Homework 1

NAME: Tianrong Chen **GTID: 903426170**

Problem	Points
Problem 1	
Problem 2	
Problem 3	
Problem 4	
Problem 5	
Problem 6	
Problem 7	
Problem 8	
Total	

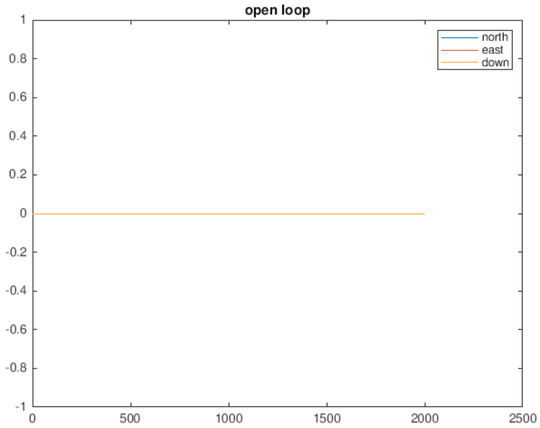
Helicopter control problem 1

a:
Values of trim:
Aileron trim:0.0065
elevator trim:0.0016
rudder trim:0.0094
collective trim:0.4001

Problem 2

b:

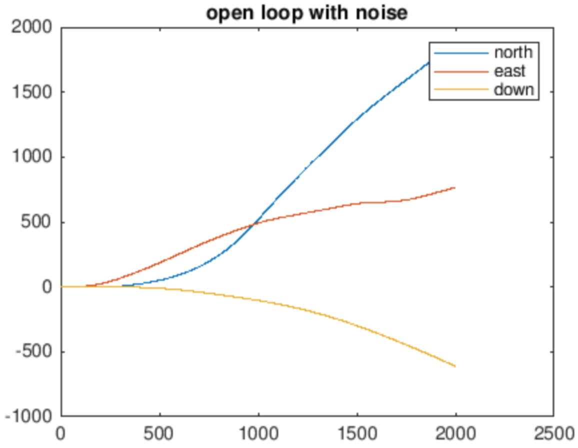
The plot of North,East, Down over time is following:



Problem 3

c:

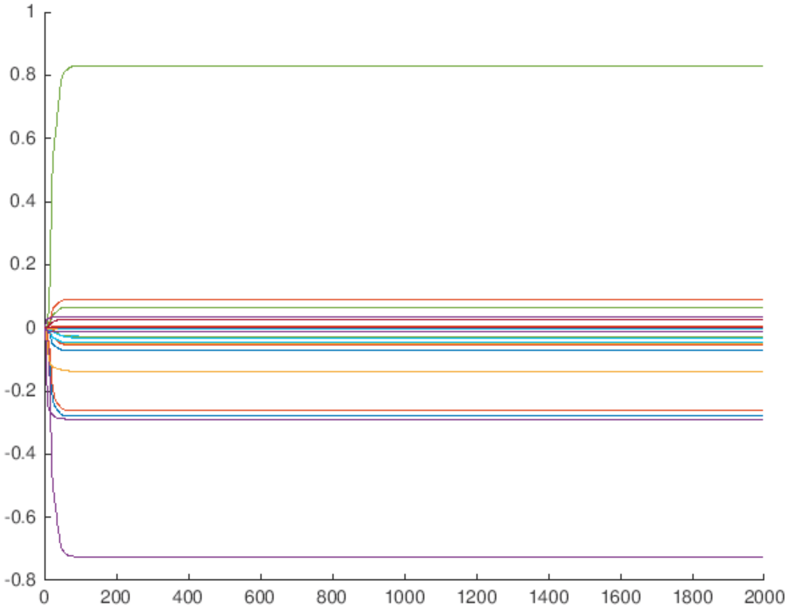
The plot of North, East, Down over time with Noise is following:



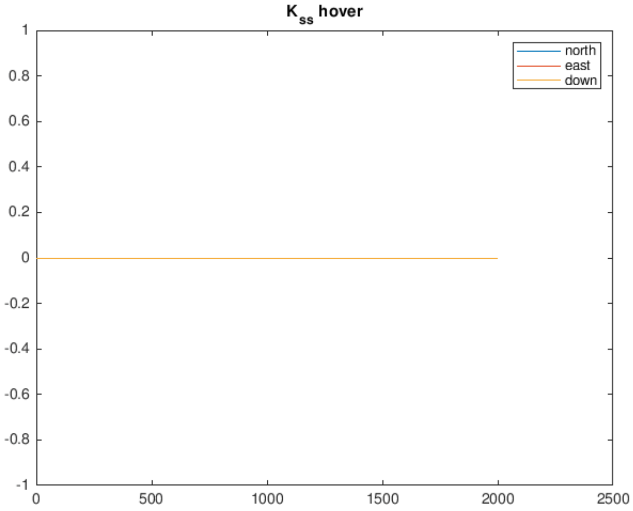
Since it is a open loop system, it does not have the ability to recover from the fault situation. Precisely, while noise is introduced, the state of the helicopter would be influenced, however, the action u keeps unchanged, which results in the deviation of North east down state.

Problem 4

d i:
The plot of entries of K is following:

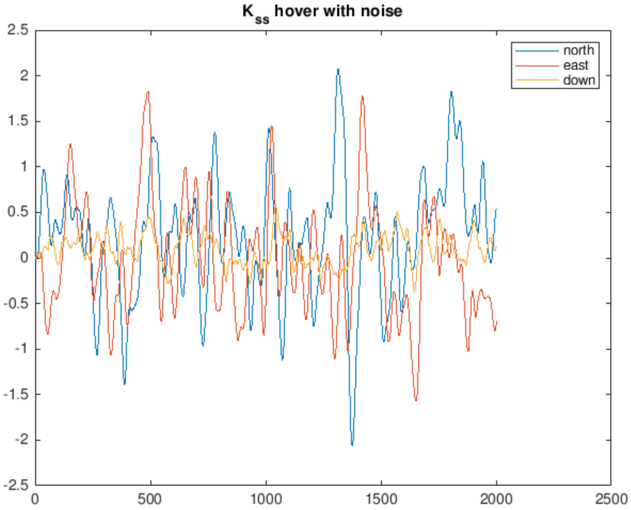


ii:
NED plot without noise is following:



iii:

NED plot with noise is following:



Problem 5

e: i:

The table for the parameters which lead to unstable system is following:

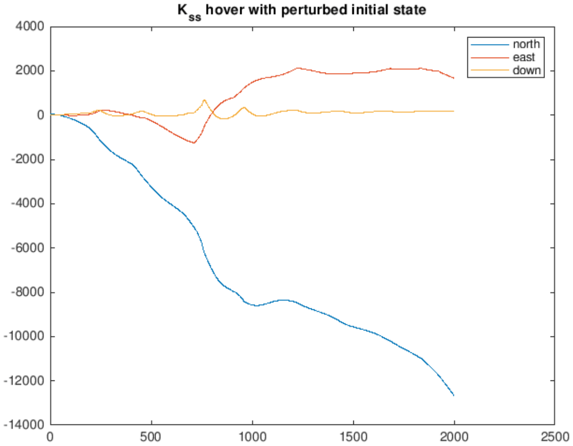
North	East	South
30	20	60

For Axis:

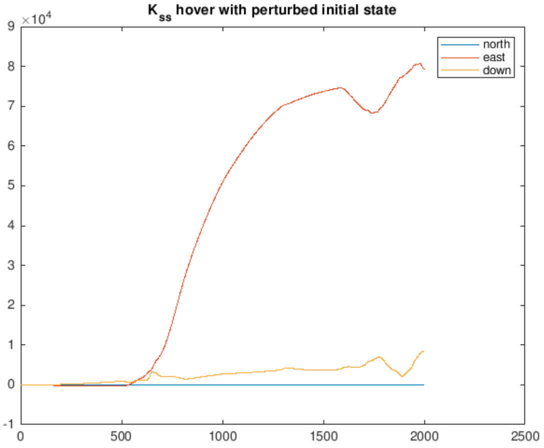
X-axis	Y-axis	Z-axis
$11/16\pi$	$5/8\pi$	$11/8pi$

The plot of state is following:

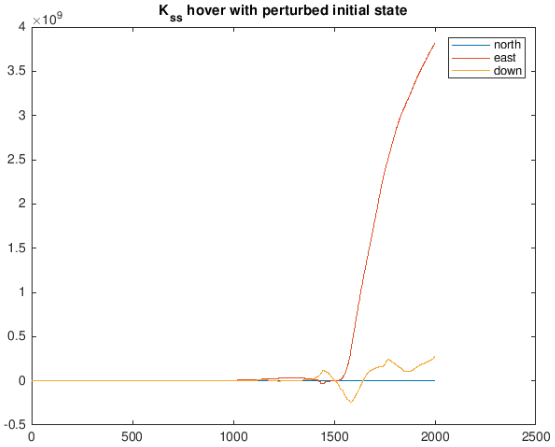
The smallest value for north that makes the controller perform poorly is 30, the plot of state whose initial ned parameter is:[30,0,0] is following:



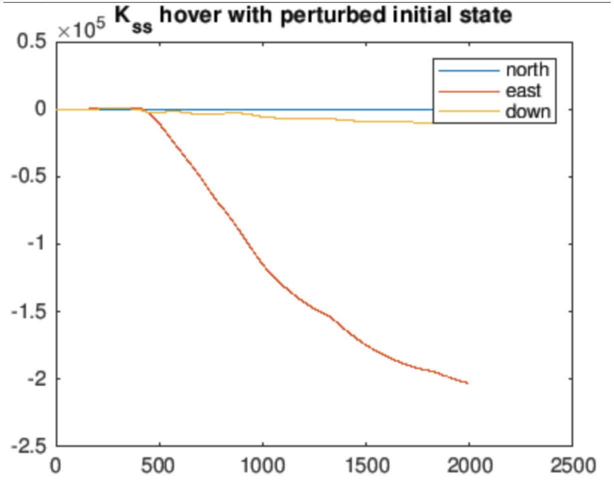
The smallest value for east that makes the controller perform poorly is 20, the plot of state whose initial ned parameter is:[0,20,0] is following:



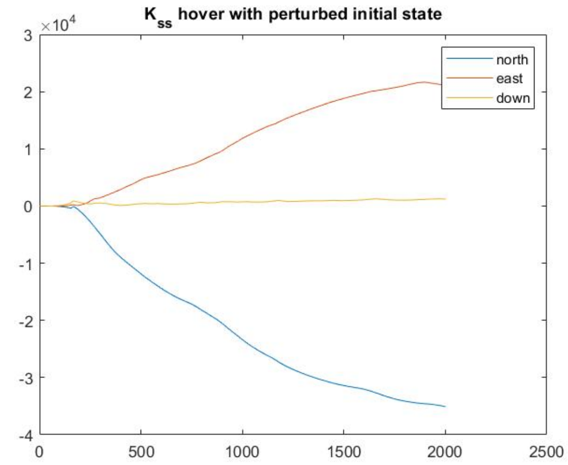
The smallest value for down that makes the controller perform poorly is 60, the plot of state whose initial ned parameter is:[0,0,60] is following:



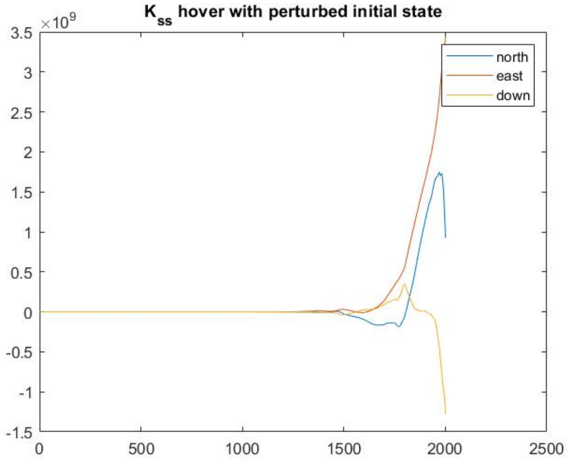
The smallest value for x-axis that makes the controller perform poorly is 11/16 π , the plot of state is following:



The smallest value for y-axis that makes the controller perform poorly is 5/8 π, the plot of state is following:

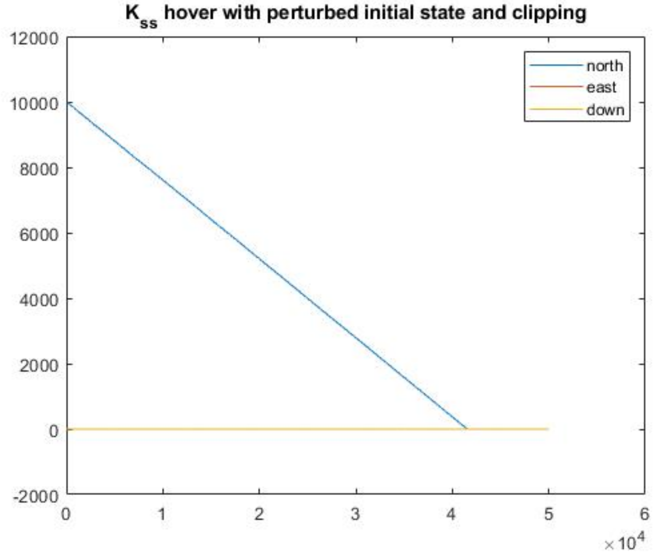


The smallest value for z-axis that makes the controller perform poorly is $11/8\pi$, the plot of state is following:

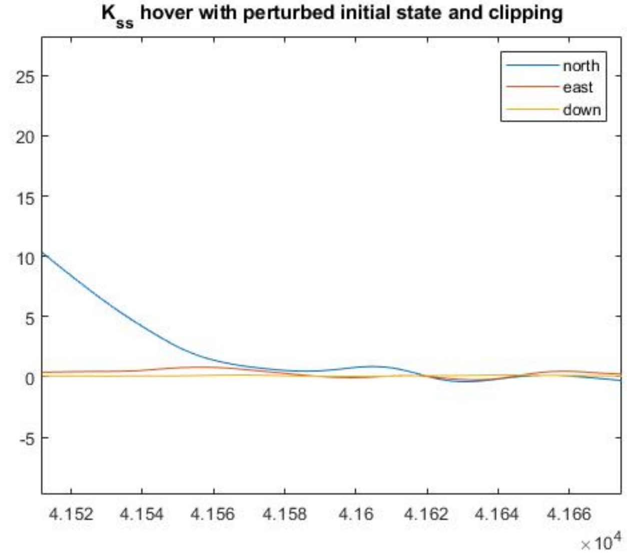


Problem 6

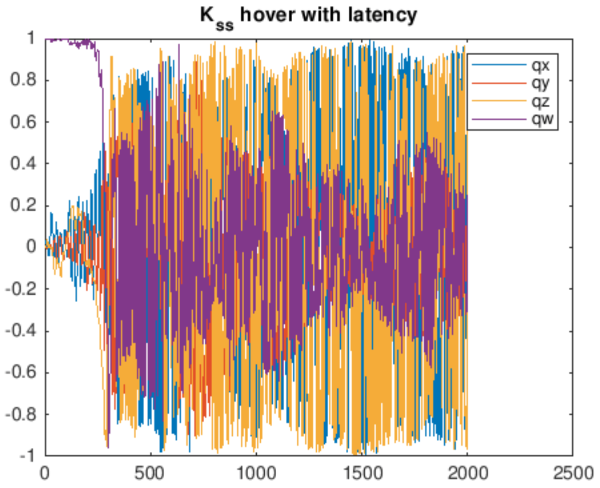
f1:
The clipping distance I pick is 10, meanwhile, the plot of NED is following:



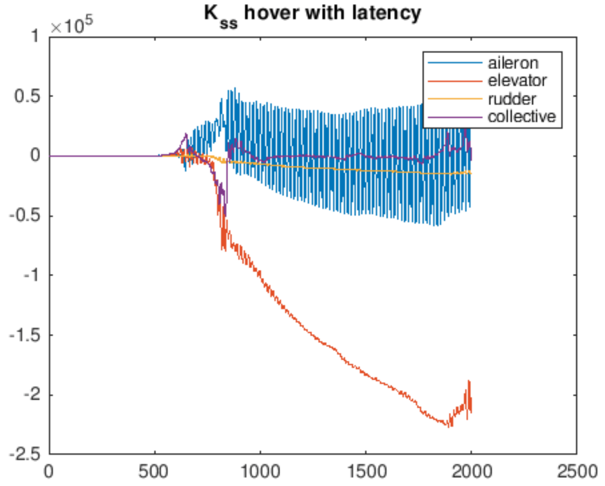
When we zoom in, we can find that the final state became stable around target hover state:



f2:
The minimal latency which lead to worse performance I picked is 3, the plot of q is following:



The plot of u_prev is following:



Problem 7

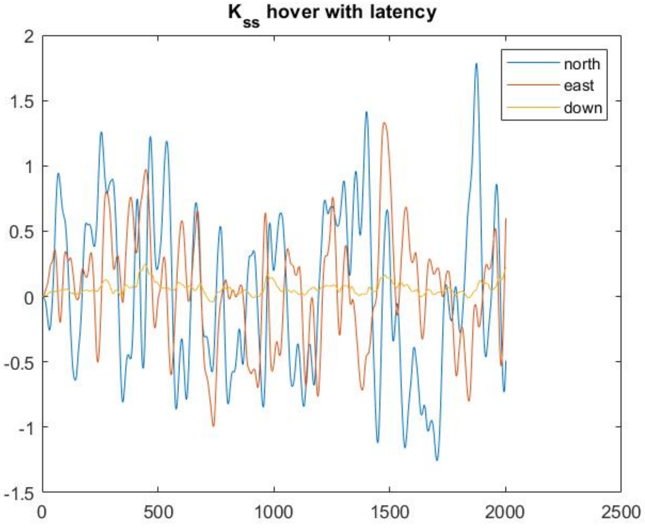
In this section, I tried two method: Neural Network and Fminsearch. Fminsearch works but Neural Network fails.

Description of first method: Fminsearch is a optimizer which is used to find the optimal result to minimum a specific function. In this case, the cost function I used is:

$$L = \int_0^H x^T Qx + uRu + ||x' - x^*||_2 dx$$

where x' represents for next state, x^* represents for target hover state. The start point we used to search is K_{LQR} which is generated from LQR.

The result of Fminsearch: The plot of NED after applying this method is following:



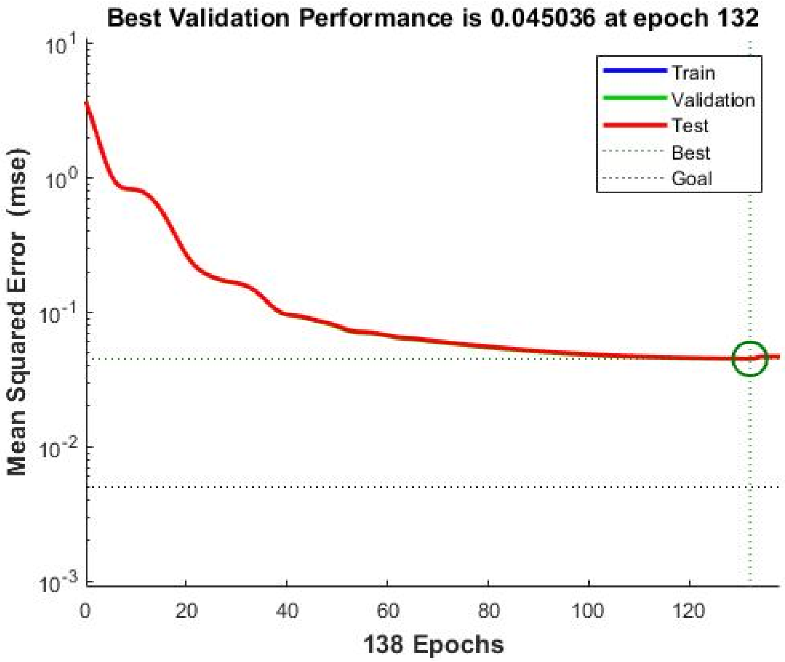
;

Analysis:

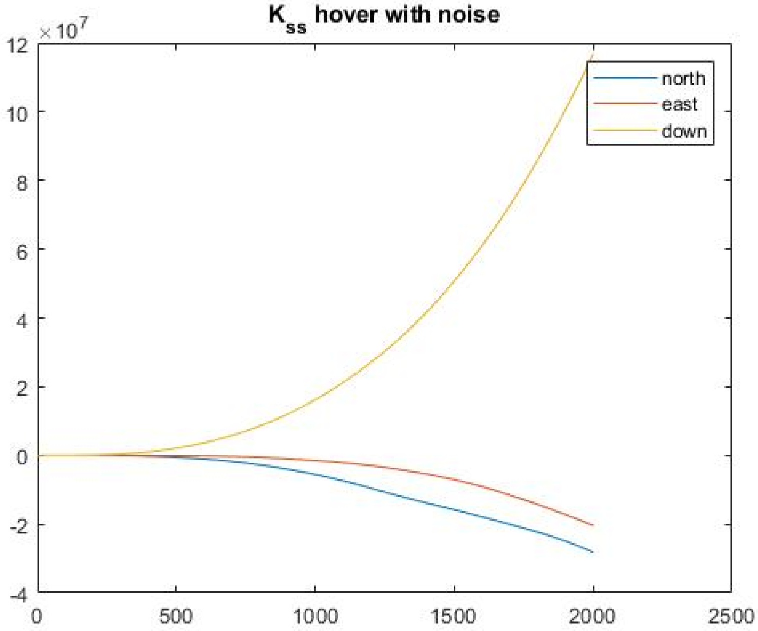
From the figure shows above, we can find that this method perform well under the situation with noise. Compared with LQR method, they have similar result, because both of them are using linear controller under non-linear environment.

In terms of Neural Network method: I thought it could success, if the data set is plentiful, to map from state to action directly, but it does not. First I run the model by using LQR for 100 times, the time horizon for each time is 2500. Totally we can get 250000 data set. The label is LQR's control output. Then I normalized the data because NED sometimes would get extremely large in some case compared with other features which would effect the performance of neural network. Since the Next state have strong relationship with current state, I randomly choose 200000 data from data set to make the input data iid.

After training the neural network, the cost function is less than 0.1 as following:



However, when we run this neural network module back to the real non-linear system, it does not work:



;

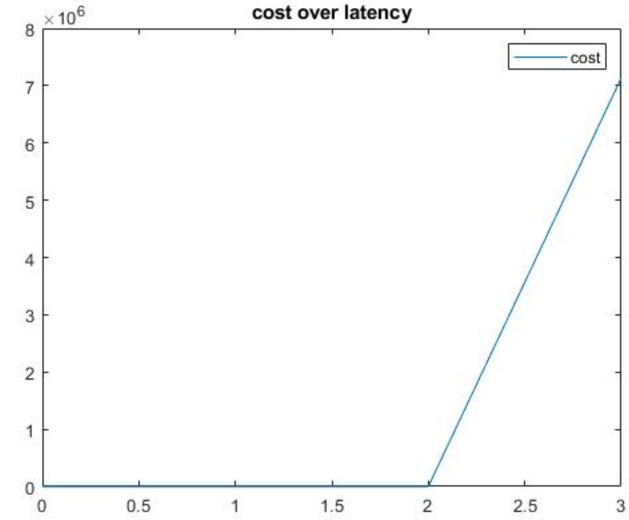
Analysis:
1. Problem of neural network: the label is not the true label. The label we can get is LQR's output. However, it is not the true label. The delimma is, if we do not use LQR's

output, we can not get the true label instead: it is impossible to invite a experienced pilot to label all data set. Correspondingly, in this case, I think we could use Imitation learning like DAgger. However, in DAgger, we still need to label some of the data set even though number of data set needed to be label would hopefully be less than the whole data set.

(digression stuff: When I was trying the Actor-critic method, the problem occured when I was training the policy network. The policy network I used composes of two networks. One of them is the network to produce the σ, and another one produces the μ. Then we can choose the action under the Gaussian distribution parametized by σ and μ produced by former network. However, I ran out of time because I cannot find an efficient way to train this network in Matlab.)

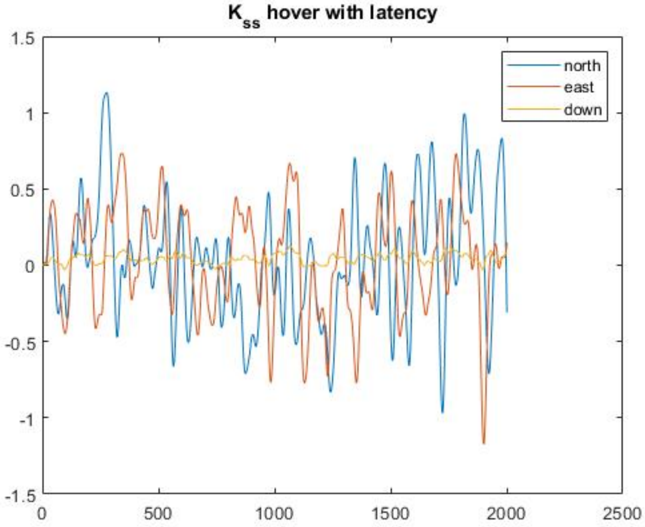
Problem 8

The plot of cost over Latency:



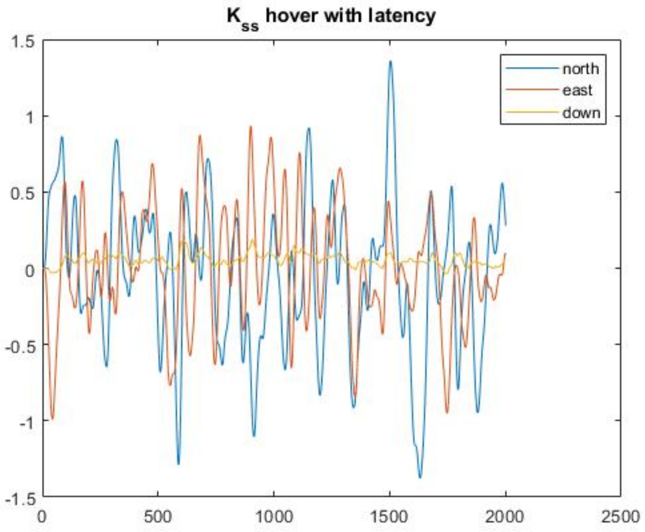
;

The plot of NED when latency is 0:



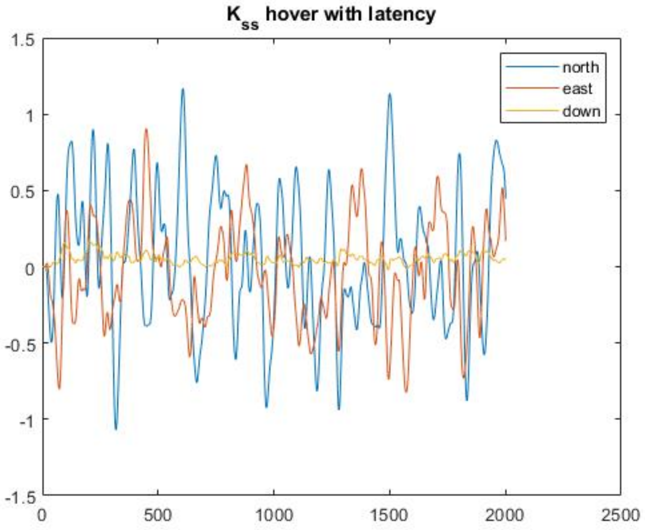
;

The plot of NED when latency is 1:



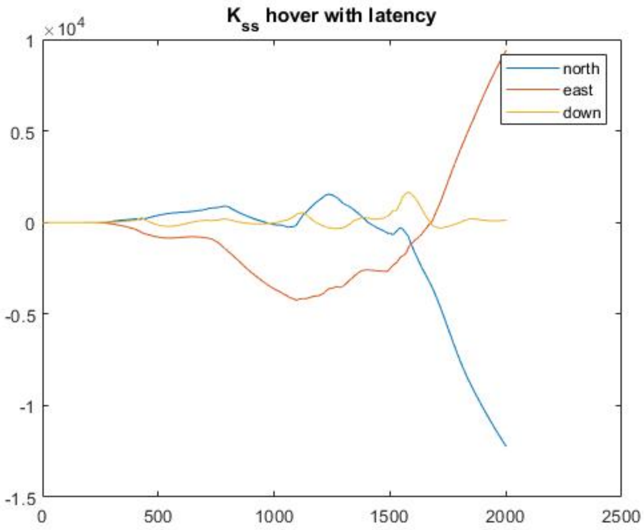
;

The plot of NED when latency is 2:



;

The plot of NED when latency is 3:



;

From the figures above we can find that the fminsearch method works well when latency is smaller than 3.