Problem 1

i.

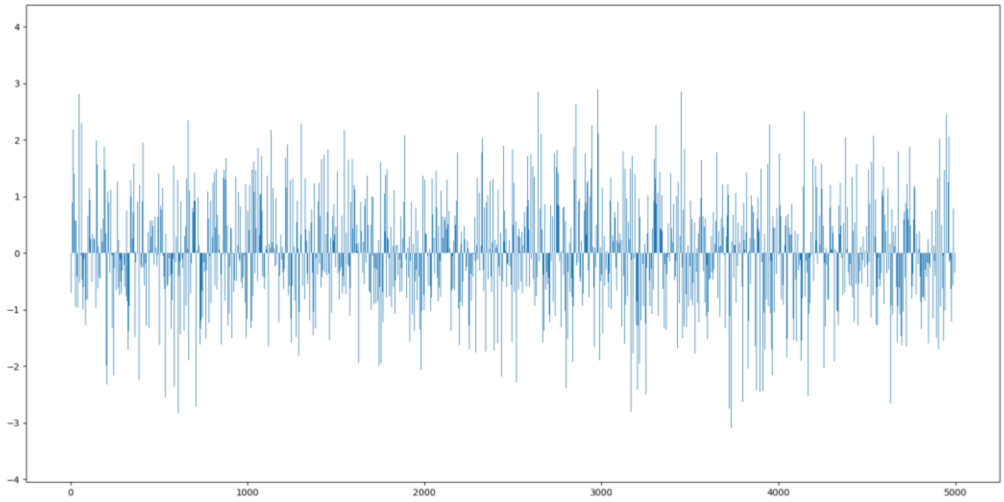

Figure 1 $y[n]$ $(Gaussian\ random\ variable\ with\ zero\ mean\ and\ unit\ variance)$

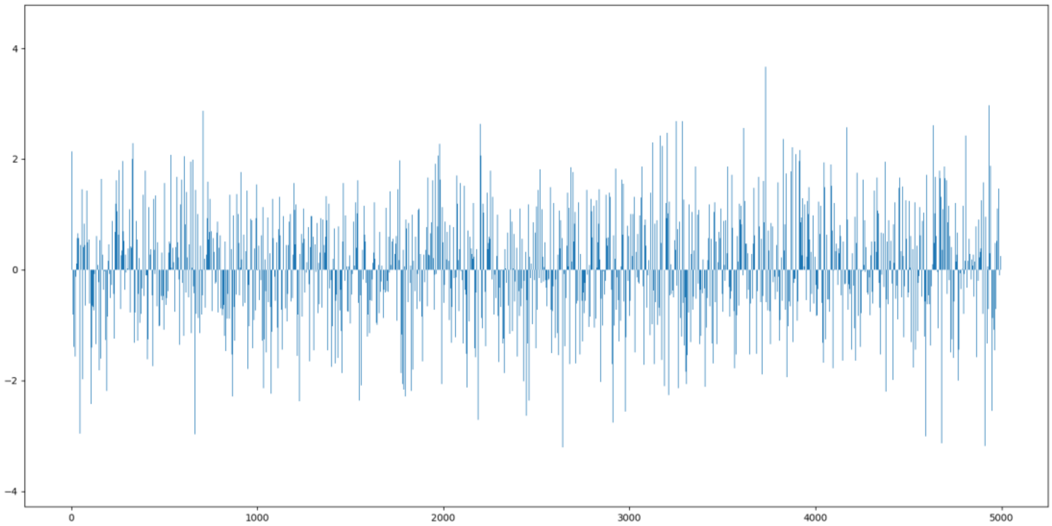

Figure 2 $t_n = -0.8y_n + 0.7y_{n-1}$

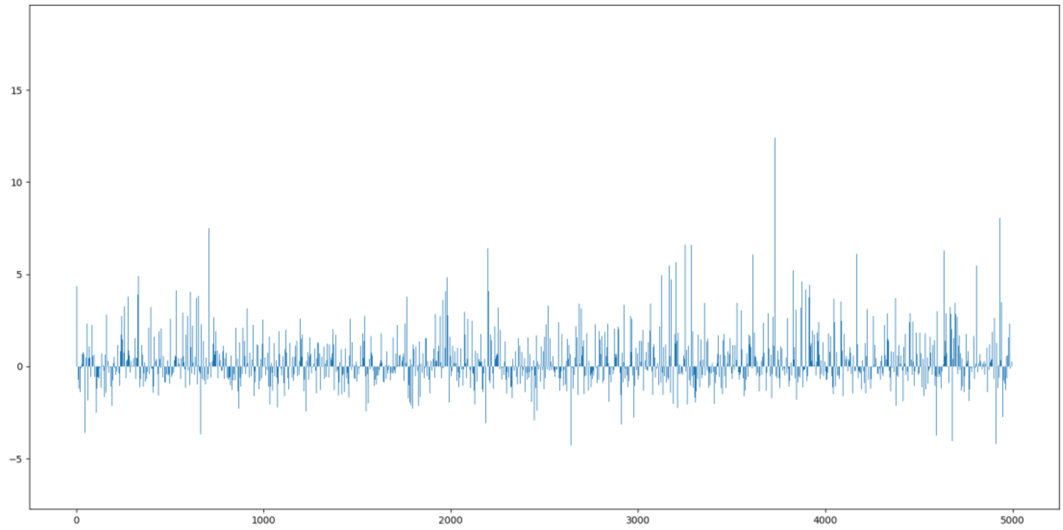

Figure 3 $q_n = t_n + 0.25t_n^2 + 0.11t_n^3$

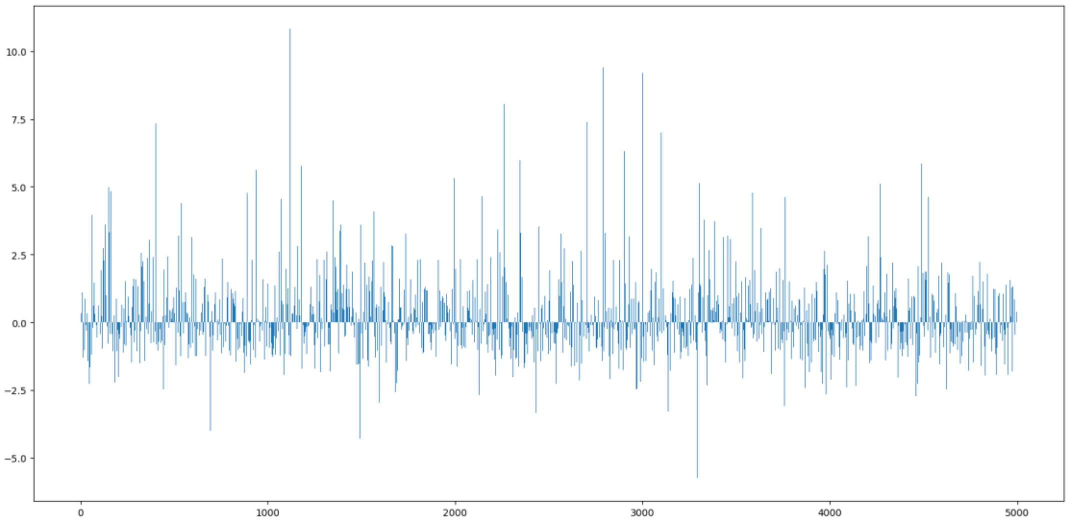

Figure 4 $x_n = q_n + noise$

In this problem, the input signal y_n is Gaussian random variable with zero mean and unit variance which is shown in Figure 1. After passing a linear filter and a nonlinear filter, we get q_n. The signal q_n is then corrupted by 15dB AWGN and then it is observed as x_n, which is the input of the adaptive filter.

ii.

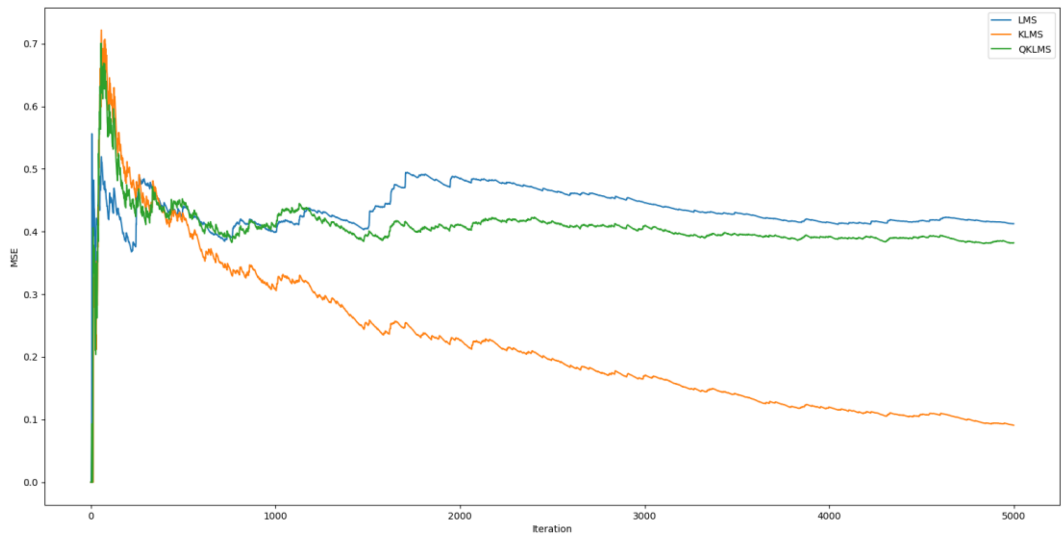

Figure 5 MSE in LMS, KLMS, QKLMS

In this problem, I set the time delay = 2 and the filter length = 5.
For drawing Figure 5, I set step size = 0.1, kernel width = 3, quantization size = 1.
From Figure 5 we can see that KLMS algorithm has the lowest MSE. The MSE of KLMS algorithm is lower than QKLMS.

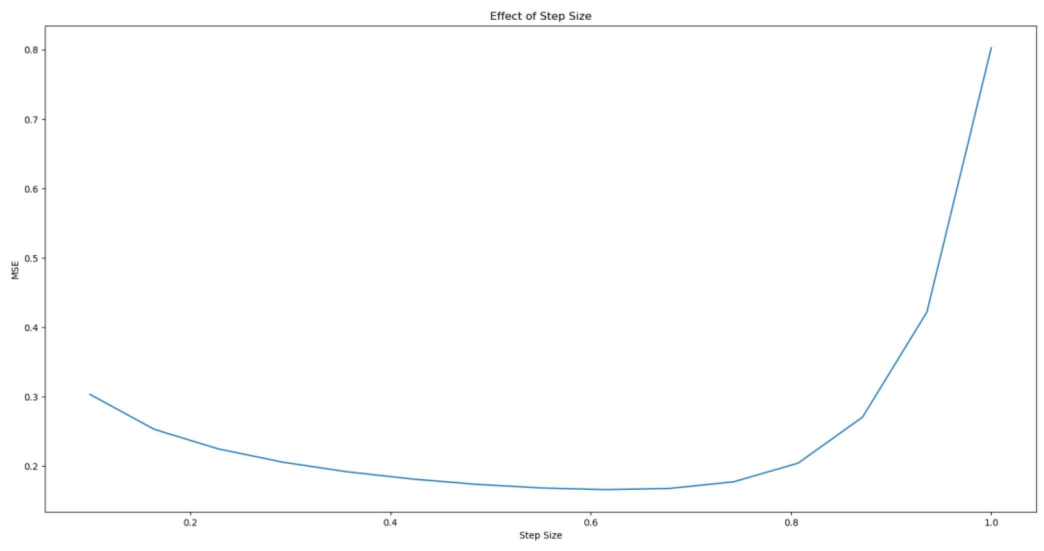

Figure 6 Effect of Step Size in KLMS (kernel width=3)

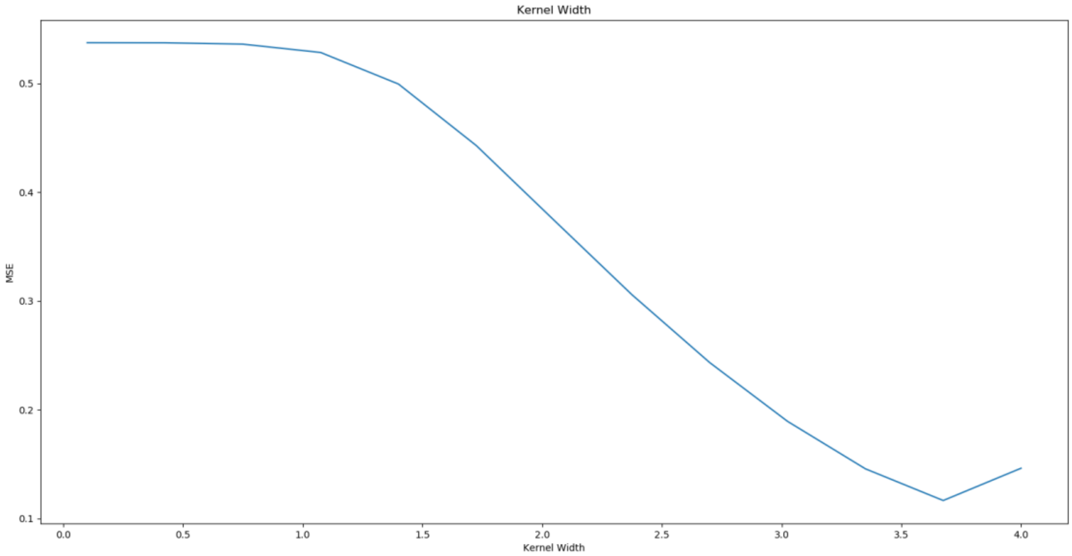

Figure 7 Effect of Kernel Width in KLMS (step size=0.6)

In KLMS algorithm, I set kernel width = 3 to find the best step size. From figure 6 we can see that when step size = 0.6, the KLMS algorithm has the lowest MSE, which means it has the best performance.

I set step size = 0.6 to find the best kernel width. To find the best kernel width, I use Silverman's rule to calculate a starting value($\sigma = 1.47$). From figure 7 we can see that when kernel width $\sigma = 3.6$, the KLMS algorithm has the lowest MSE, which means it has the best performance.

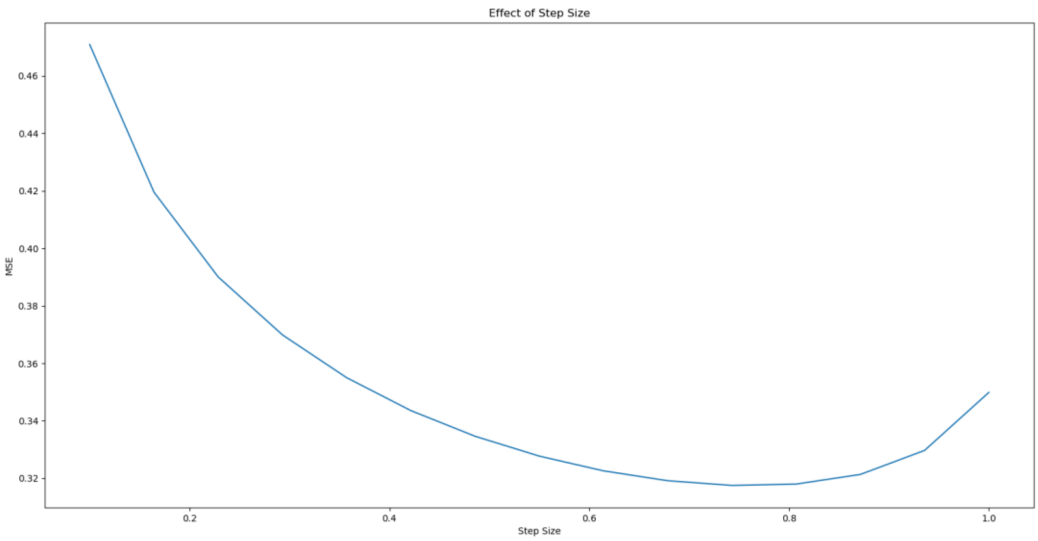

Figure 8 Effect of Step Size in QKLMS (kernel width=3)

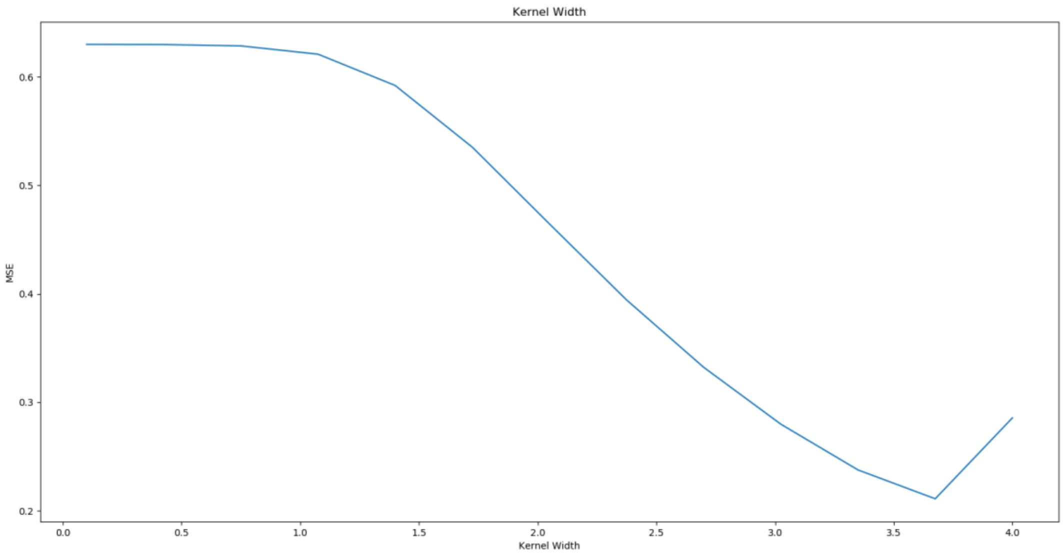

Figure 9 Effect of Kernel Width in QKLMS (step size=0.6)

In QKLMS algorithm, I set kernel width = 3 and quantization size = 1 to find the best step size. From figure 8 we can see that when step size = 0.7, the QKLMS algorithm has the lowest MSE, which means it has the best performance.

I set step size = 0.6 to find the best kernel width. From figure 9 we can see that when kernel width σ = 3.6, the KLMS algorithm has the lowest MSE, which means it has the best performance. From figures above we can see that the best kernel size for KLMS and QKLMS is the same.

iii.

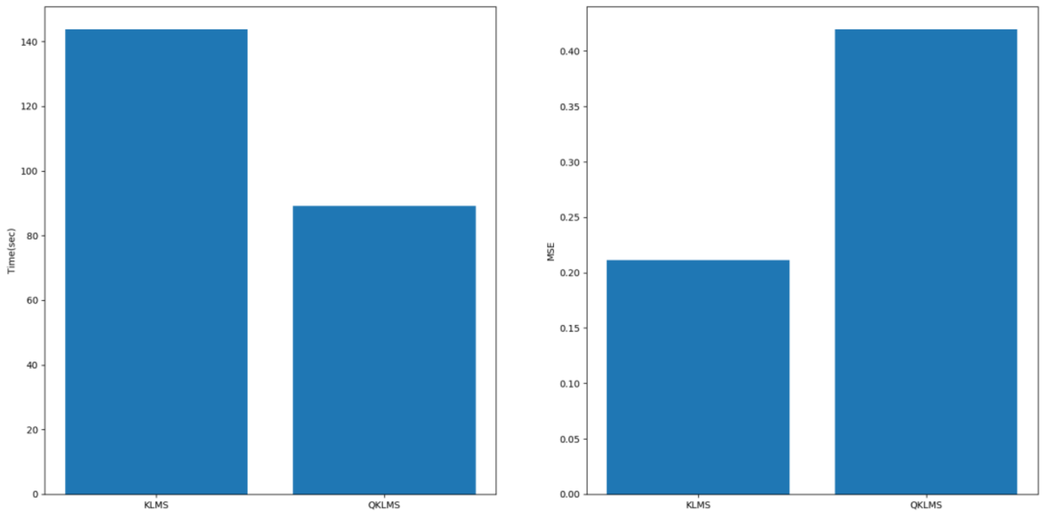

Figure 10 Running time and MSE between KLMS and QKLMS

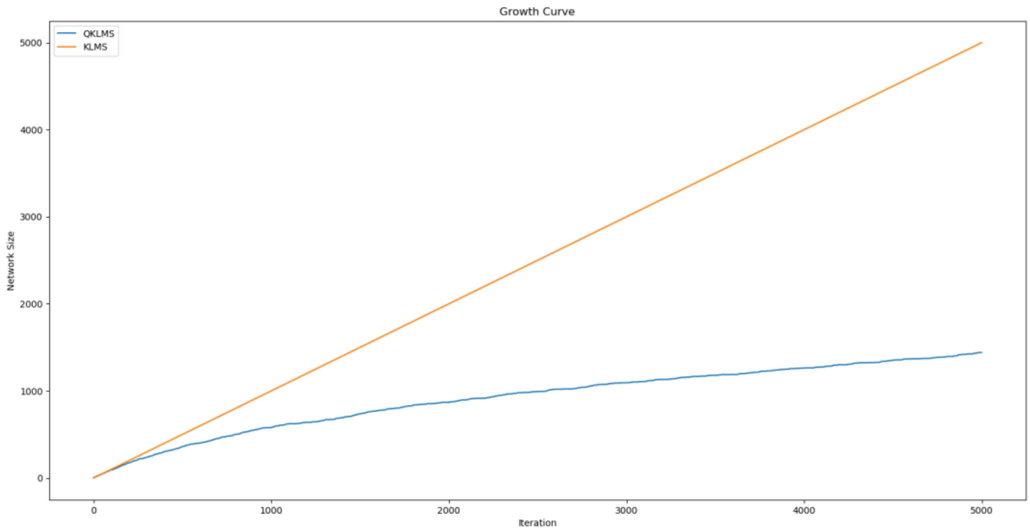

Figure 11 Growth curve between KLMS and QKLMS

In this problem, I set the time delay = 2 and the filter length = 5.
For KLMS, step size = 0.6, kernel width σ = 3.6.
For QKLMS, step size = 0.6, kernel width σ = 3.6, quantization size = 1.
From figure 10 we can see that running time of QKLMS is shorter than running time of KLMS, but the MSE of QKLMS is larger than MSE of KLMS. This is because we use the quantization method to compress the input space and hence to compact the RBF structure of the kernel adaptive filter by reducing the network size. We can see in Figure 11 that the network size of QKLMS is about 1500, which is smaller than KLMS. So that the performance of KLMS is better than the performance of QKLMS.