of the norm for LMS for a=0.1 for different alphas



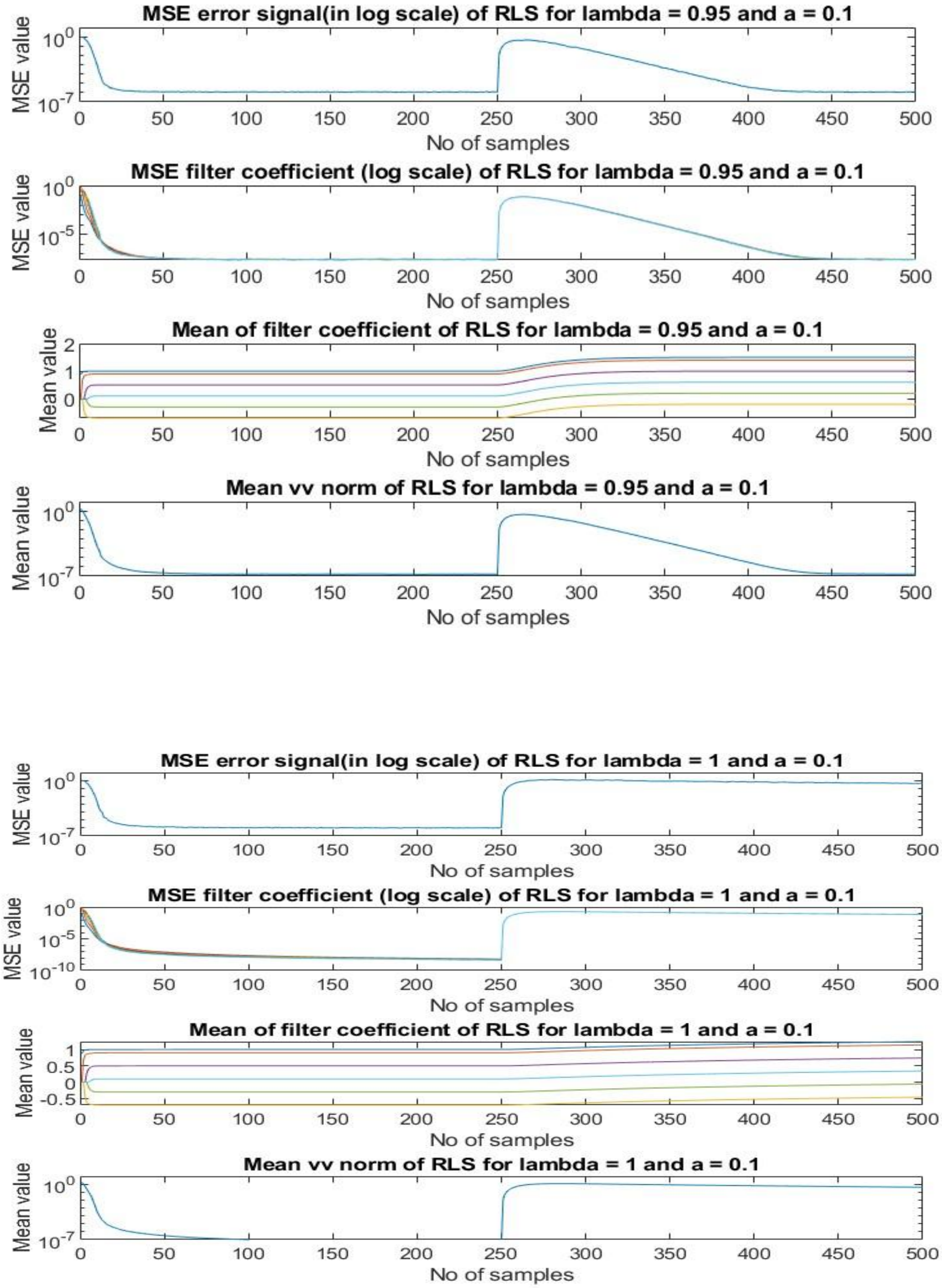


Figure 15. MSE e, MSE f, Mean of f and Mean of the norm for RLS for a=0.1 for different lambdas

Table 3 Last 50 iteration result comparison for $a = 0.1$

	MSE error	Mean value of norm
LSM ($\alpha = 0.05$)	1.2396×10^{-6}	2.2527×10^{-7}
LSM ($\alpha = 0.1$)	1.5278×10^{-6}	4.9522×10^{-7}
RLS ($\lambda = 0.9$)	1.3446×10^{-6}	3.5744×10^{-7}
RLS ($\boldsymbol{\lambda = 0.95}$)	$\mathbf{1.1712 \times 10^{-6}}$	$\mathbf{1.6722 \times 10^{-7}}$
RLS ($\lambda = 1$)	0.4587	0.4539

From the table and the previous figures it can be observed that for a=0.1, smaller values of alpha yields better error value but with longer time to converge. For RLS algorithm lambda=0.95 gives much lower error values but again with longer time to converge while lambda=0.9 has a better behavior in **converging quickly.**

In general, for exponential change slow growth gives both the algorithm to **converge slowly** while the larger growth value a yields gives a better time for converging

4.3 Linear change

Now let’s assume that h grows linearly with some different values of constant slope m. Thus, it is modeled using the linear function

$$h = h1 + m * (n - step\_index)\ for\ n > step\_index \text{ and } h < h2$$

$$h = h2 \quad for > step\_index \text{ and } h > h2\ , \quad \text{where the slope is } m = \tan(\alpha)$$

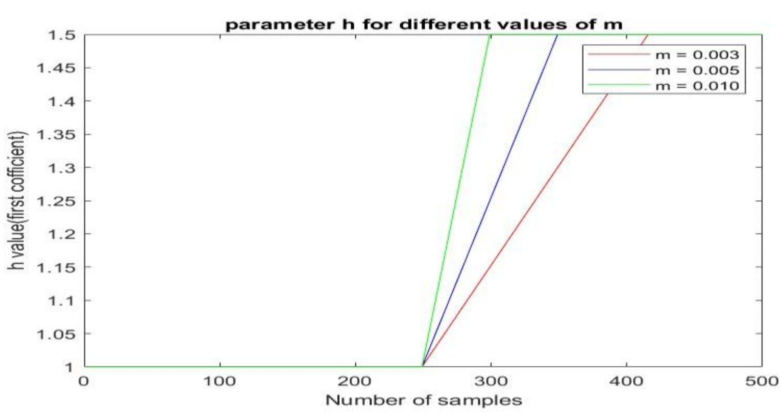


Figure 16. Parameter change for different slope values

Next taking these values (specially $m = 0.003\ and\ m = 0.01$) the error trajectory, filter coefficient and norm error behavior of both algorithms is investigated

a. For $m = 0.003$

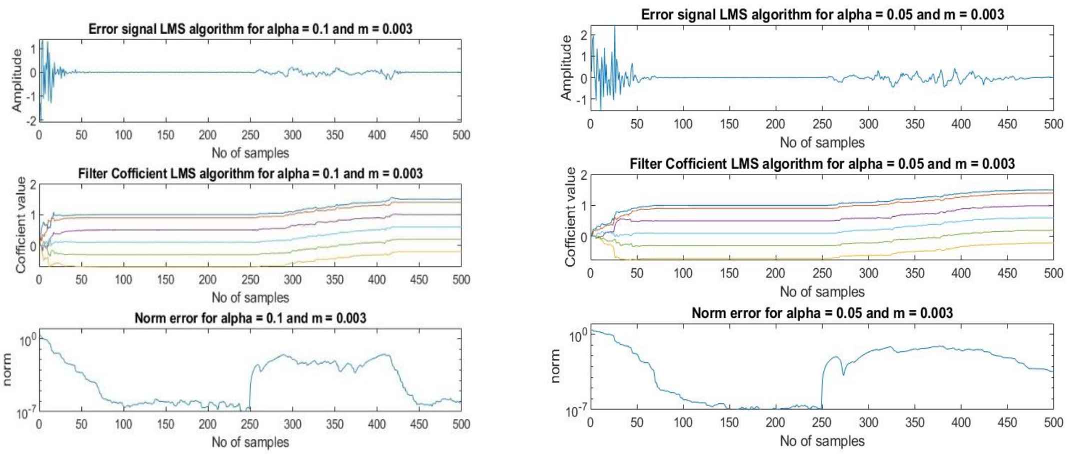


Figure 17. Error signal, filter coefficient and norm error of LMS algorithm at m =0.003 for different alphas

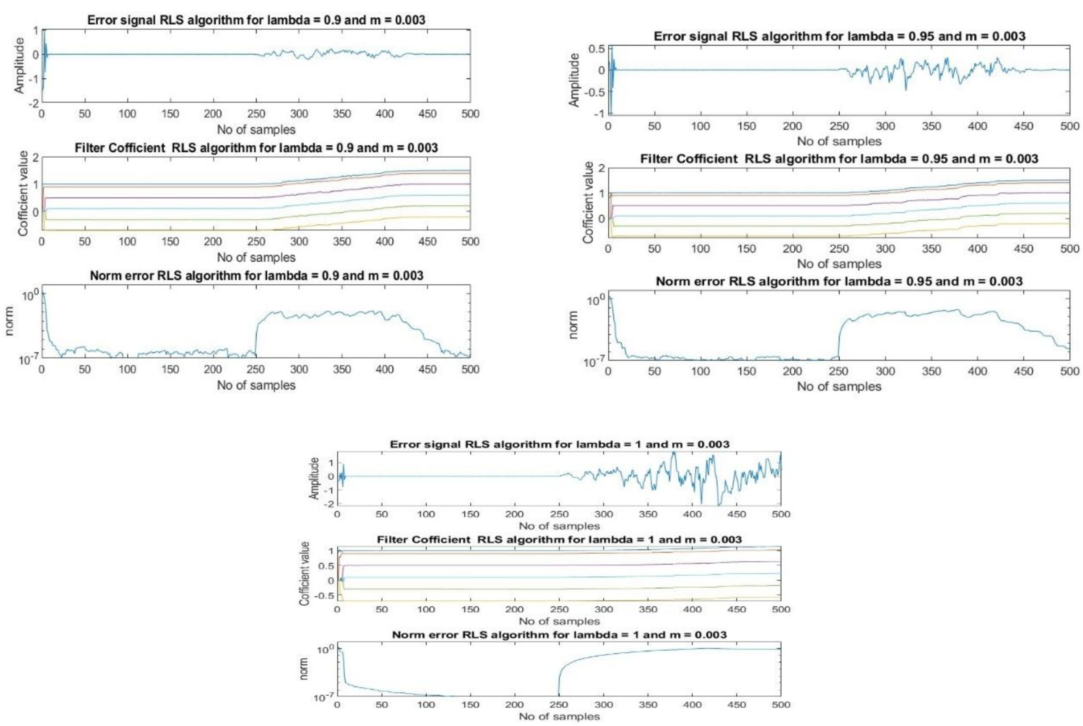


Figure 18. Error signal , filter coefficient and norm error of RLS algorithm at m =0.003 for different lambdas

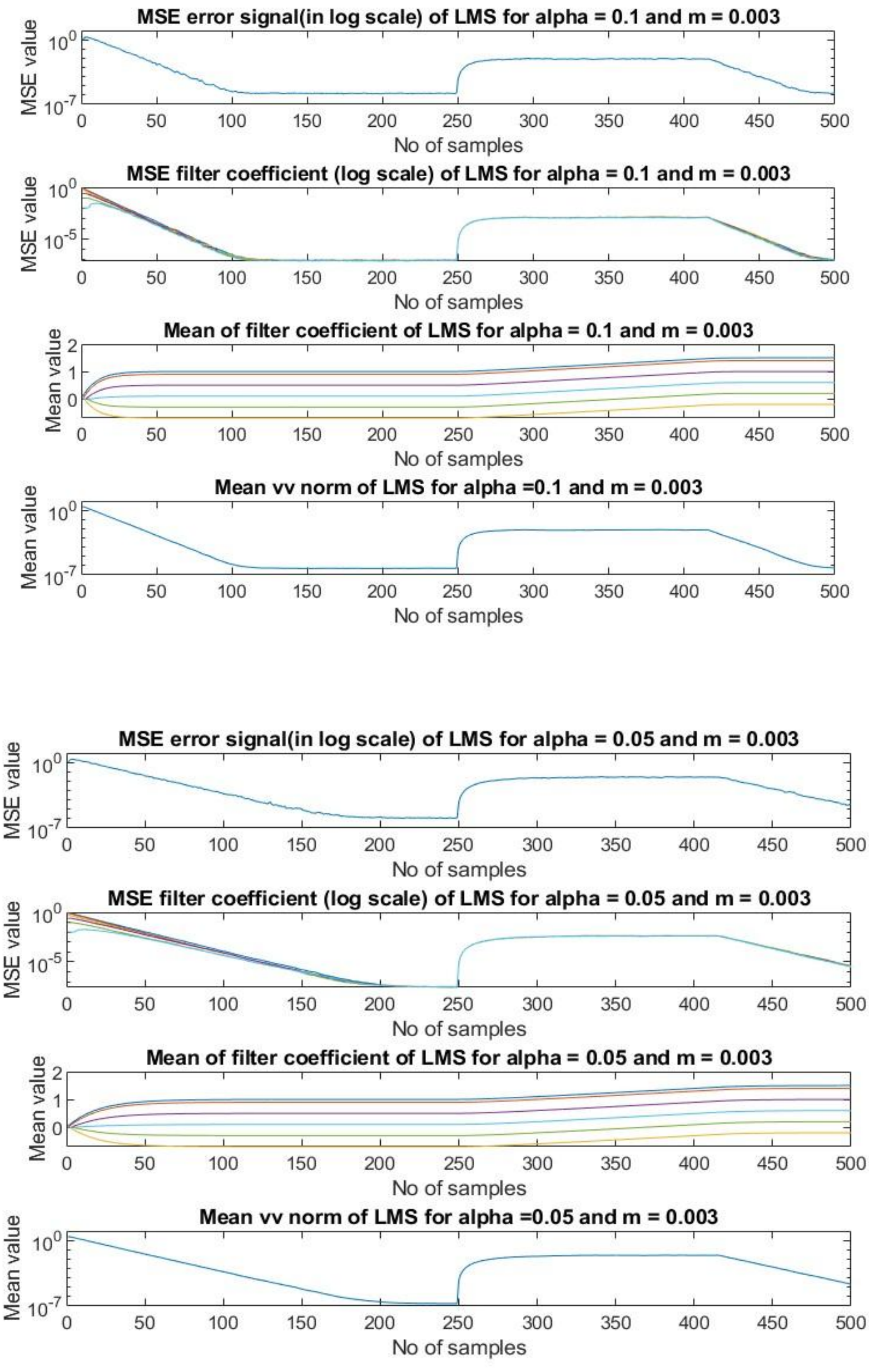
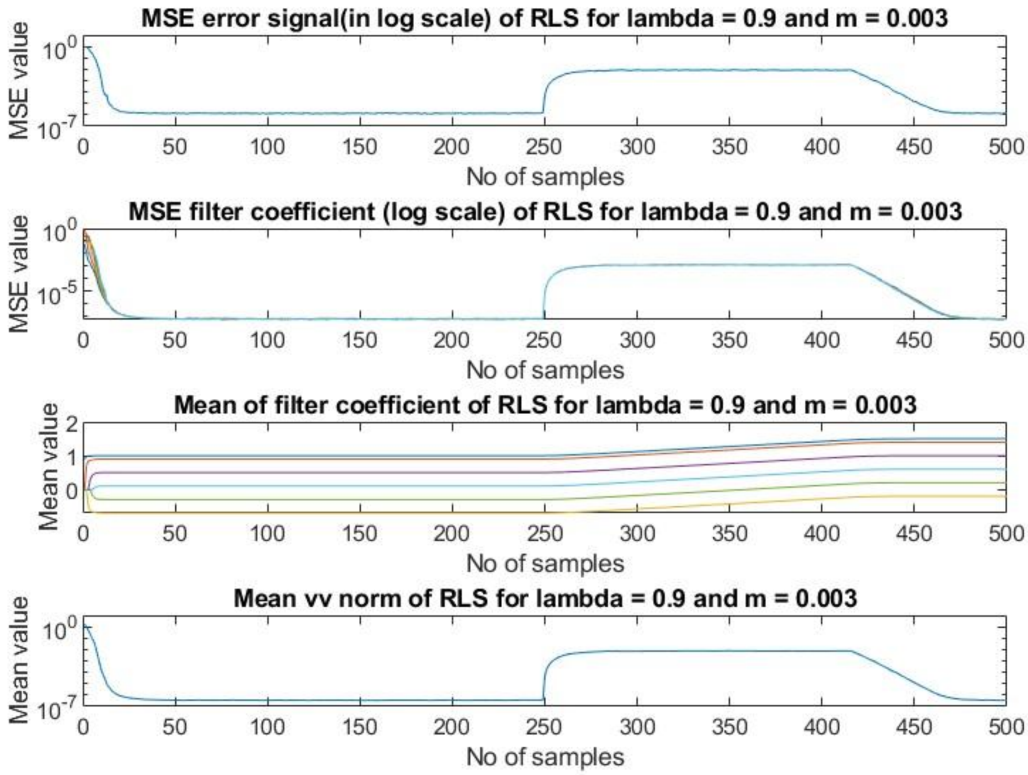
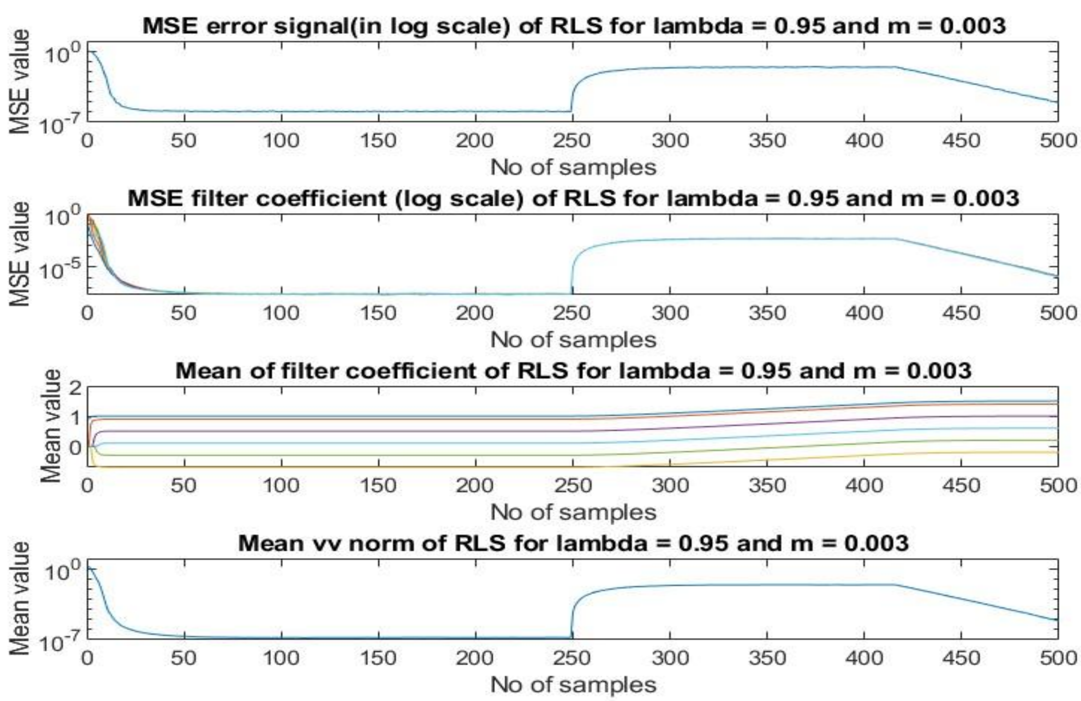


Figure 19. MSE e, MSE f, Mean of f and Mean of the norm for LMS for m=0.003 for different alphas


MSE error signal(in log scale) of RLS for lambda = 0.9 and m = 0.003
MSE value
No of samples
MSE filter coefficient (log scale) of RLS for lambda = 0.9 and m = 0.003
MSE value
No of samples
Mean of filter coefficient of RLS for lambda = 0.9 and m = 0.003
Mean value
No of samples
Mean vv norm of RLS for lambda = 0.9 and m = 0.003
Mean value
No of samples


MSE error signal(in log scale) of RLS for lambda = 0.95 and m = 0.003
MSE value
No of samples
MSE filter coefficient (log scale) of RLS for lambda = 0.95 and m = 0.003
MSE value
No of samples
Mean of filter coefficient of RLS for lambda = 0.95 and m = 0.003
Mean value
No of samples
Mean vv norm of RLS for lambda = 0.95 and m = 0.003
Mean value
No of samples

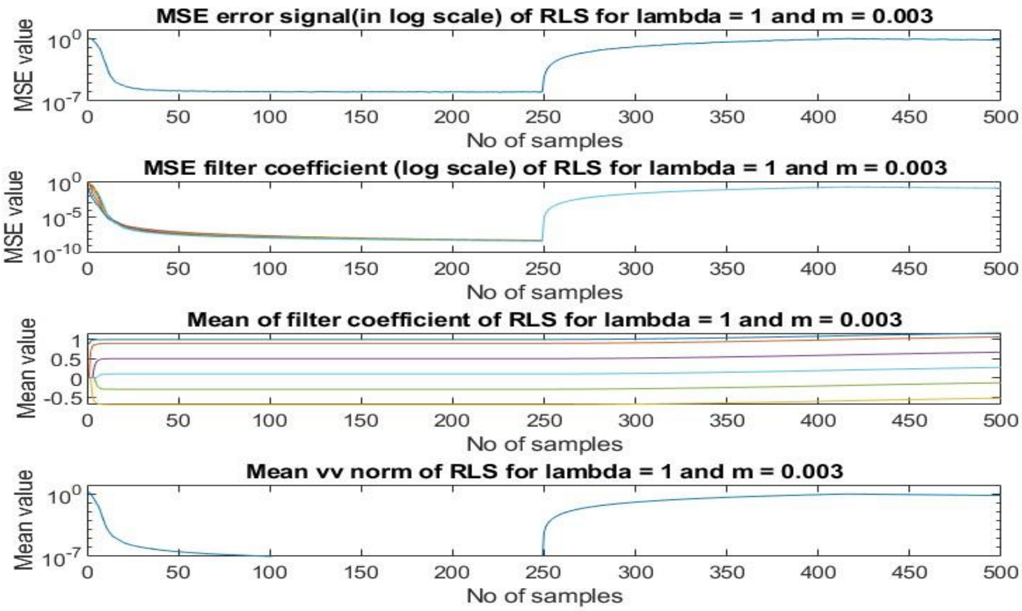


Figure 20. MSE e, MSE f, Mean of f and Mean of the norm for RLS for m=0.003 for different lambdas

Table 4 Last 50 iteration result comparison for $m = 0.003$

	MSE error	Mean value of norm
LSM ($\alpha = 0.05$)	4.0582×10^{-4}	3.9533×10^{-4}
LSM ($\alpha = 0.1$)	1.3404×10^{-5}	1.0955×10^{-5}
RLS ($\boldsymbol{\lambda = 0.9}$)	$\mathbf{2.8477 \times 10^{-6}}$	$\mathbf{1.6265 \times 10^{-6}}$
RLS ($\lambda = 0.95$)	2.2441×10^{-4}	2.1125×10^{-4}
RLS ($\lambda = 1$)	0.7503	0.7377

From the above figures we can observe that for small slope linear change RLS with lambda 0.9 has both the small MSE errors and faster convergence rates, while for the LMS alpha value 0.05 much longer time to converge

b. For $\boldsymbol{m = 0.01}$

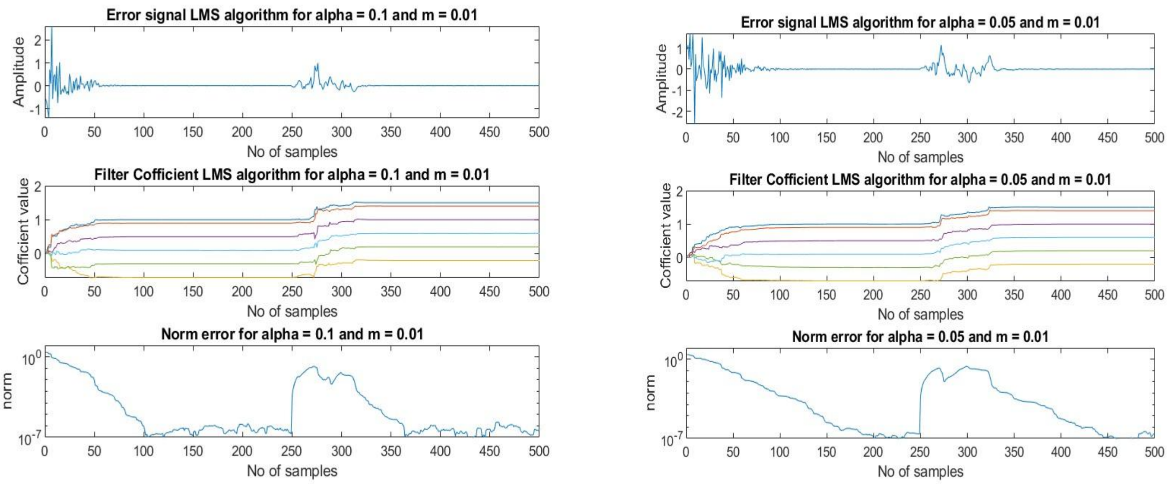


Figure 21. Error signal, filter coefficient and norm error of LMS algorithm at m =0.01 for different alphas

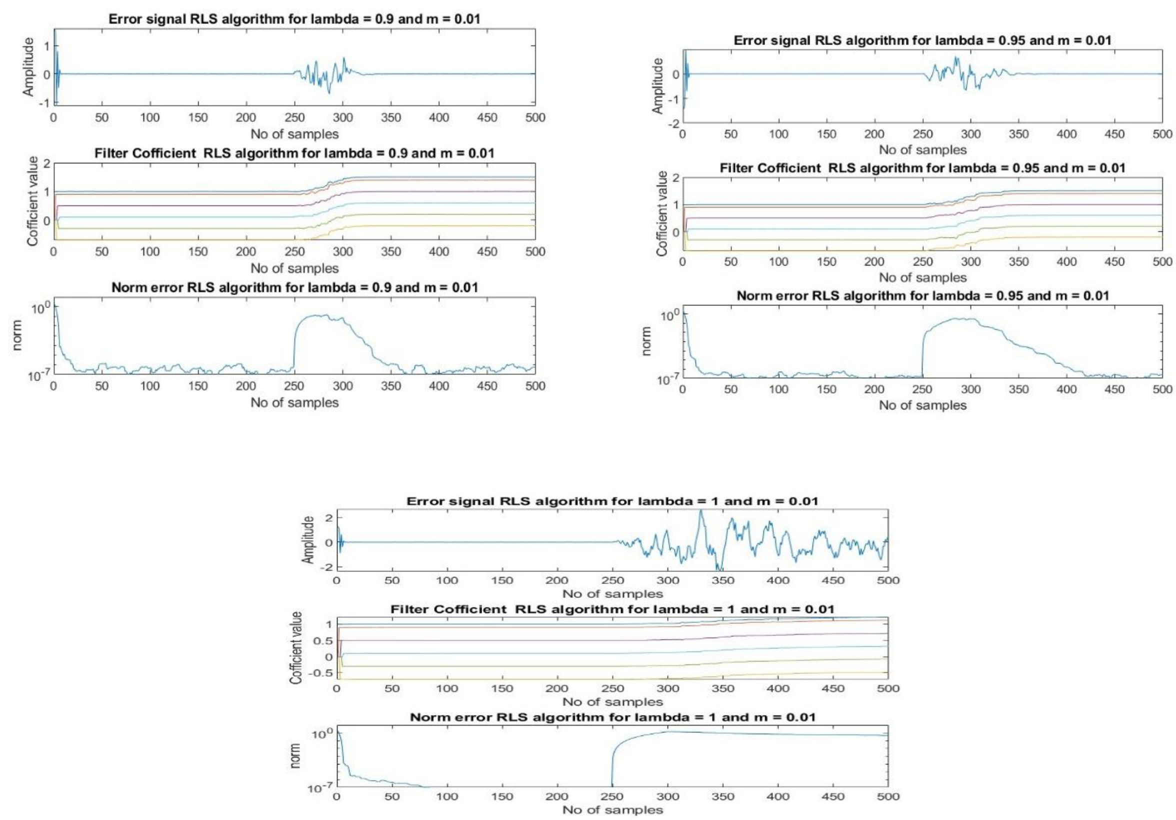
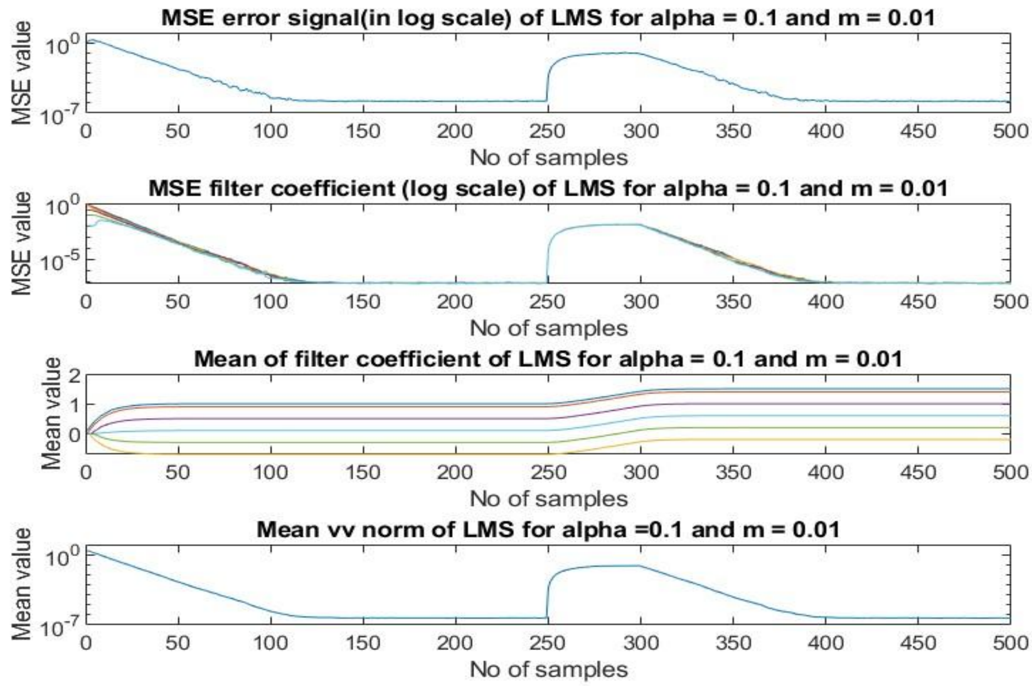


Figure 22. Error signal , filter coefficient and norm error of RLS algorithm at m =0.01 for different lambdas



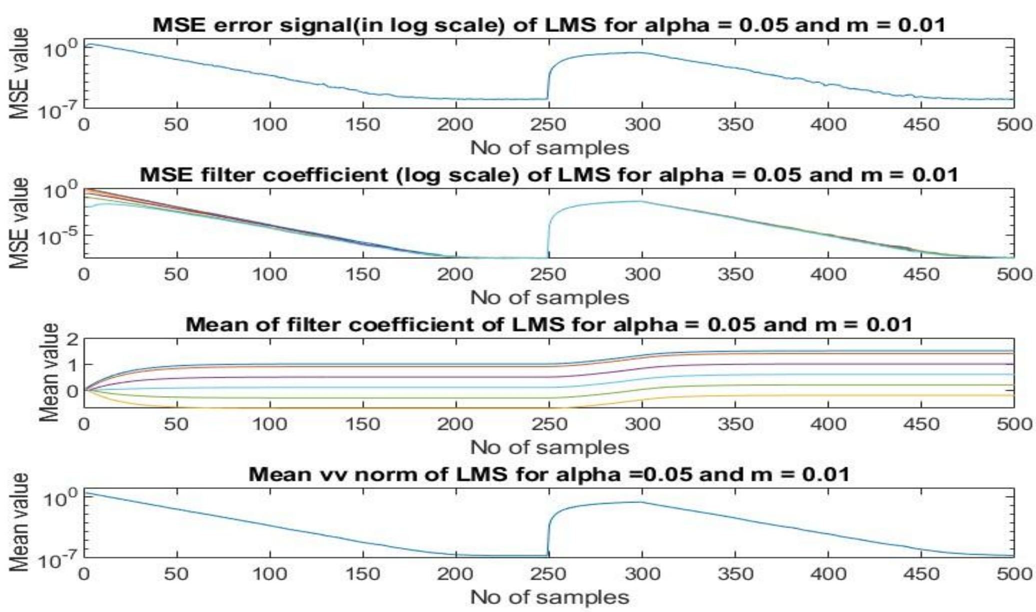
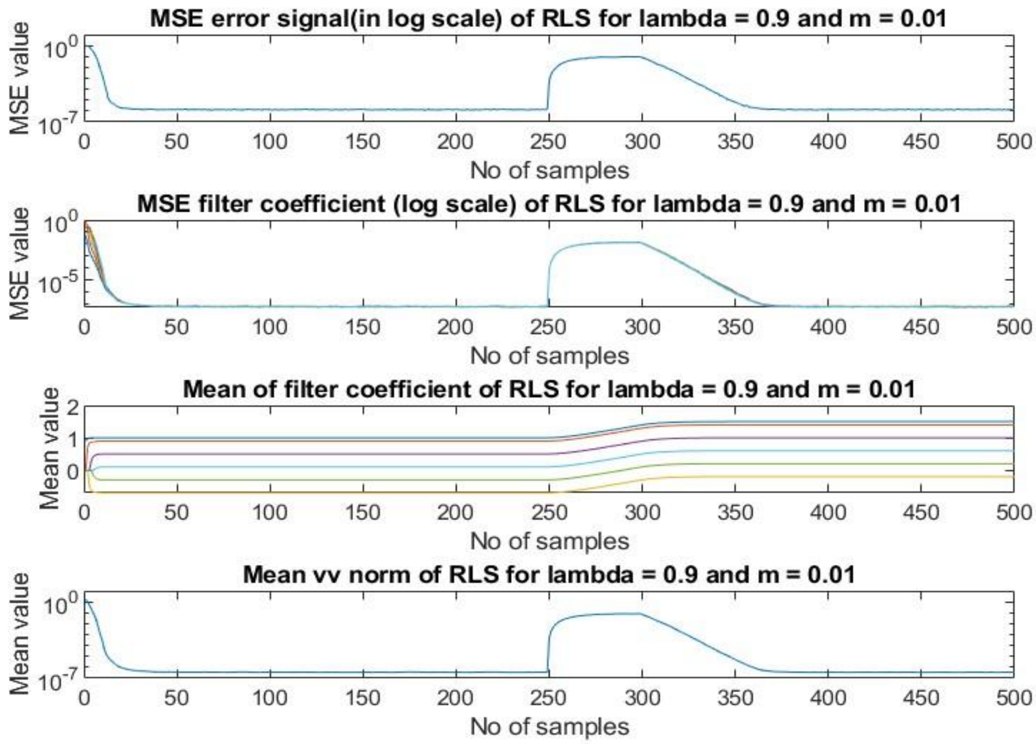
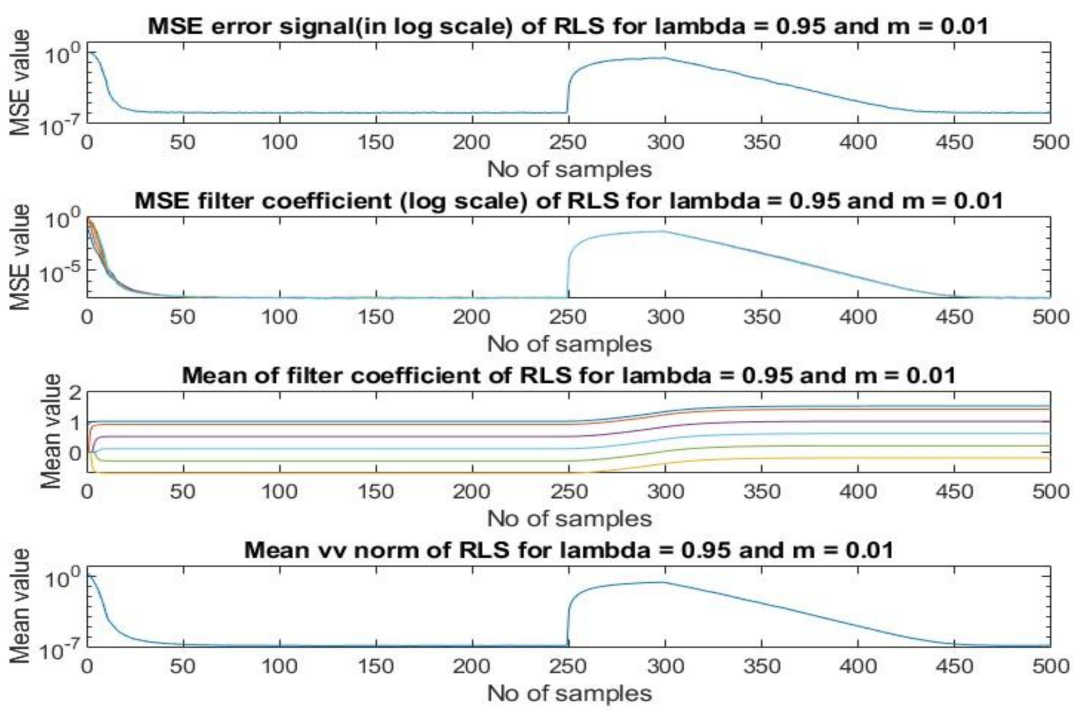


Figure 23. MSE e, MSE f, Mean of f and Mean of the norm for LMS for m=0.01 for different alphas


MSE error signal(in log scale) of RLS for lambda = 0.9 and m = 0.01
MSE value
No of samples
MSE filter coefficient (log scale) of RLS for lambda = 0.9 and m = 0.01
MSE value
No of samples
Mean of filter coefficient of RLS for lambda = 0.9 and m = 0.01
Mean value
No of samples
Mean vv norm of RLS for lambda = 0.9 and m = 0.01
Mean value
No of samples


MSE error signal(in log scale) of RLS for lambda = 0.95 and m = 0.01
MSE value
No of samples
MSE filter coefficient (log scale) of RLS for lambda = 0.95 and m = 0.01
MSE value
No of samples
Mean of filter coefficient of RLS for lambda = 0.95 and m = 0.01
Mean value
No of samples
Mean vv norm of RLS for lambda = 0.95 and m = 0.01
Mean value
No of samples

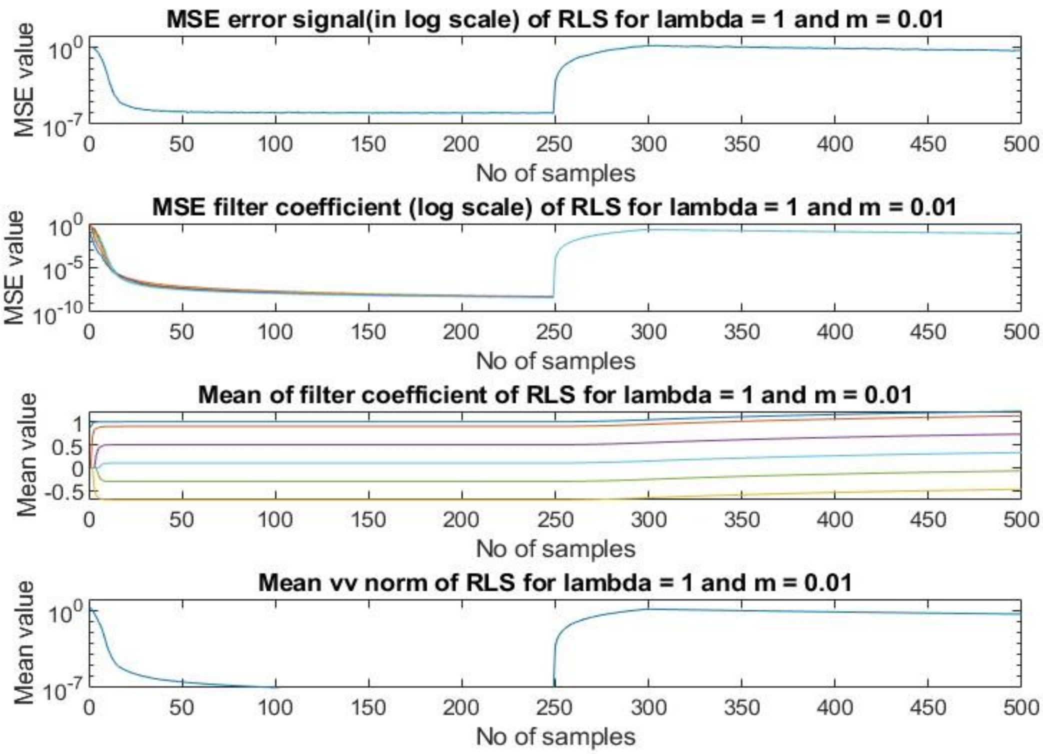


Figure 24. MSE e, MSE f, Mean of f and Mean of the norm for RLS for m=0.01 for different lambdas

Table 5 Last 50 iteration result comparison for $m = 0.01$

	MSE error	Mean value of norm
LSM ($\alpha = 0.05$)	1.3296×10^{-6}	3.152×10^{-7}
LSM ($\alpha = 0.1$)	1.5106×10^{-6}	4.9708×10^{-7}
RLS ($\lambda = 0.9$)	1.3516×10^{-6}	$3.4812.\times 10^{-7}$
RLS ($\boldsymbol{\lambda = 0.95}$)	$\mathbf{1.1526 \times 10^{-6}}$	$\mathbf{1.7911 \times 10^{-7}}$
RLS ($\lambda = 1$)	0.5013	0.4977

From the table and the figures for large slope of linear change all the algorithms converge better. With RLS of lambda 0.95 we get a better MSE error values

5. Conclusion

he simulation results from the project demonstrate that the tracking capability of the LMS and RLS algorithms in scenarios involving abrupt changes underscores the superior performance of the RLS algorithm when the lambda value is 0.95, despite its longer convergence time. In cases of exponential and linear parameter changes, systems with a high lambda value and low alpha value exhibit slower convergence and less accurate error performance, particularly when the parameter's growth (slope) is low. Furthermore, in all scenarios, when lambda equals 1, the system completely fails to track changes while maintaining stability.

References

[1]. Adaptive Signal Processing (EASP) lectures: EASP L6-7.pdf.

[2]. Adaptive Signal Processing (EASP) labs : EASP Lab 2.pdf.

[3]. Adaptive Signal Processing (EASP) lectures: EASP L7-8.pdf.

[4]. Adaptive Signal Processing (EASP) labs : EASP Lab 3.pdf.