HOMEWORK 3

i.

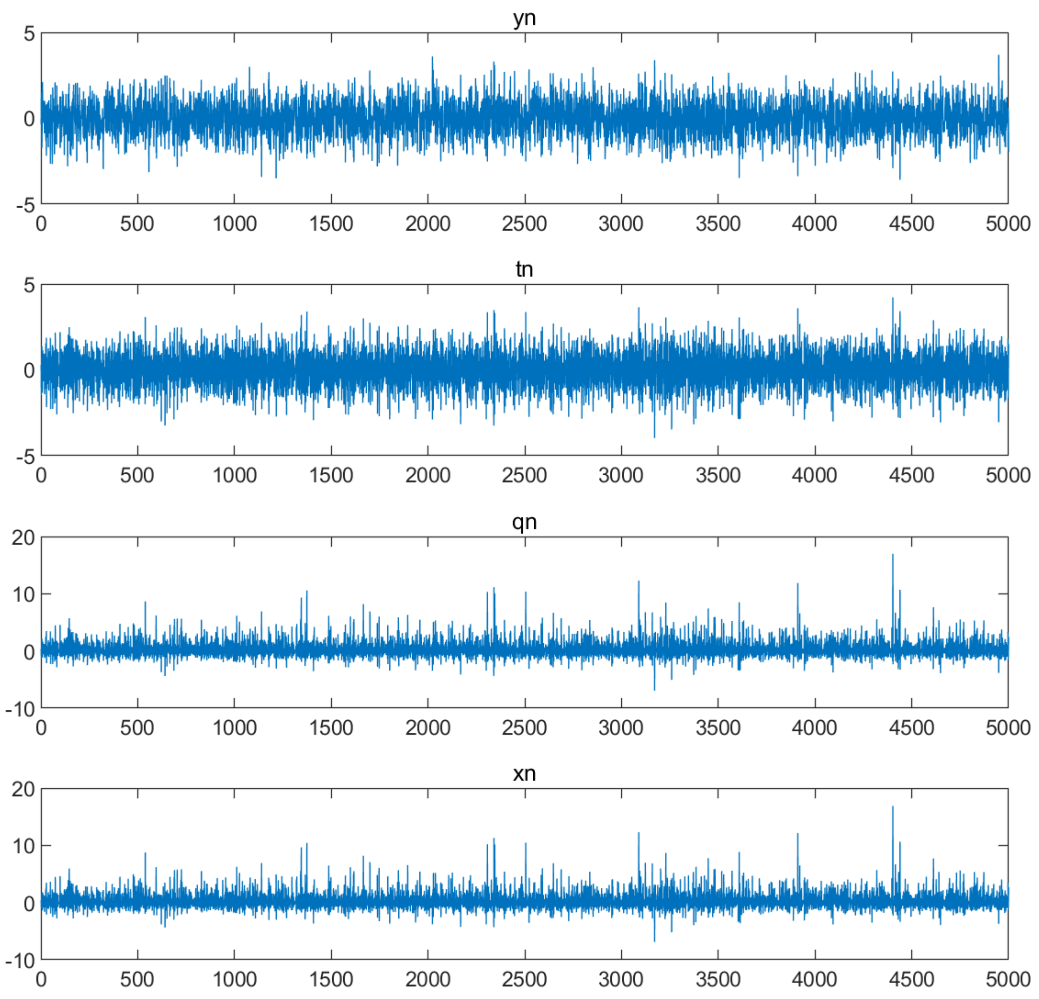

Figure 1. Generated points according to the figure in Problem 1

Figure 1. shows required yn, tn, qn and xn.

ii.

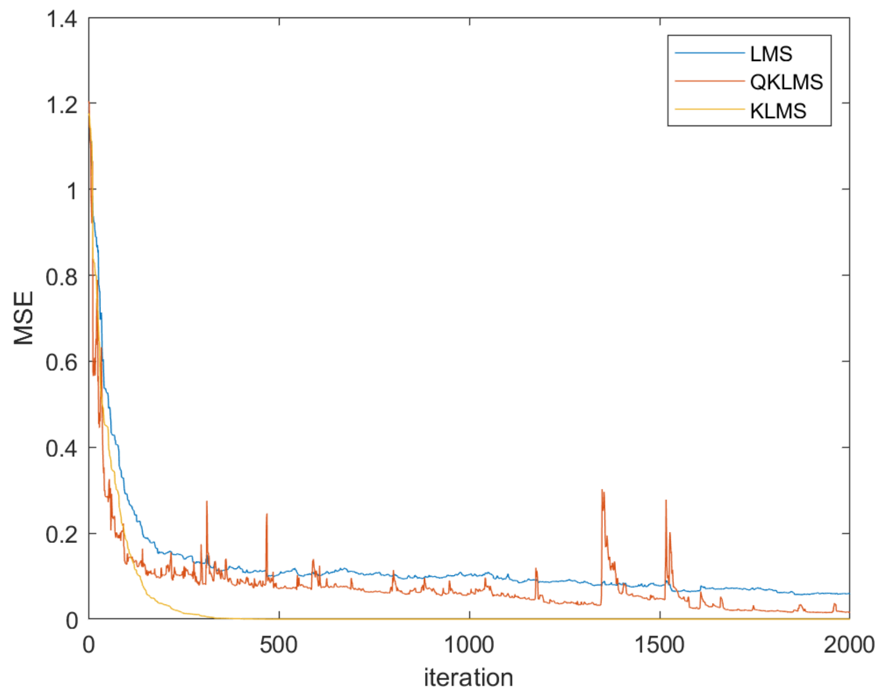

Figure 2. MSE of different methods

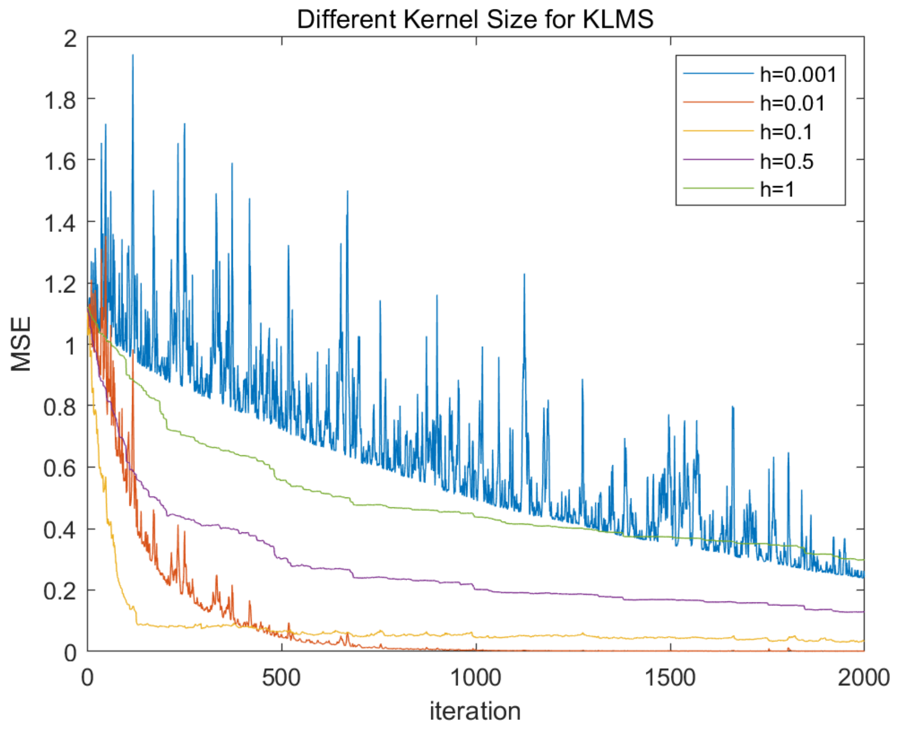

Figure 3. MSE of different kernel size for KLMS

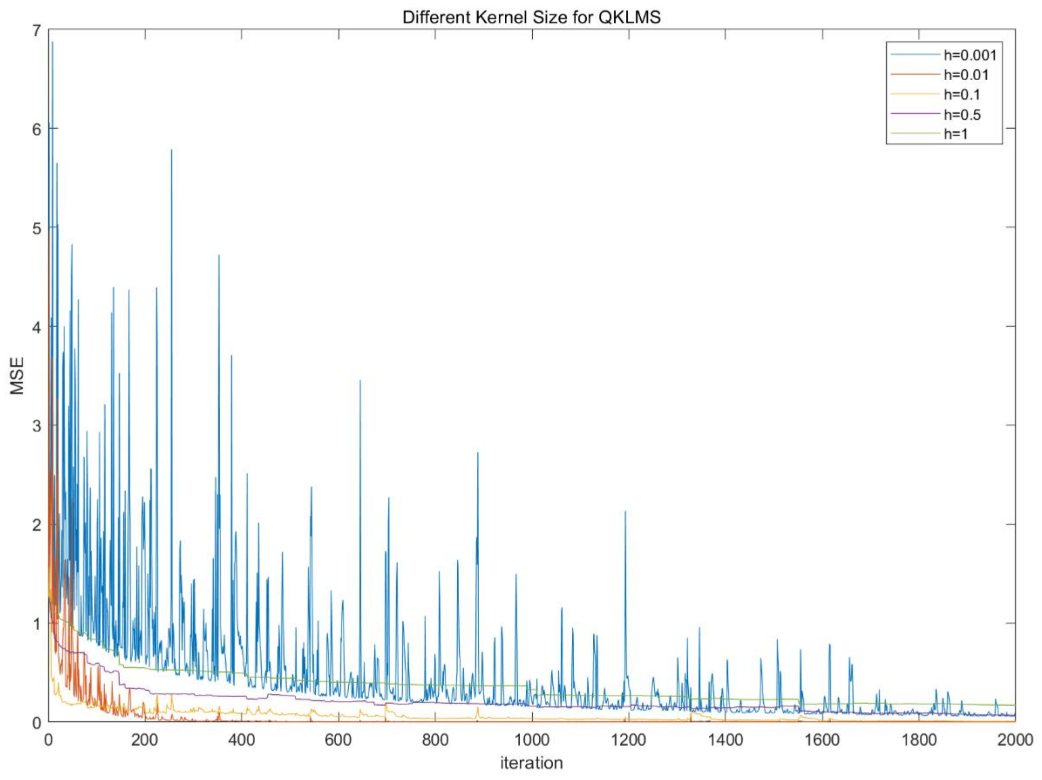

Figure 4. MSE of different kernel size for QKLMS

Figure 2. shows performances of LMS, KLMS and QKLMS. Here I trained 2000 points and chose 50 points as test dataset. We can find KLMS has the smallest MSE and it converges fast. QKLMS has a similar performance with KLMS. And LMS performs worst.

Figure 3&4. show different kernel size for KLMS and QKLMS. We can find when kernel size equal to 0.1 or 0.01, we get best result. Here I chose 0.1 because its MSE is almost same with that of 0.01 but its curve is smoother and it converges faster.

iii.

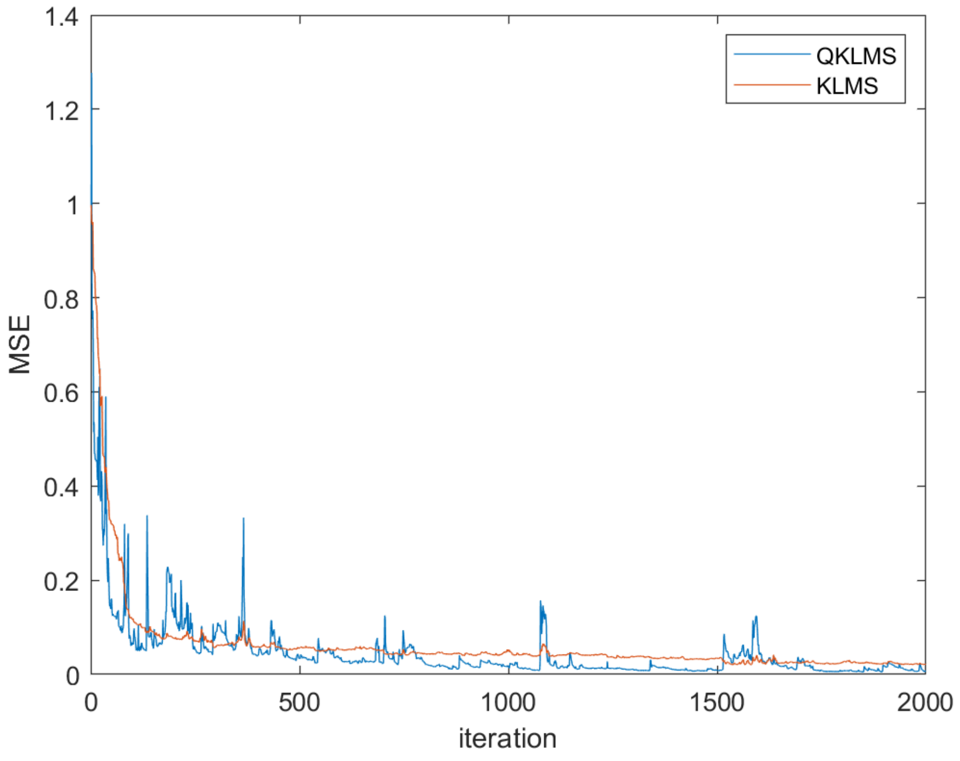

Figure 5. MSE for KLMS and QKLMS

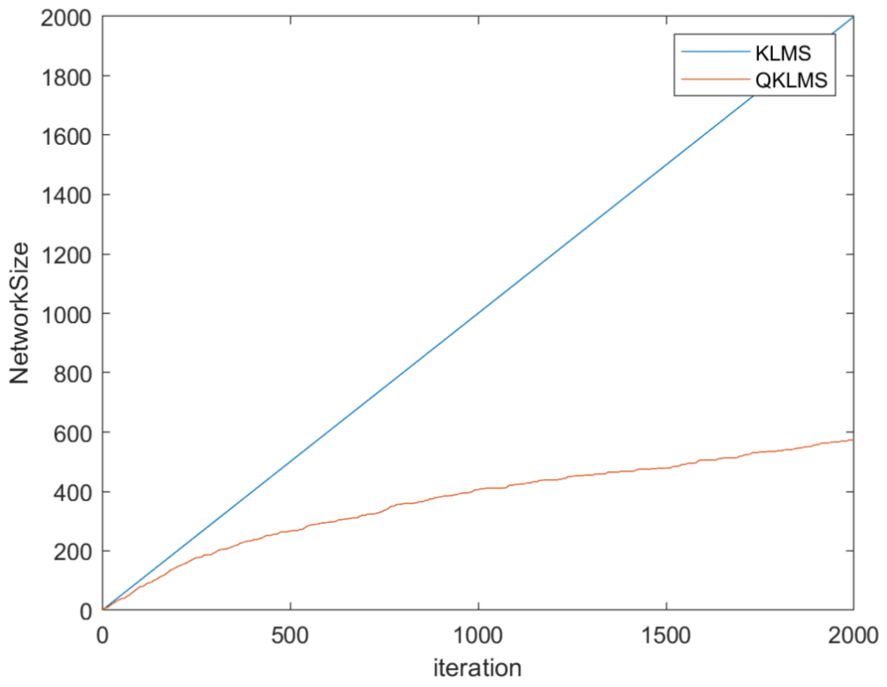

Figure 6. Network size for KLMS and QKLMS

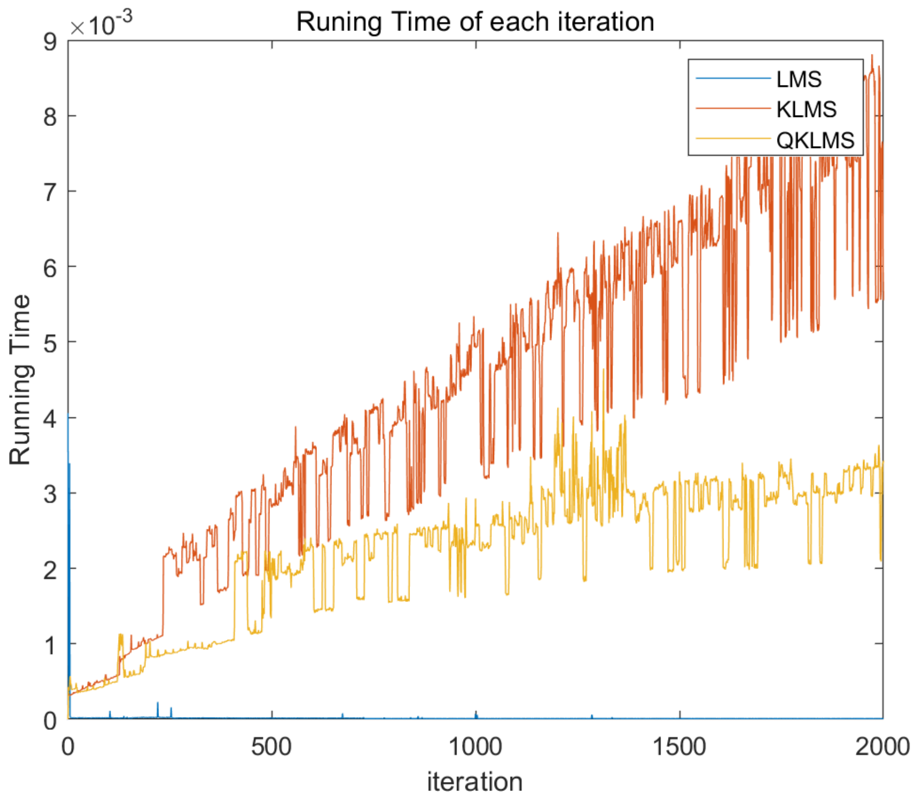

Figure 7. Running time for each iteration

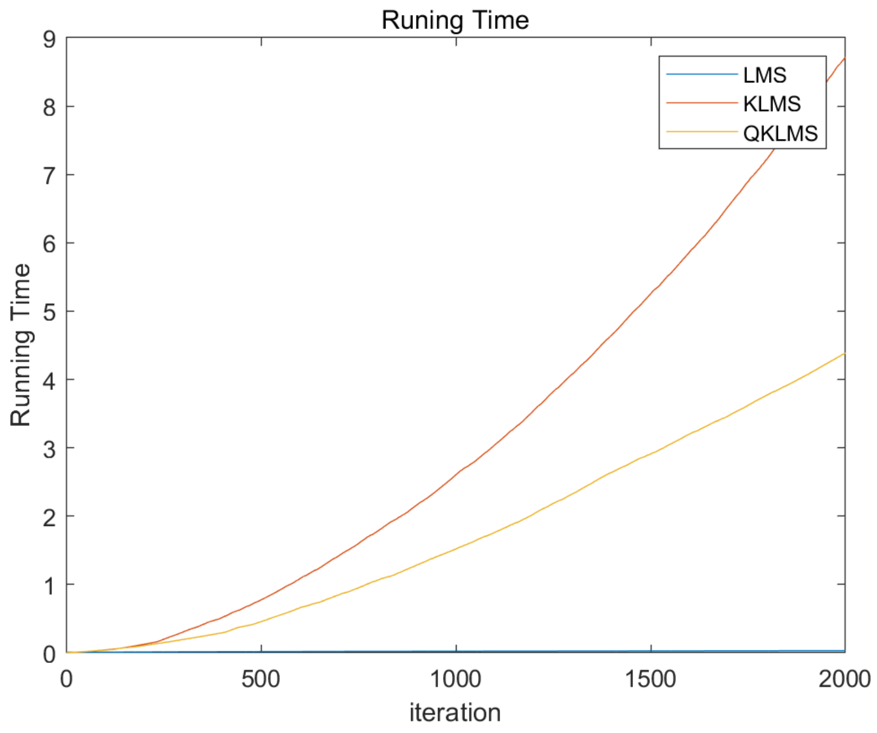

Figure 8. Running time for KLMS and QKLMS

As is discussed in question 2, from Figure 2, we can find KLMS has the smallest MSE and it converges fast. QKLMS has a similar performance with KLMS. And LMS performs worst. Therefore, linear filter (LMS) performs worse and non linear filter (KLMS and QKLMS) performs better.

Figure 6. shows network size of KLMS and QKLMS. QKLMS owns only half network size. Also we can make it smaller by setting a large threshold value. Figure 7&8 shows results of running time. LMS runs fastest. By using QKLMS, we can save time dramatically, which is really useful for some large iteration or large computation missions. If we need it run faster, we can set a larger threshold value at the cost of a larger MSE. But this method is helpful when utilized in real world. We can trade off to get ideal result.