Advanced Digital Signal Processing

Adaptive Linear Prediction Filter
(Using the LMS Algorithm)

Erick L. Oberstar

Adaptive Linear Prediction Filter Using the LMS Algorithm

A complete analysis/discussion of my results is given. Comparisons are made between my experimental results and theory.

The project is organized according to problem number, e.g., 1, 2a, 2b, etc. For each, a summary sheet that highlights the results obtained and contains discussion comments included. Consider a linear prediction filter with an input consisting of two sinusoids in noise:

$$u(n) = A_1 e^{j\omega_1 n} + A_2 e^{j\xi} e^{j\omega_2 n} + v(n)$$

where the phase difference ξ is a random variable uniformly distributed on the interval $[0,2\pi]$ and the additive noise v(n) is a complex-valued Gaussian random variable with zero mean and unit variance. The noise is white, that is, $E\{v(k)v^*(n)\} = \delta(k-n)$.

1. Find a closed form expression for the eigenvalues of the data covariance matrix as a function of the linear predictor order M, A_1, A_2, ω_1, and ω_2. Simplify your result for the special case $A = A_1 = A_2$.

2. The input $\{u(n)\}$ is applied to an adaptive four tap predictor that uses the LMS algorithm. Considering the following sets of parameters:

(i) $A^2 = 100$, $\omega_1 = \pi/8$, $\omega_2 = 3\pi/8$

(ii) $A^2 = 100$, $\omega_1 = \pi/8$, $\omega_2 = \pi/4$

(iii) $A^2 = 1000$, $\omega_1 = \pi/8$, $\omega_2 = \pi/4$

a) What is the permissible range of the step size parameter μ in each case for the LMS algorithm to be convergent in the mean?

b) What is the permissible range of the step size parameter μ in each case for the LMS algorithm to be convergent in the mean square?

c) Use the computer to generate 200 statistically independent ensembles of at least 300 samples representing $\{u(n)\}$ for parameter set (i). That is, the ith ensemble is defined as

$$u_i(n) = A_1 e^{j\omega_1 n} + A_2 e^{j\xi_i} e^{j\omega_2 n} + v_i(n), \quad n = 0,1,2,\ldots$$

Note that the same value of ξ is used for each ensemble. Apply the LMS algorithm to each ensemble. Average the squared error over the 200 ensembles to plot the learning curves of the LMS algorithm for $\mu = 10^{-4}$ and $\mu = 10^{-5}$. Estimate the corresponding values of the misadjustment by time averaging over the last 200 iterations (after convergence) of the ensemble-averaged squared error.

d) For parameter set (i), also estimate mean values for the tap weights that result from the LMS algorithm for $\mu = 10^{-4}$ and $\mu = 10^{-5}$. You may do this by averaging the steady-state values of the tap weights (obtained at the last iteration) over 200 ensembles.

e) Repeat parts c) and d) for the parameter set (iii). Use $\mu = 10^{-5}$ and $\mu = 10^{-6}$. You may need to increase the number of temporal samples to achieve convergence with LMS.

PART 1

The eigen values for the data covariance matrix ($E\{u(n-1)u^H(n-1)\}$) of the linear prediction filter were calculated as a function of the number of prediction taps, amplitudes, and frequencies of the input signal. The covariance matrix was written as sum of component covariance matrices. The sinusoidal components were rewritten in matrix form. With some manipulation this rewritten form was expressed as a matrix BB^H. Using the matrix identity that states the nonzero eigen values of BB^H are the same as the eigen values of B^HB, a 2x2 matrix was generated. The nonzero eigen values of the 2x2 matrix are the same as nonzero eigen values of the individual sinusoidal covariance matrices. The 2x2 system was solved by hand in closed form. See the attached hand calculations.

The eigen values for the three parameter sets are shown in the below table.

	$A^2 = 100, \omega_1 = \pi/8, \omega_2 = 3\pi/8$	$A^2 = 100, \omega_1 = \pi/8, \omega_2 = \pi/4$	$A^2 = 1000, \omega_1 = \pi/8, \omega_2 = \pi/4$
$\lambda_1=$	1.0e+003 *0.6623	1.0e+003 *0.7635	1.0e+003 *7.6255
$\lambda_2=$	1.0e+003 *0.1397	1.0e+003 *0.0385	1.0e+003 *0.3765
$\lambda_3=$	1.0e+003 *0.0010	1.0e+003 *0.0010	1.0e+003 *0.0010
$\lambda_4=$	1.0e+003 *0.0010	1.0e+003 *0.0010	1.0e+003 *0.0010

The eigen value spread impacts the performance of the LMS algorithm. The excess mean squared error (Misadjustment) is primarily determined by the largest eigen values. The time taken for E{w_hat(n)} to converge is limited by the smallest eigen values. The eigen value spread ($\lambda_{max}/\lambda_{min}$) for each parameter set is shown in the below table.

	$A^2 = 100, \omega_1 = \pi/8, \omega_2 = 3\pi/8$	$A^2 = 100, \omega_1 = \pi/8, \omega_2 = \pi/4$	$A^2 = 1000, \omega_1 = \pi/8, \omega_2 = \pi/4$
$\lambda_{Spread}=$	1.0e+003 * 0.6613	1.0e+003 * 0.7625	1.0e+003 * 7.6245

PART 2.a.i-2.a.iii

The criteria: $0 < \mu < \frac{2}{\lambda_{MAX}}$ is required for the LMS algorithm to be convergent in the mean, where μ is step size parameter. The eigen values for all the parameter sets were calculated and applied to the above criteria.

For convergence in the mean the range for μ is between zero and μ_{max}:

	$A^2 = 100, \omega_1 = \pi/8, \omega_2 = 3\pi/8$	$A^2 = 100, \omega_1 = \pi/8, \omega_2 = \pi/4$	$A^2 = 1000, \omega_1 = \pi/8, \omega_2 = \pi/4$
$\mu_{max} =$	0.00301972213908	0.00261968358967	0. 00026227754685

PART 2.b.i-2.b.iii

The criteria: $0 < \mu < \frac{2}{\sum_{i=1}^{M} \lambda_i}$ is required for the LMS algorithm to be convergent in the mean square, where μ is step size parameter. The eigen values for all the parameter sets were calculated and applied to the above criteria.

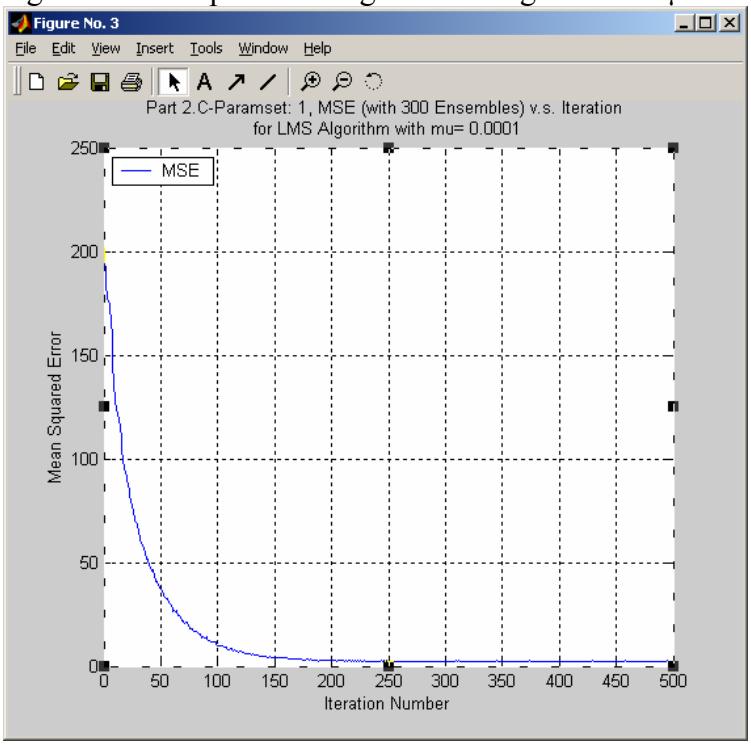
The range of μ for convergence in the mean square is between zero and μ_{max}:

	$A^2 = 100, \omega_1 = \pi/8, \omega_2 = 3\pi/8$	$A^2 = 100, \omega_1 = \pi/8, \omega_2 = \pi/4$	$A^2 = 1000, \omega_1 = \pi/8, \omega_2 = \pi/4$
$\mu_{max} =$	0.00248756218905	0.00248756218905	0.00024987506247

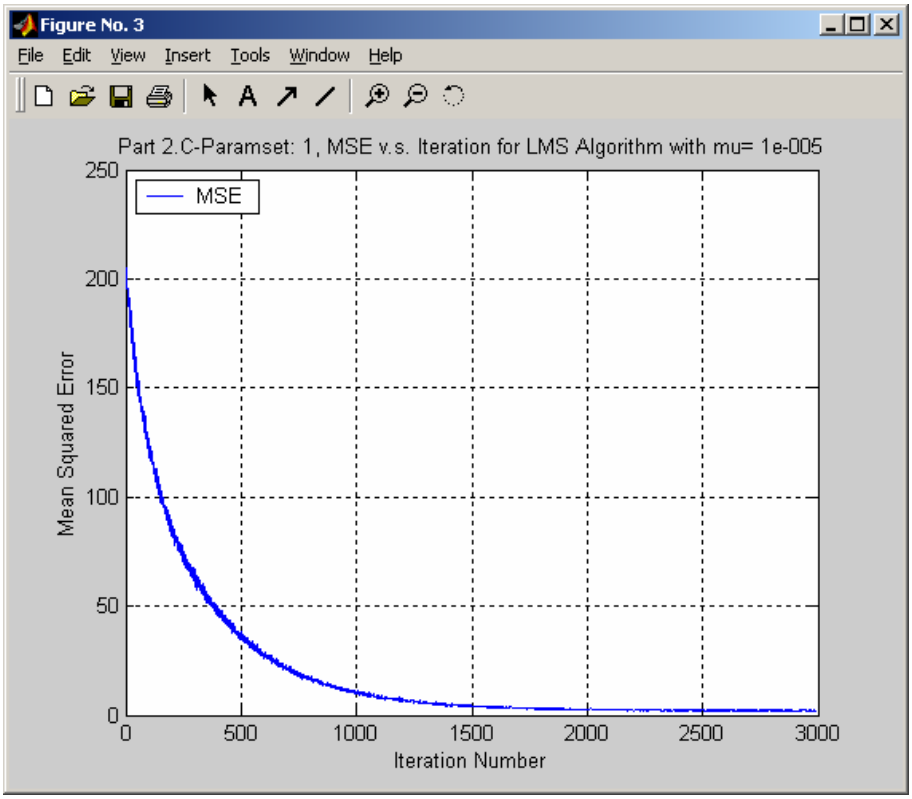
Addititionally see the attached hand calculations.

PART 2.c

The algorithms implemented in this section are shown on the attached hand written sheet. The LMS algorithm was implemented and applied to 300 statistically independent input sequences each 3000 points long for step sizes $\mu = 10^{-4}$ and $\mu = 10^{-5}$ for parameter set 1 ($A^2 = 100$, $\omega_1 = \pi/8$, $\omega_2 = 3\pi/8$). A plot of the ensemble averaged mean squared error (MSE) was generated to show convergence of the algorithm. The plot showing the learning curve for $\mu = 10^{-4}$ is shown below:



The excess MSE was calculated by averaging the last 200 iterations of the MSE for each step size(from iterations 2800 – 3000). The minimum MSE was calculated from the formula $P_M = r(0) - \underline{r}^H \underline{w}_f$. The experimental misadjustment for $\mu = 10^{-4}$ was calculated by Matlab to be $J_{ex}/J_{min} =$ 0.07986078214583. The theoretical misadjustment for $\mu = 10^{-4}$ was calculated by Matlab to be $\frac{J_{ex}}{J_{min}} = \sum_{i=1}^{M} \frac{\mu\lambda_i}{2-\mu\lambda_i} = 0.04138333436766$ for parameter set 1(i). This seems to agree with the experimentally calculated value. It is however off. The explanation for this difference is that the theoretical is calculated based on the true eigen values of the linear prediction filter. The experimental value is calculated from average value of the last 200 samples of the ensemble averaged MSE. The average value of the last 200 samples of the ensemble averaged MSE is only an approximation of $J(\infty)$. Therefore the experimental misadjustment M is calculated using an approximation to the excess MSE $J_{ex}(\infty) = J_{estimated}(\infty) - J_{min}$. Another way of looking at it is that we only have an estimated covariance matrix in the experimental calculation. Therefore we can only have estimated eigen values thus leading to the error between theoretical and experimental calculations. The plot showing the learning curve for $\mu = 10^{-5}$ is shown below:

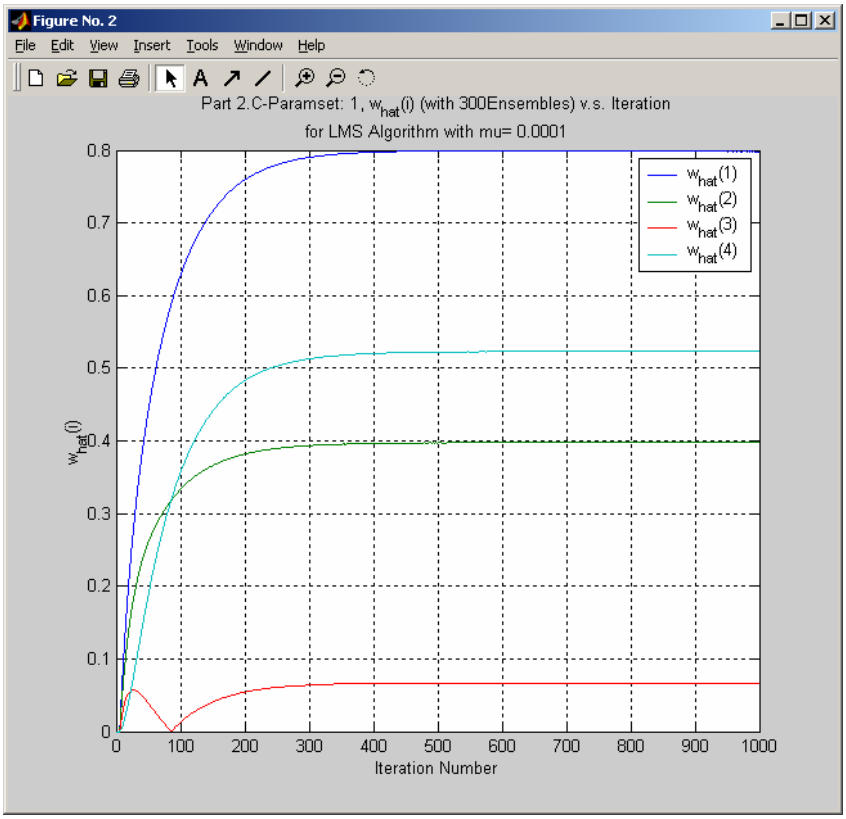


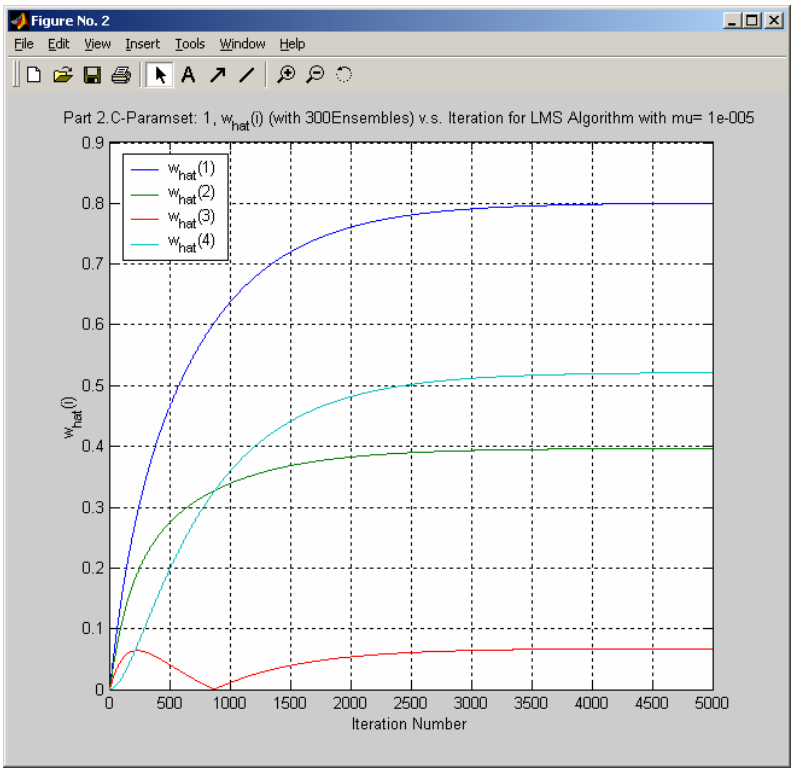
The misadjustment was calculated by averaging the last 200 iterations of the MSE for each step size. The experimental misadjustment for $\mu = 10^{-5}$ was calculated by Matlab to be $J_{ex}/J_{min} = 0.0352781$ (from iterations 2800 – 3000). The theoretical misadjustment for $\mu = 10^{-5}$ was calculated by Matlab to be $\frac{J_{ex}}{J_{min}} = \sum_{i=1}^{M} \frac{\mu\lambda_i}{2-\mu\lambda_i} = 0.00403623$ for parameter set 1(i).). This seems to agree with the experimentally calculated value. It is however off. The explanation for this difference is that the theoretical is calculated based on the true eigen values of the linear prediction filter. The experimental value is calculated from average value of the last 200 samples of the ensemble averaged MSE. The average value of the last 200 samples of the ensemble averaged MSE is only an approximation of $J(\infty)$. Therefore the experimental misadjustment M is calculated using an approximation to the excess MSE $J_{ex}(\infty) = J_{estimated}(\infty) - J_{min}$. Another way of looking at it is that we only have an estimated covariance matrix in the experimental calculation. Therefore we can only have estimated eigen values thus leading to the error between theoretical and experimental calculations.

A larger number of iterations were used for the smaller step size to allow the LMS algorithm time to converge. The results are as expected. Smaller step size takes longer to converge but yields a smaller misadjustment. The larger step size converged significantly faster but yielded a slightly larger misadjustment. There is an obvious competing tradeoff between the convergence rate and misadjustment. There appears to be approximately a ten times faster convergence rate with the larger step size. This is expected also since the average time constant $(\tau)_{mse,av} \approx \frac{1}{2\mu\lambda_{av}}$ is linear in μ, and $\mu = 10^{-4}/\mu = 10^{-5}=10$.

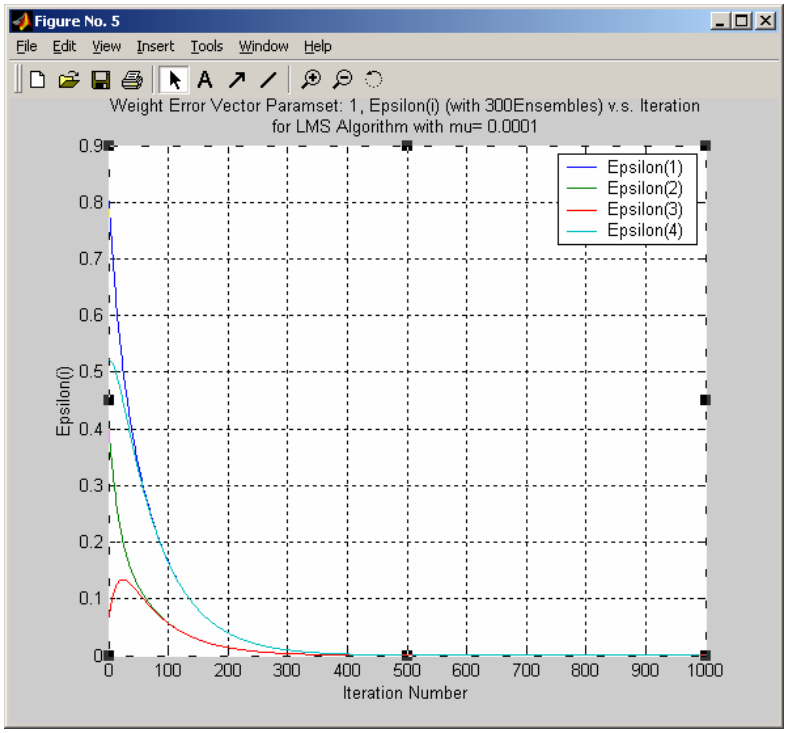
PART 2.d

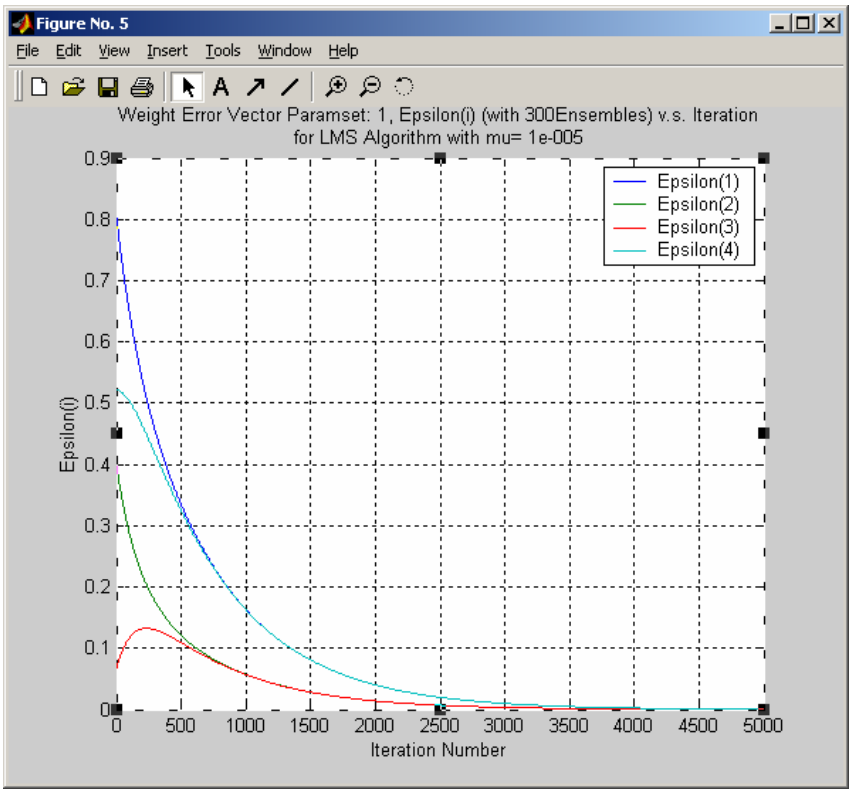
The optimum filter weights were calculated from using $\underline{w}_f = \underline{R}^{-1}\underline{r}$ (the tap weight solution for the linear prediction filter) for parameter set 1 (i). The optimum filter weights for parameter set 1 (i) are: $w_f = [0.56631 + 0.56631i \quad 0.0 + 0.39679i \quad 0.04788 - 0.04788i \quad 0.52190 + 0.0i]^T$. The ensemble-averaged learning curves were plotted for the tap weights. These plots are shown below for $\mu = 10^{-4}$ and $\mu = 10^{-5}$.





The theoretical weight error vectors ε(n) were plotted versus iteration number. The equation used to calculate the theoretical weight error vectors here and in subsequent parts is: $\underline{\varepsilon}(n+1) = (\underline{I} - \mu\underline{R})\underline{\varepsilon}(n)$. Where $\underline{\varepsilon}(n) = w_{hat(theor)}(n) - w_{opt}$ with $w_{hat(theor)}(0) = \underline{0}$. These plots follow:





It is important to note that the ensemble averaged weight vectors converge to the optimum weights at the same rate and iteration that the theoretical weight error vectors converge to zero. The experimental results for the converged tap weights agree with theory!

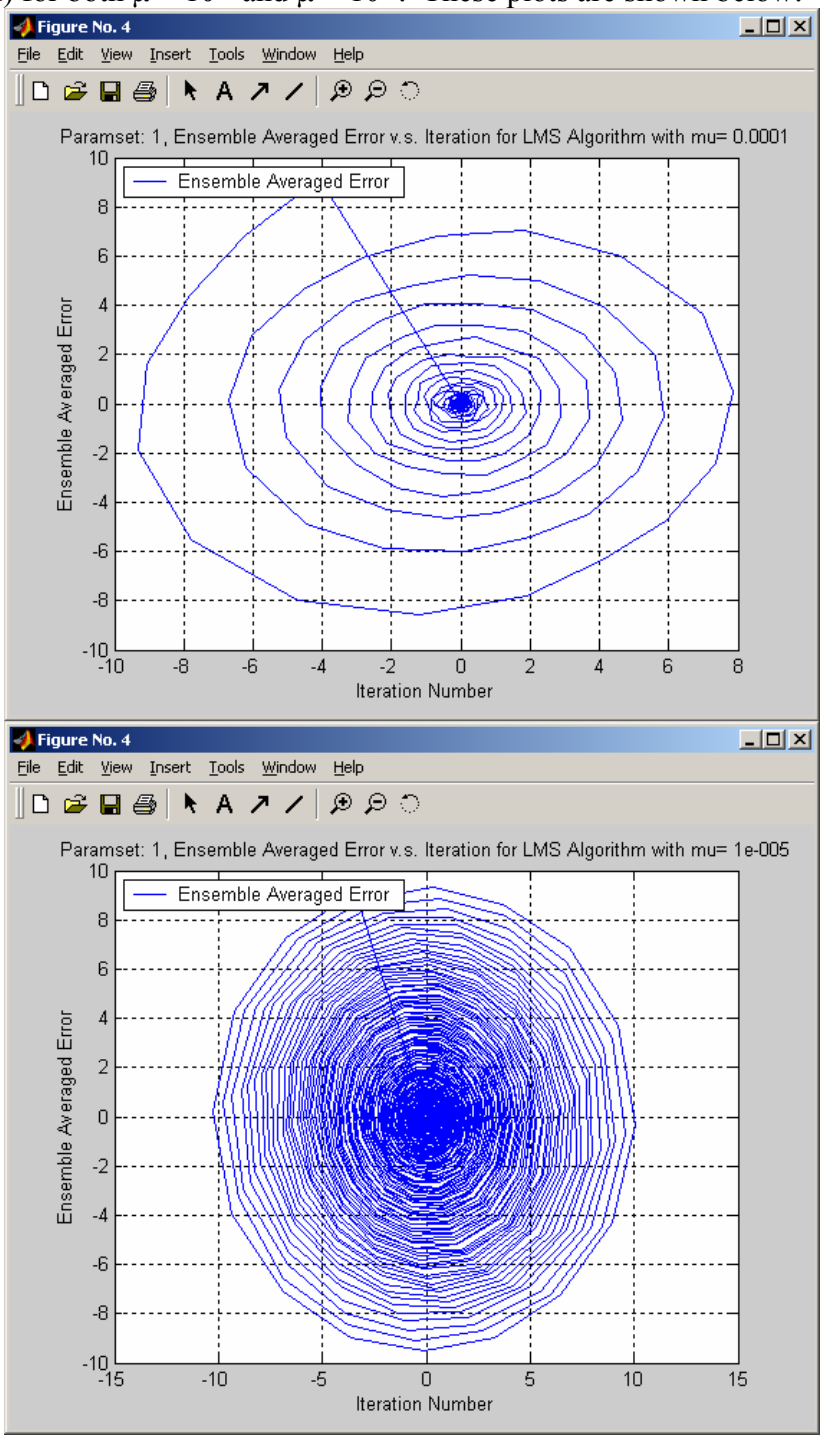
The estimate mean tap weights resulting from the LMS algorithm were calculated for $\mu = 10^{-4}$ and $\mu = 10^{-5}$ by averaging the steady state values of the tap weights over the number of ensembles run. The estimate mean tap weights for both for $\mu = 10^{-4}$ and $\mu = 10^{-5}$ are shown below:

Optimum Weights	Estimate Mean Tap Weight (300 Ensembles)	$\mu = 10^{-4}$ to 5 decimal places	$\mu = 10^{-5}$ to 5 decimal places
0.56631 + 0.56631i	w(1)	0.56191 - 0.56154i	0.56631 + 0.56631i
0.0 + 0.39679i	w(2)	-0.00031 - 0.40188i	0.0 + 0.39679i
0.04788 - 0.04788i	w(3)	0.04399 + 0.04406i	0.04788 - 0.04788i
0.52190 + 0.0i	w(4)	0.52785 - 0.00027i	0.52190 - 0.0i

The results here are exactly as expected. The tap weights for $\mu = 10^{-4}$ are slightly off. This corresponds to the larger misadjustment that results from that step size. The tap weights for $\mu = 10^{-5}$ are accurate to five decimal places but are still slightly off. This corresponds to the small misadjustment that results from the smaller step size. It is also expected in the sense that the filter weights should tune themselves closer to the optimum with a smaller step size.

It is worth noting that the tap weight vectors take longer to converge than the MSE. This is an artifact of the eigen value spread for this parameter set.

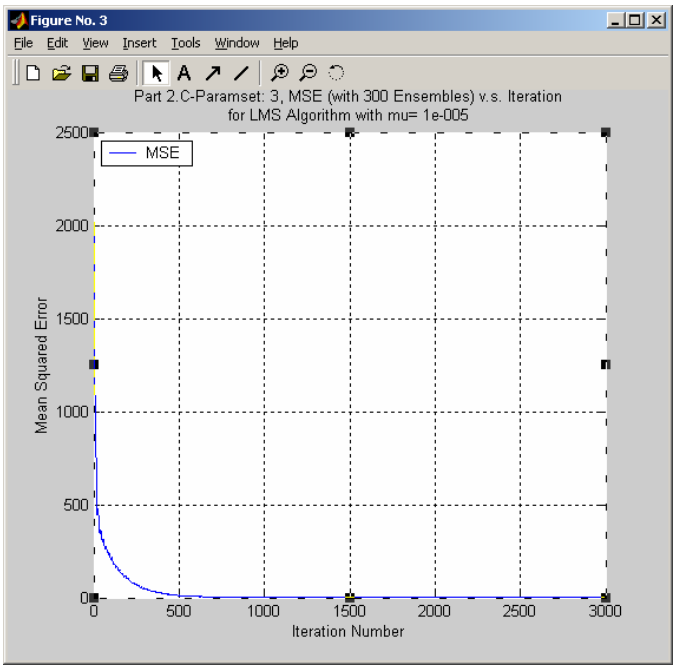
The ensemble averaged error performances (the complex error versus iteration) were plotted for parameter set 1(i) for both $\mu = 10^{-4}$ and $\mu = 10^{-5}$. These plots are shown below:

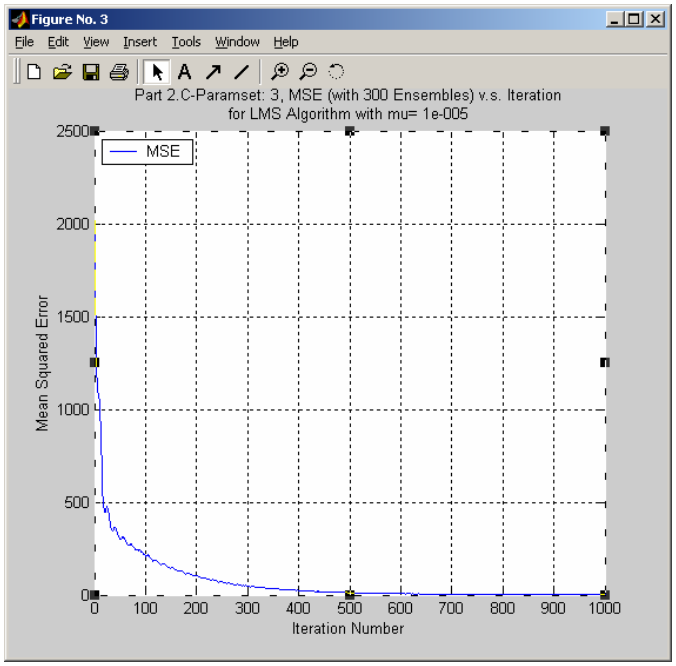


The above error plots show the LMS algorithm zeroing in on its target of zero error.

PART 2.e (c)

The LMS algorithm was implemented and applied to 300 statistically independent input sequences each 3000 points long for step sizes $\mu = 10^{-5}$ and $\mu = 10^{-6}$ for parameter set 3 ($A^2 = 1000$, $\omega_1 = \pi/8$, $\omega_2 = \pi/8$). A plot of the ensemble averaged mean squared error (MSE) was generated to show convergence of the algorithm. The plot showing the learning curve for $\mu = 10^{-4}$ is shown below:

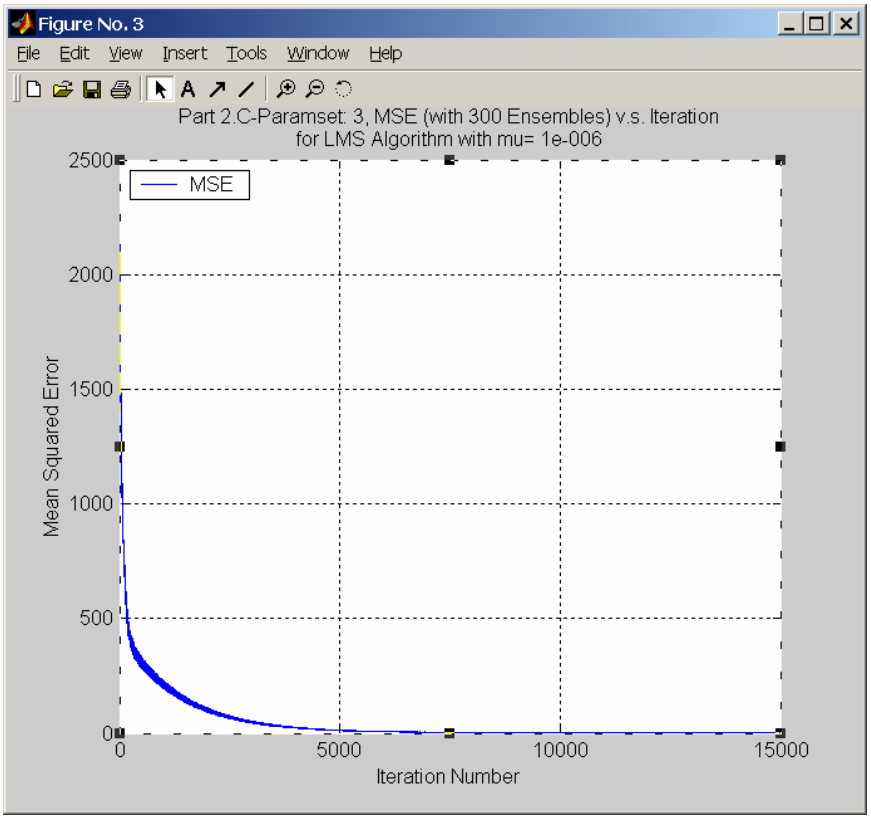




Zoomed in

The misadjustment was calculated by averaging the last 200 iterations of the MSE for each step size. The misadjustment for $\mu = 10^{-5}$ was calculated by Matlab to be $J_{ex}/J_{min} = 0.08323589$. The theoretical misadjustment for $\mu = 10^{-5}$ was calculated by Matlab to be $\frac{J_{ex}}{J_{min}} = \sum_{i=1}^{M} \frac{\mu\lambda_i}{2-\mu\lambda_i} =$ 0.041534884 for parameter set 3(iii).). This seems to agree with the experimentally calculated value. It is however off. The explanation for this difference is that the theoretical is calculated based on the true eigen values of the linear prediction filter. The experimental value is calculated from average value of the last 200 samples of the ensemble averaged MSE. The average value of the last 200 samples of the ensemble averaged MSE is only an approximation of $J(\infty)$. Therefore the experimental misadjustment M is calculated using an approximation to the excess MSE $J_{ex}(\infty) = J_{estimated}(\infty) - J_{min}$. Another way of looking at it is that we only have an estimated covariance matrix in the experimental calculation. Therefore we can only have estimated eigen values thus leading to the error between theoretical and experimental calculations.

The LMS algorithm requires more than 3000 iterations to converge to long enough to average the last 200 iterations for calculating misadjustment with $\mu = 10^{-6}$. Therefore the $\mu = 10^{-6}$ learning curve was extended to 15000 iterations. The plot showing the learning curve for $\mu = 10^{-6}$ is shown below:



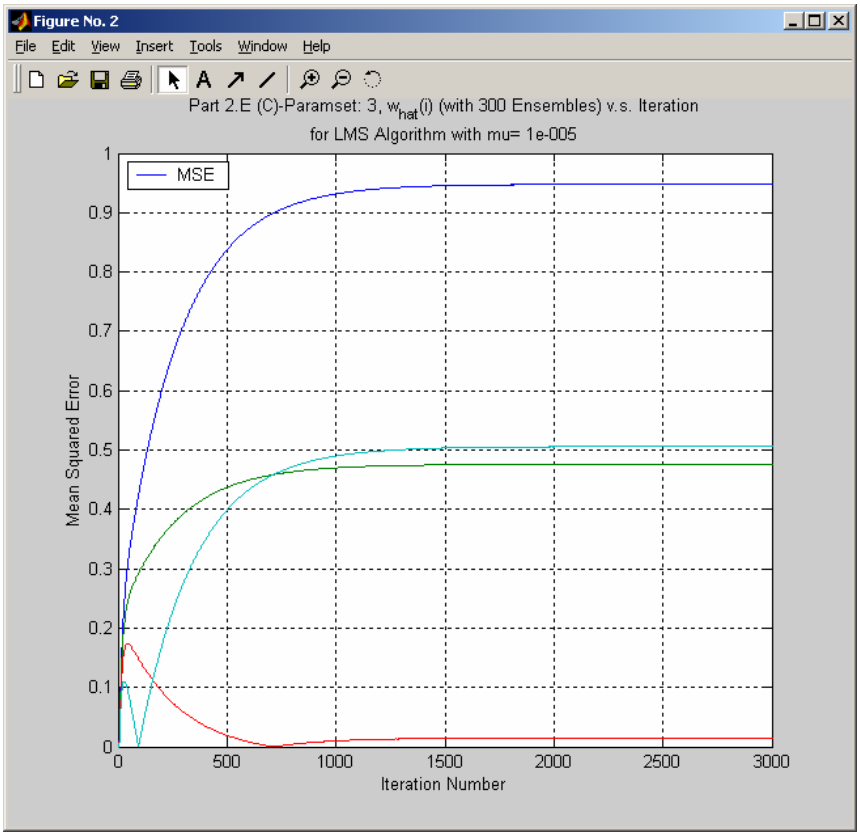
The misadjustment was calculated by averaging the last 200 iterations of the MSE for each step size. The experimental misadjustment for $\mu = 10^{-6}$ was calculated by Matlab to be $J_{ex}/J_{min} =$

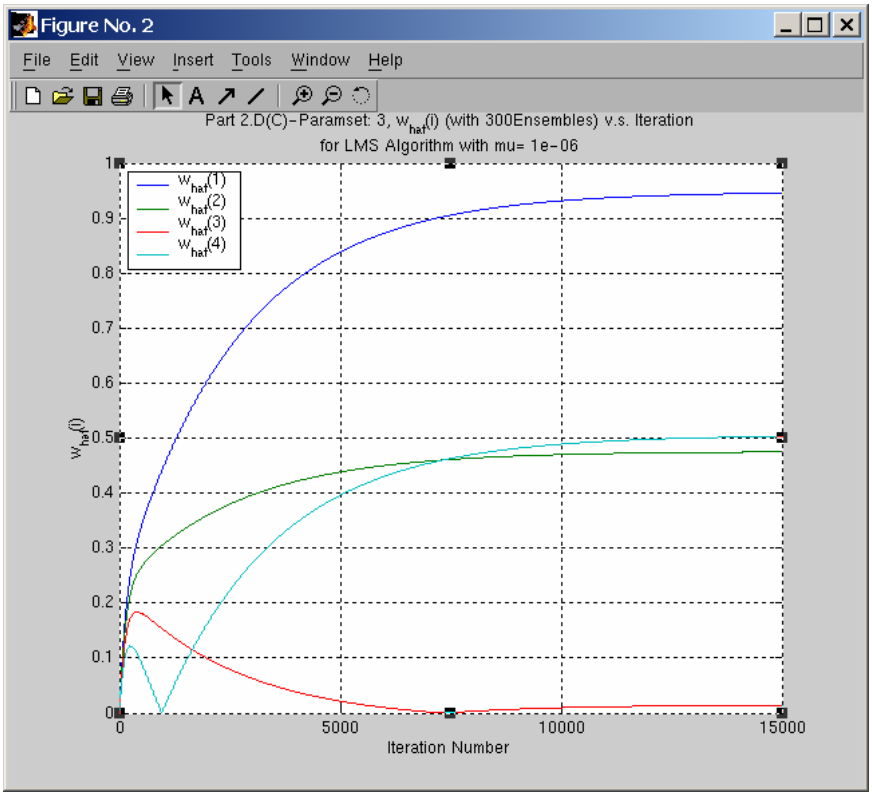
0.0146986 (from iterations 14800-15000). The theoretical misadjustment for $\mu = 10^{-6}$ was calculated by Matlab to be $\frac{J_{ex}}{J_{min}} = \sum_{i=1}^{M} \frac{\mu\lambda_i}{2-\mu\lambda_i} = 0.0401808$ for parameter set 3(iii). This seems to agree with the experimentally calculated value. It is off however. The explanation for this difference is that the theoretical is calculated based on the true eigen values of the linear prediction filter. The experimental value is calculated from average value of the last 200 samples of the ensemble averaged MSE. The average value of the last 200 samples of the ensemble averaged MSE is only an approximation of $J(\infty)$. Therefore the experimental misadjustment M is calculated using an approximation to the excess MSE $J_{ex}(\infty) = J_{estimated}(\infty) - J_{min}$. Another way of looking at it is that we only have an estimated covariance matrix in the experimental calculation. Therefore we can only have estimated eigen values thus leading to the error between theoretical and experimental calculations.

A larger number of iterations were used for the smaller step size to allow the LMS algorithm time to converge. The results are as expected. Smaller step size takes longer to converge but yields a smaller misadjustment. The larger step size converged significantly faster but yielded a slightly larger misadjustment. There is an obvious competing tradeoff between the convergence rate and misadjustment. There appears to be approximately a ten times faster convergence rate with the larger step size. This is expected also since the average time constant $(\tau)_{mse,av} \approx \frac{1}{2\mu\lambda_{av}}$ is linear in μ, and $\mu = 10^{-5}/\mu = 10^{-6}=10$.

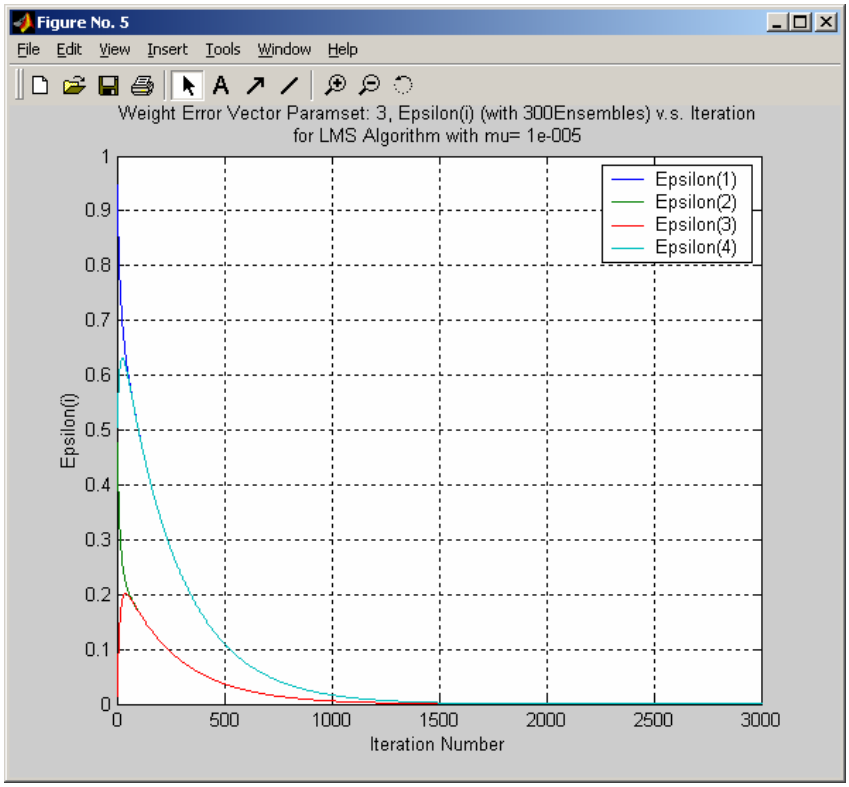
PART 2.e (d)

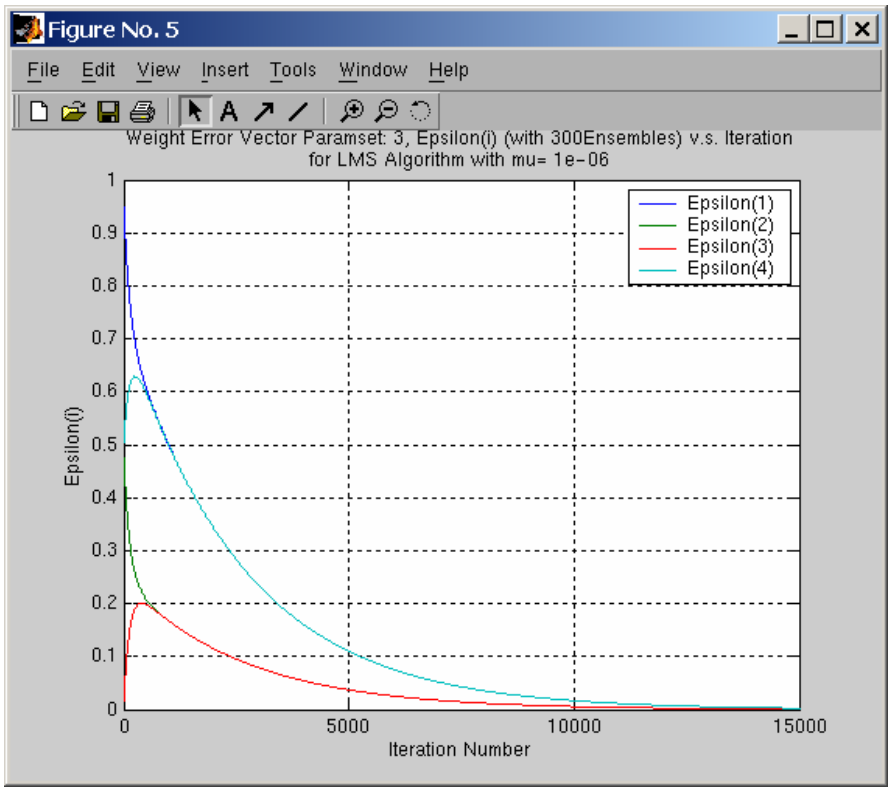
The optimum filter weights were calculated from using $w_f = R^{-1}r$ (the tap weight solution for the linear prediction filter) for parameter set 3 (iii). The optimum filter weights for parameter set 3 (iii) are: $w_f = [0.78846 + 0.52683i \quad 0.18202 + 0.43944i \quad 0.00298 - 0.01496i \quad 0.35749 - 0.35749i]^T$. The learning curves for the tap weights were plotted for both . These plots are shown below:



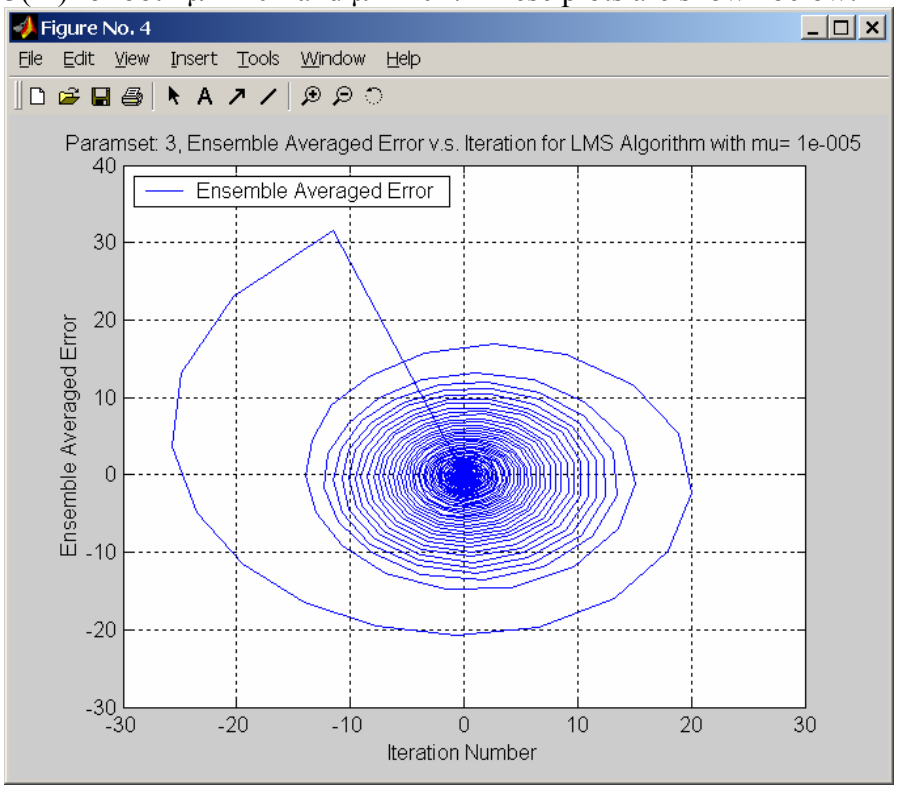


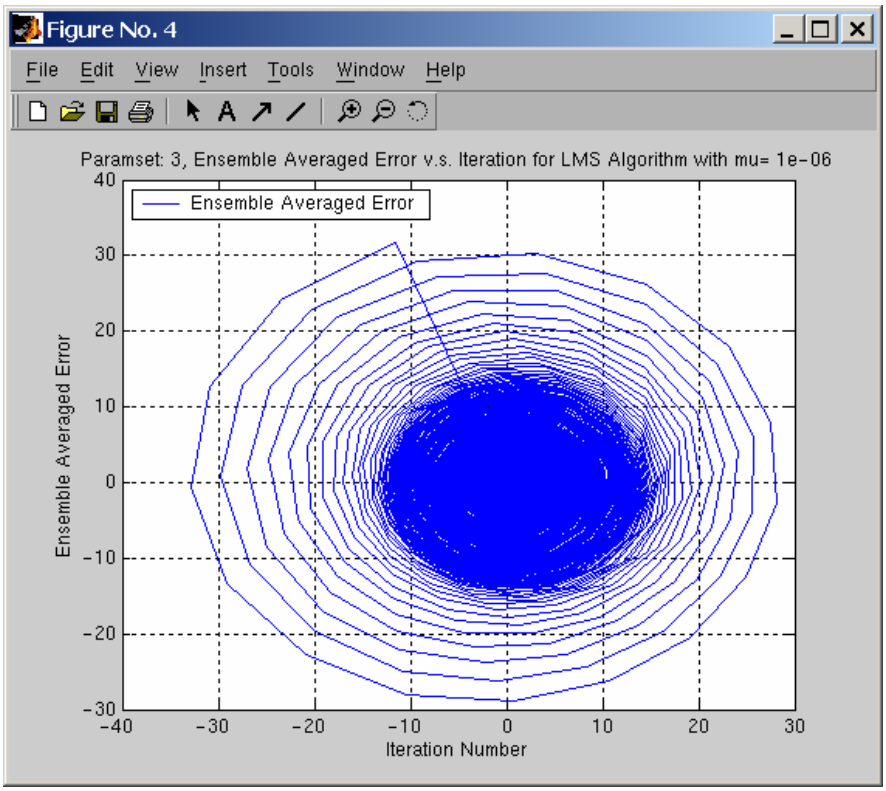
The theoretical weight error vectors were plotted versus iteration number. These plots follow:





The ensemble averaged error performances (the complex error versus iteration) were plotted for parameter set 3(iii) for both $\mu = 10^{-5}$ and $\mu = 10^{-6}$. These plots are shown below:





The above error plots show the LMS algorithm zeroing in on its target of zero error.

The estimate mean tap weights resulting from the LMS algorithm were calculated for $\mu = 10^{-5}$ and $\mu = 10^{-6}$ by averaging the steady state values of the tap weights over the number of ensembles run. The estimate mean tap weights for both for $\mu = 10^{-5}$ and $\mu = 10^{-6}$ are shown below:

Optimum Weights	Estimate Mean Tap Weight (300 Ensembles)	$\mu = 10^{-5}$ to 5 decimal places	$\mu = 10^{-6}$ to 5 decimal places
0.78846 + 0.52683i	w(1)	0.78774 - 0.52639i	0.78632 - 0.52540i
0.18202 + 0.43944i	w(2)	0.18224 - 0.43999i	0.18170 - 0.43865i
0.00298 - 0.01496i	w(3)	0.00284 + 0.01437i	0.00282 + 0.01409i
0.35749 - 0.35749i	w(4)	0.35790 + 0.35792i	0.35572 + 0.35573i

The results here are as expected. The tap weights for $\mu = 10^{-5}$ are slightly off. This corresponds to the larger misadjustment that results from that step size. The tap weights for $\mu = 10^{-6}$ are also very accurate but are still slightly off. This corresponds to the small misadjustment that results from the smaller step size. It is also expected that the filter weights would tune themselves closer to the optimum with a smaller step size.

Conclusion:
Note: The Matlab source code used to generate the plots follows this conclusion.

The LMS algorithm implemented in this project agrees well with the theoretical best-case tap weights and MSE/misadjustment. Obvious trade offs were observed with regards to step size versus misadjustment. The tap weights eventually converge bounce around the optimum tap weights but never equal the optimum. The overall agreement of the experimental results with the theoretical results improves as the number of iterations is increased.

For this project we clearly violate the independence assumption of the statistical analysis of the LMS algorithm for the linear prediction filter. I.e. the linear prediction filter is subject to the shifting property of the input data. This violation appears when calculating the excess MSE i.e. equation 9.64 and 9.65 in the textbook. However the when evaluating the expectations in these equations ($E\{u(n)u^H(n)\}$ & $E\{\varepsilon(n)\,\varepsilon^H(n)\}$) the correlation structure of the sources producing u(n) and ε(n) are preserved. Thus the independence theory retains enough information about the structure of the adaptive process to allow the LMS algorithm to work.

Other Conclusions:
I need a much faster computer with a boatload more RAM to run simulations in a more reasonable amount of time. I need a hobby to do while waiting for simulations to run on my old computer.