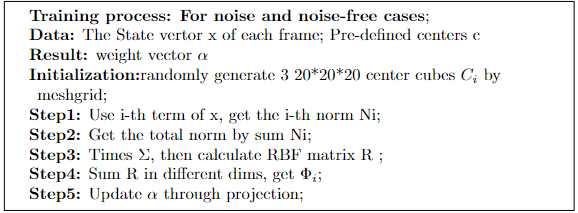
**Problem 4**

My teammate is Yinzhe Zhang

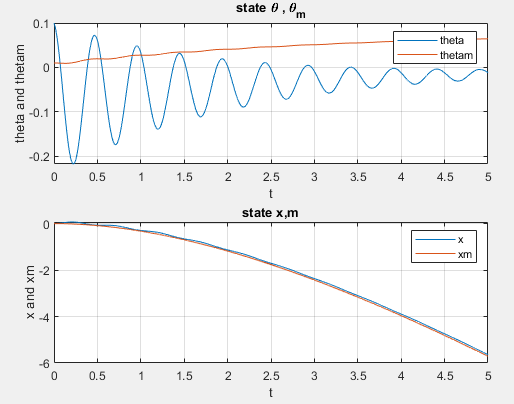
For training process, all I need to do is to generate some centers, then calculate RBF, sum them and update. For test process just use the alpha for prediction. And our group’s pseudo code is as follow:

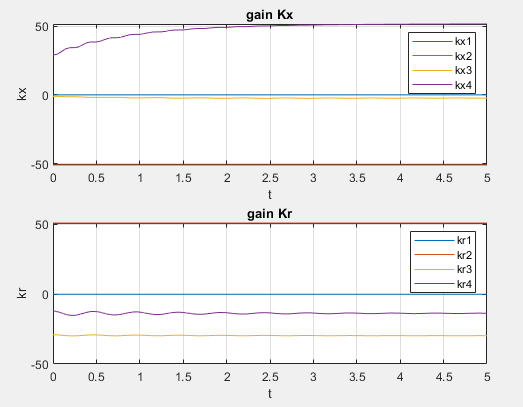


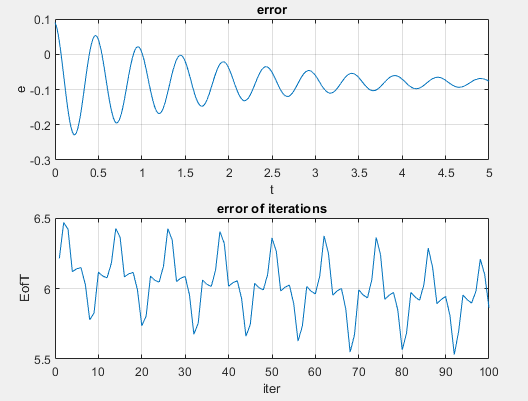
More details can be found in our code.

**(a)These are our test for noise-free + stable initial condition cases:**

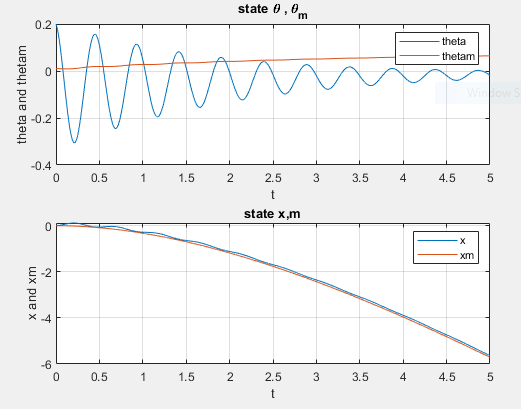
**<1>tspan=[0,5], State0 = [0;0.1;0;0]**

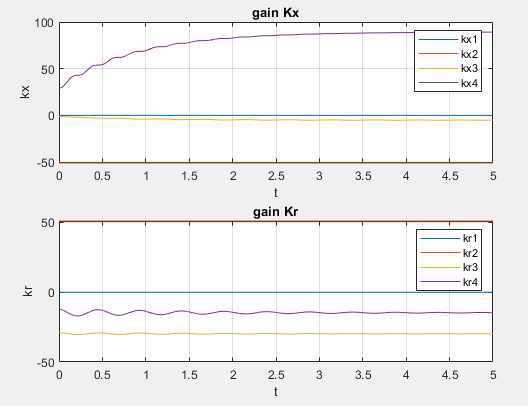


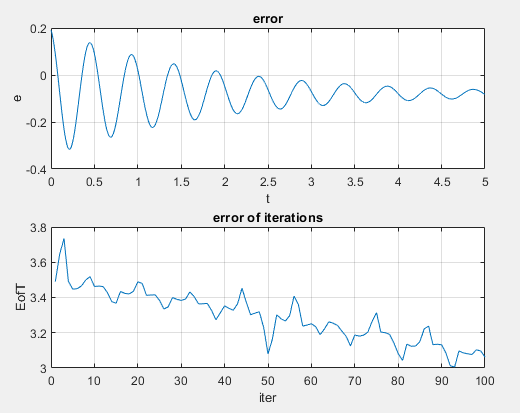




**<2>tspan=[0,5], State0 = [0;0.2;0;0]**



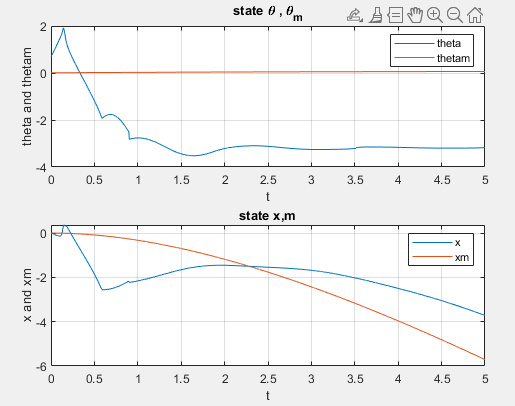


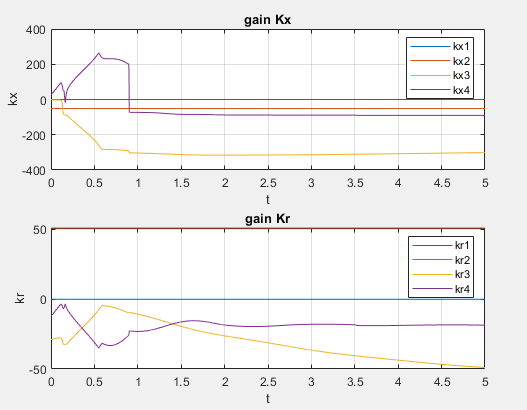


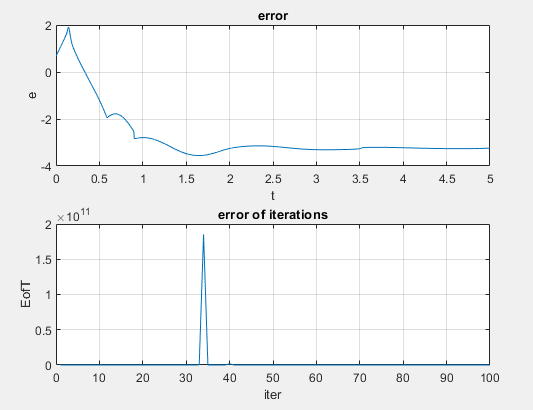
**And for initial case till theta0 = 0.5, the system is good.**

**(b)These are our test for noise-free + unstable initial condition cases:**

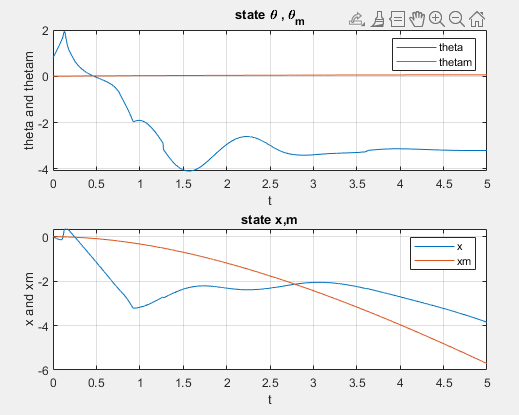
**<1>tspan=[0,10], State0 = [0;0.7;0;0]**

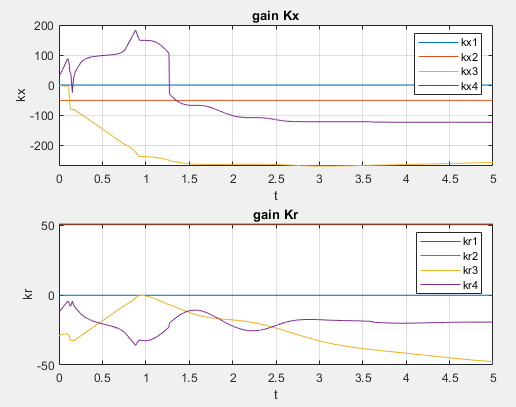


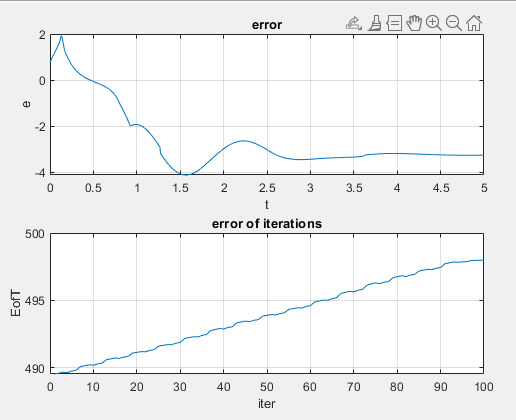




**<2>tspan=[0,5], State0 = [0;0.8;0;0]**

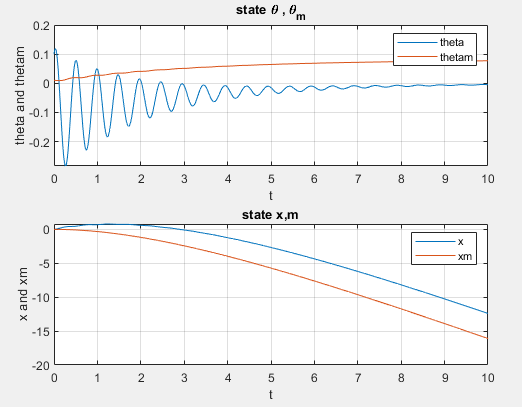


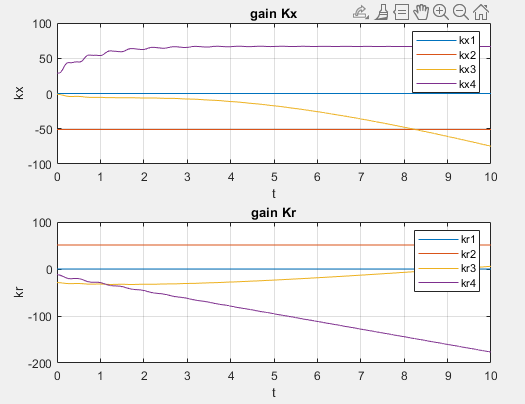


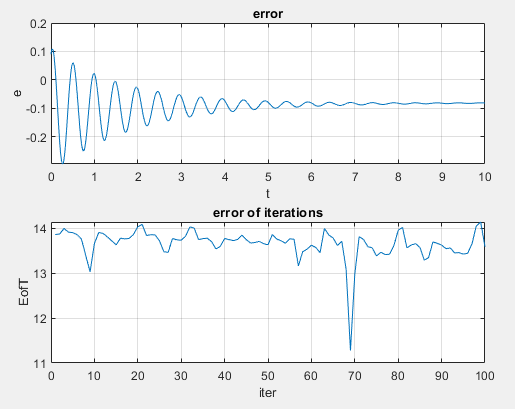


**(c)These are our test for noise + stable initial condition cases:**

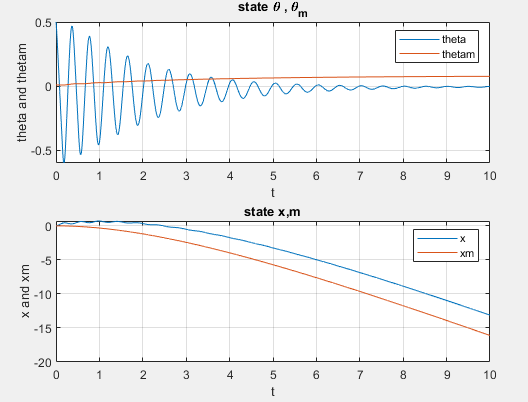
**<1>tspan=[0,10], State0 = [0;0.1;0;0]**

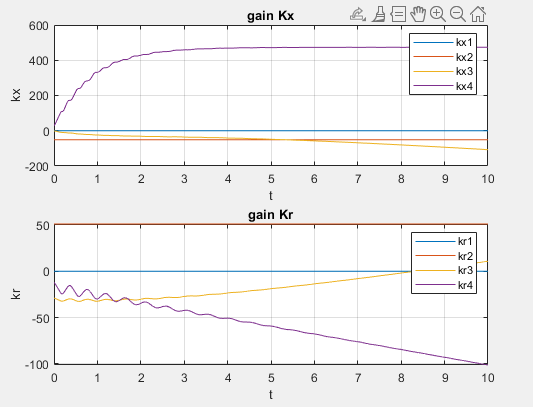


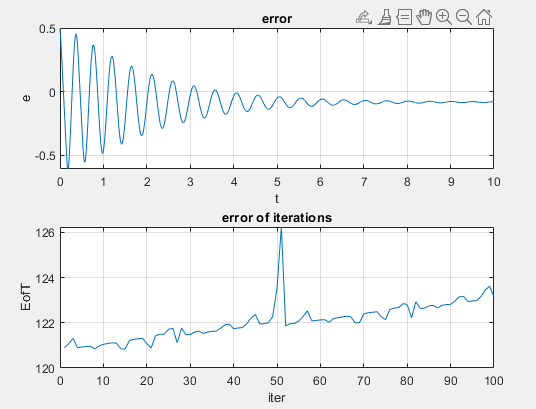




**<2>tspan=[0,10], State0 = [0;0.5;0;0]**

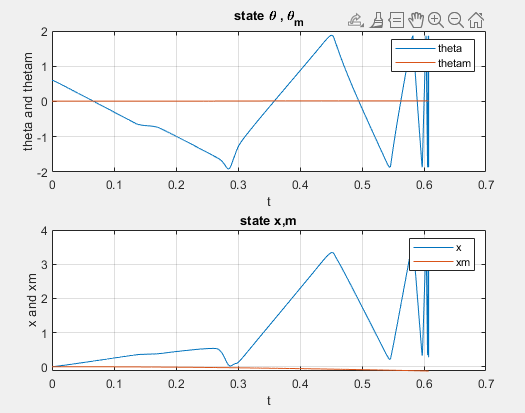


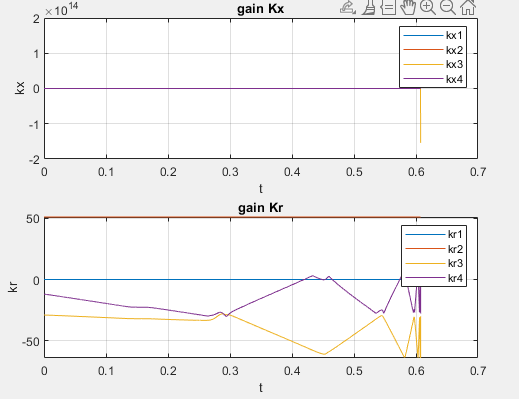


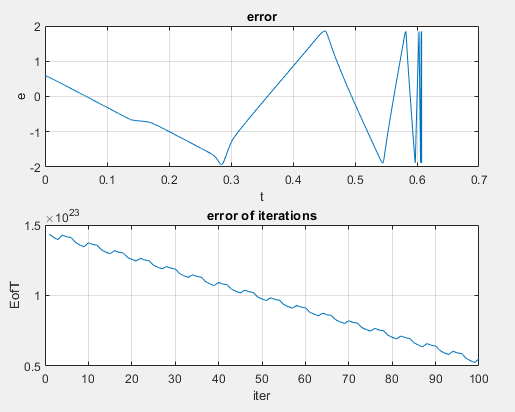


**(d)These are our test for noise + unstable initial condition cases:**

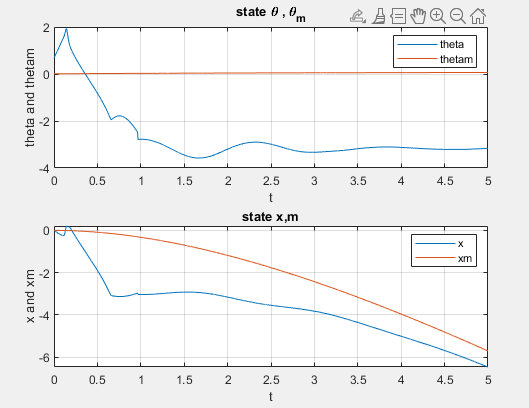
**<1>tspan=[0,5], State0 = [0;0.6;0;0]**

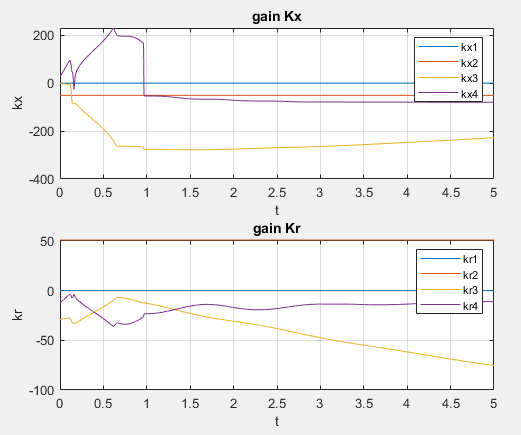


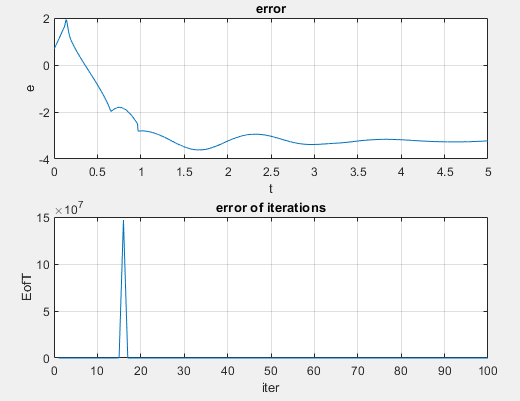




**<2>tspan=[0,5], State0 = [0;0.7;0;0]**

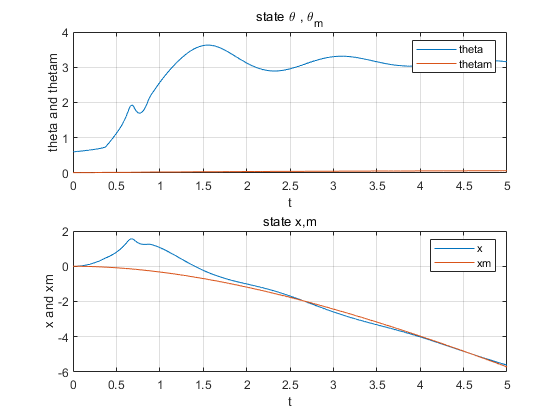


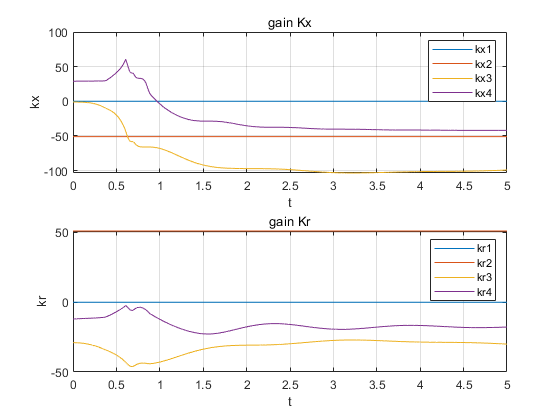


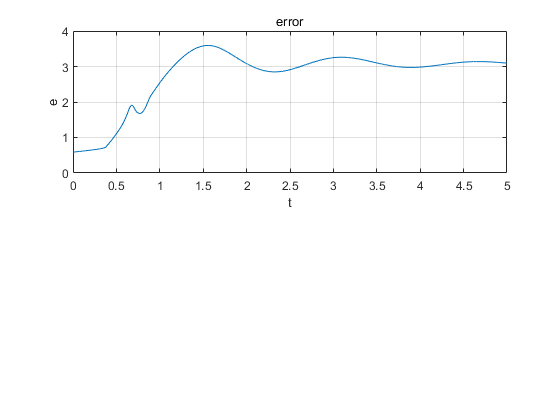


**(e)These are our test for noise free+trained-alpha + unstable initial condition:**

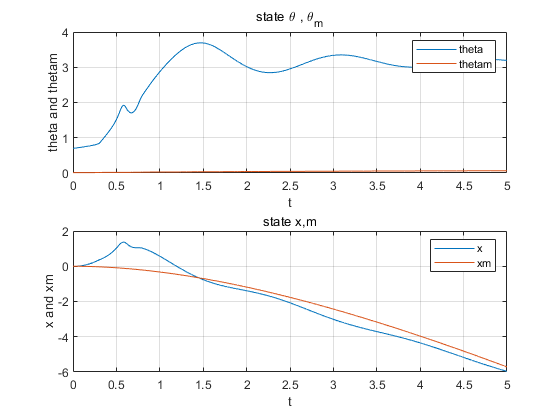
**<1>tspan=[0,5], State0 = [0;0.6;0;0]**

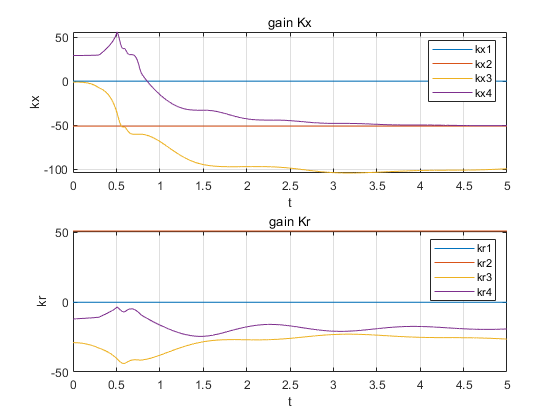


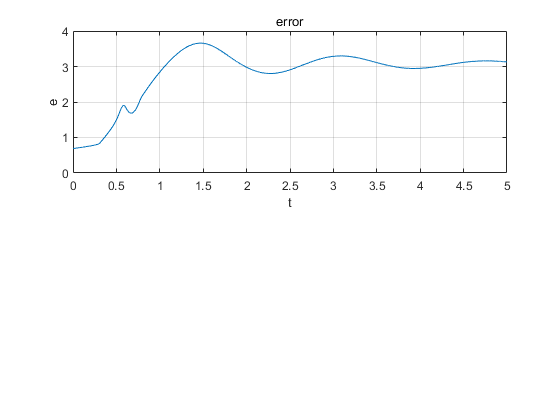




**<2>tspan=[0,5], State0 = [0;0.7;0;0]**

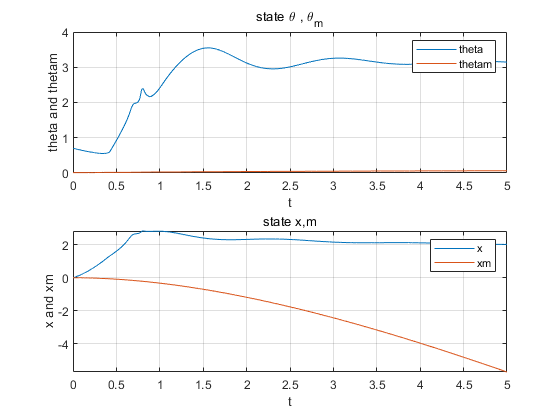


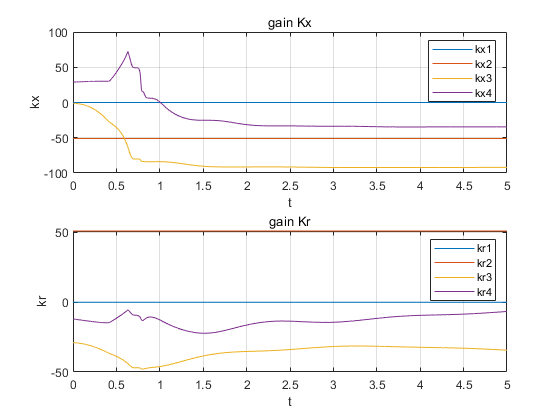


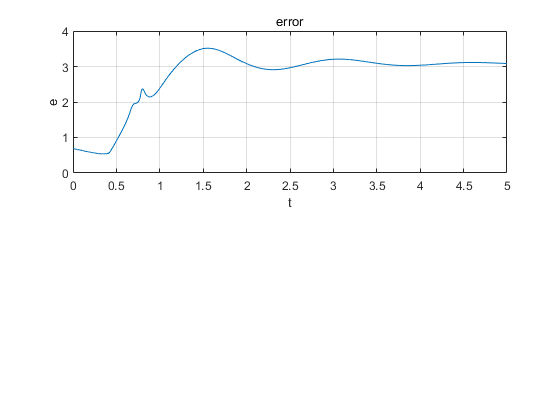


**(f)These are our test for noise+trained-alpha + unstable initial condition:**

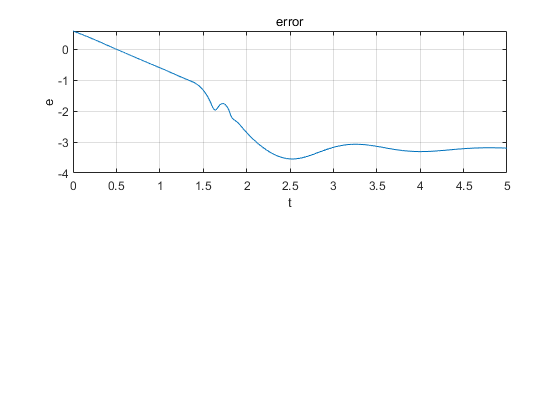
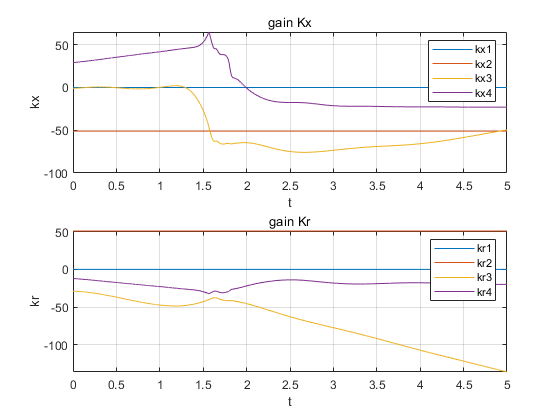
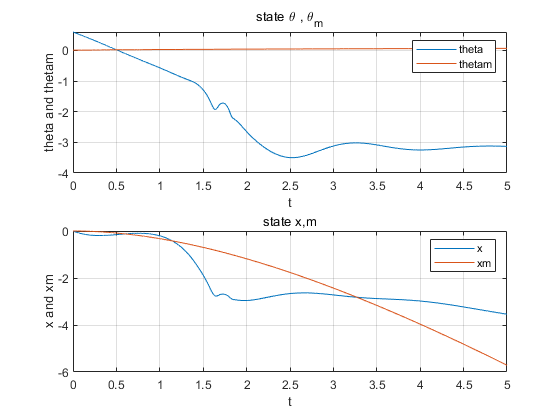
**<1>tspan=[0,5], State0 = [0;0.6;0;0]**







**<2>tspan=[0,5], State0 = [0;0.7;0;0]**



**(g)Analysis and Conclusion**

From the experimental results of a-f above, it is clear that the NN adaptive control system we designed accomplishes the expected function. During the training process, the system can make the pendulum return to 0 at input theta 0.1-0.5, but it becomes unstable at 0.6 and above. In addition, the system is resistant to noisy inputs and remains basically stable at the origin. However, it is worth noting that when we experiment with those unstable initial values using trained parameters, the systems do maintain stability around some fixed angle although they cannot return to the origin. We conjecture that there are two main reasons for the inability to return to the origin: one is that our system has fewer neurons and the performance of the RBF network needs to be improved. Second, our system may still not have learned the optimal parameters. This may be due to our short training time (our computation is time-consuming, so one simulation does not last long.) And the system can stay at a fixed value because the alpha parameter we trained can give the system some stability