**ME424 Project 2 Report**

**11912404 江轶豪**

**11912237 姚雨君**

1. **Full state space model:**



where 



1. **Angle dynamic model:**



1. **Linearized model for angle dynamics:**

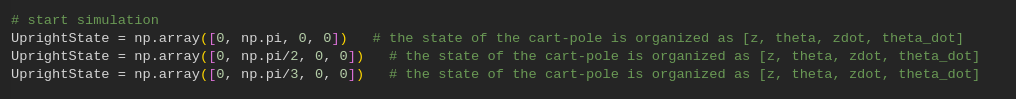
, where

1. **Discrete Time Angel Dynamics Model:**

****

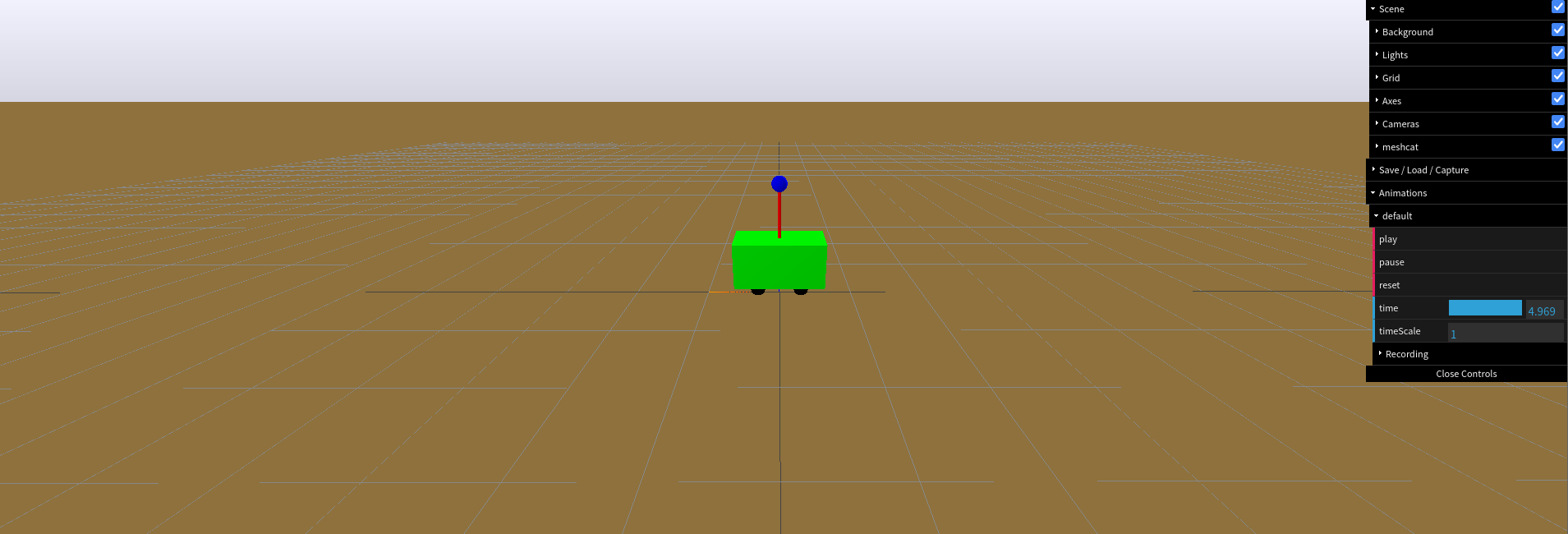
1. **Drake Simulation Setup and Testing**

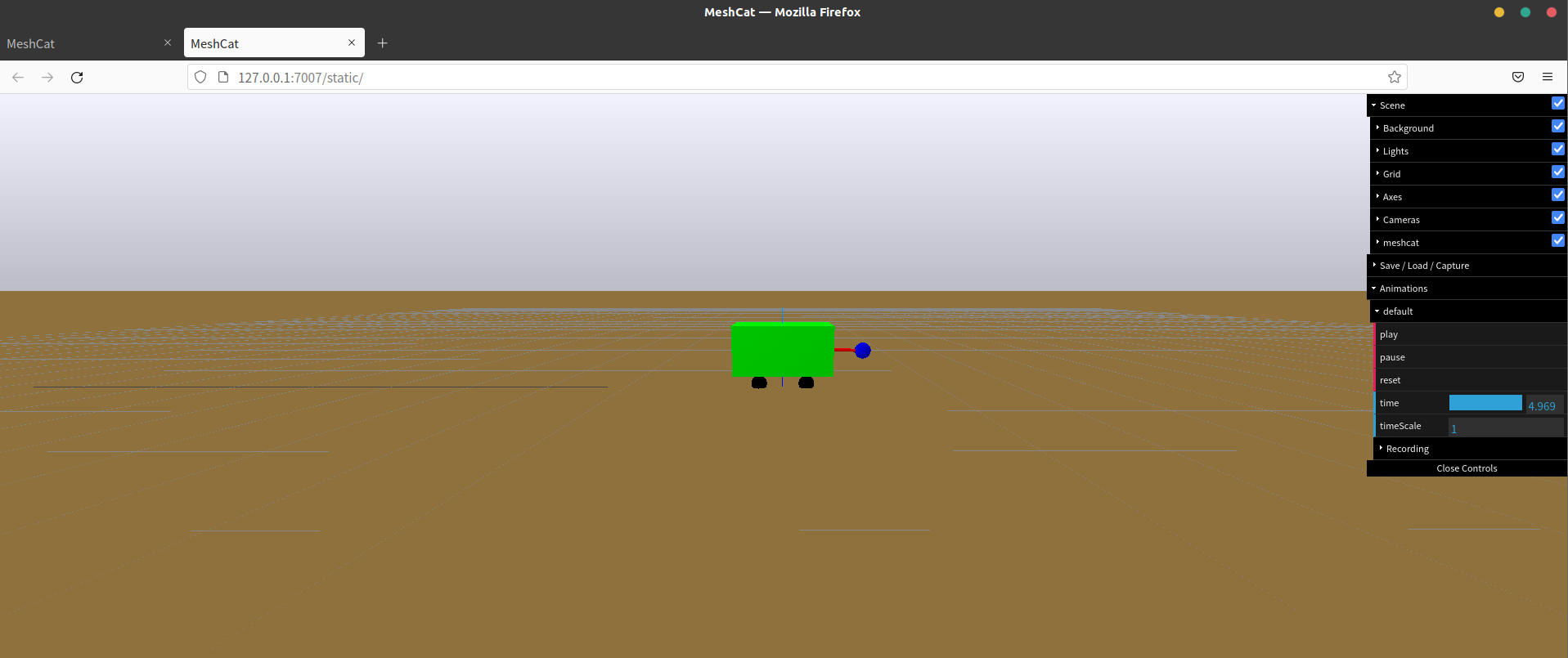
Initial state vectors are shown as below:

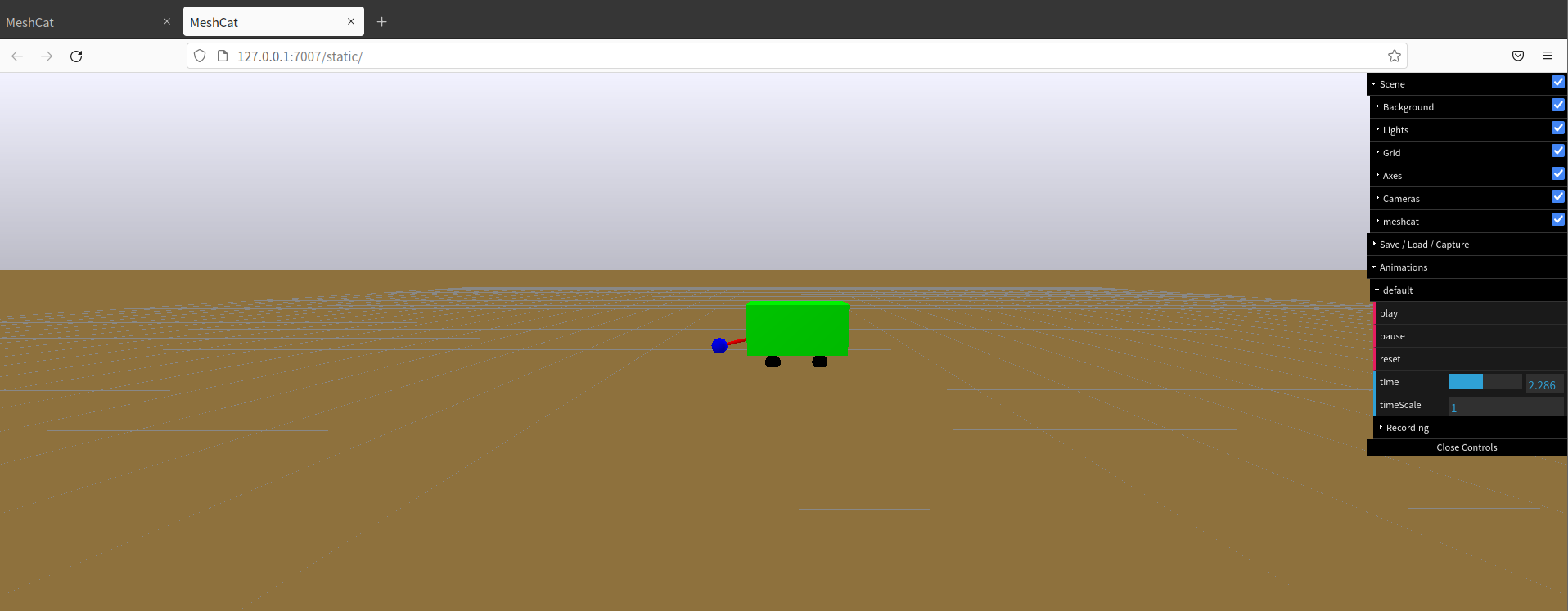


All snapshots shown as below are the simulations with no disturbance.

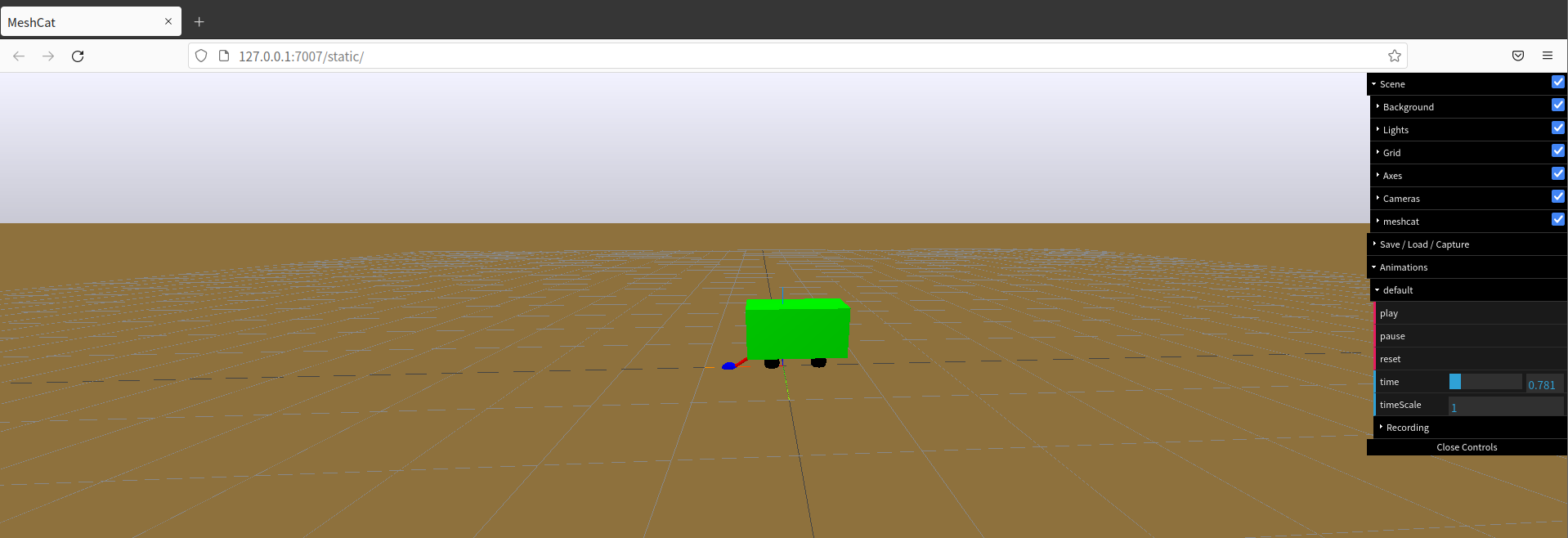


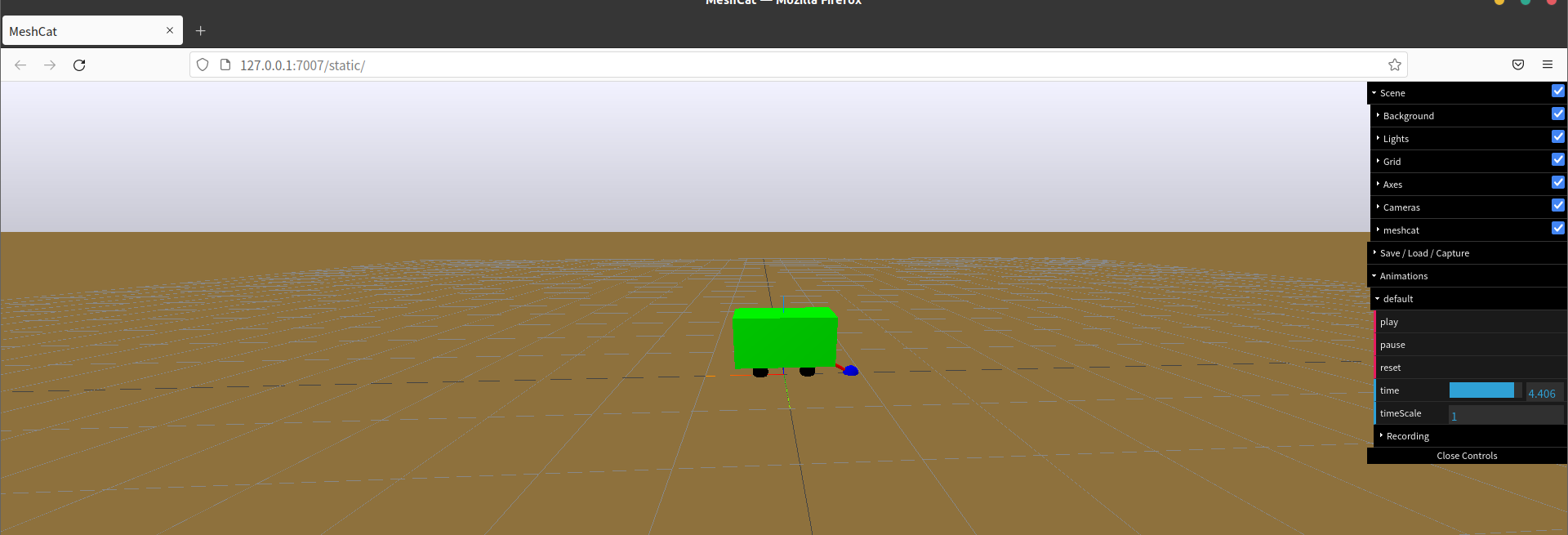
 (GIF file(q5\_0.5pi.gif) is attached in the document)





(GIF file(q5\_0.33pi.gif) is attached in the document)





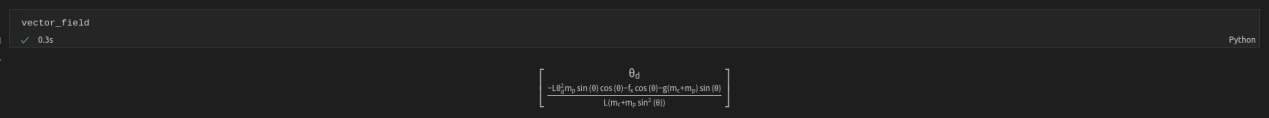
1. **Simulation studies for state-feedback control**
2. **Eigenvalue assignment**

To find K such that A-BK has eigenvalue , where , we should:

The first step:

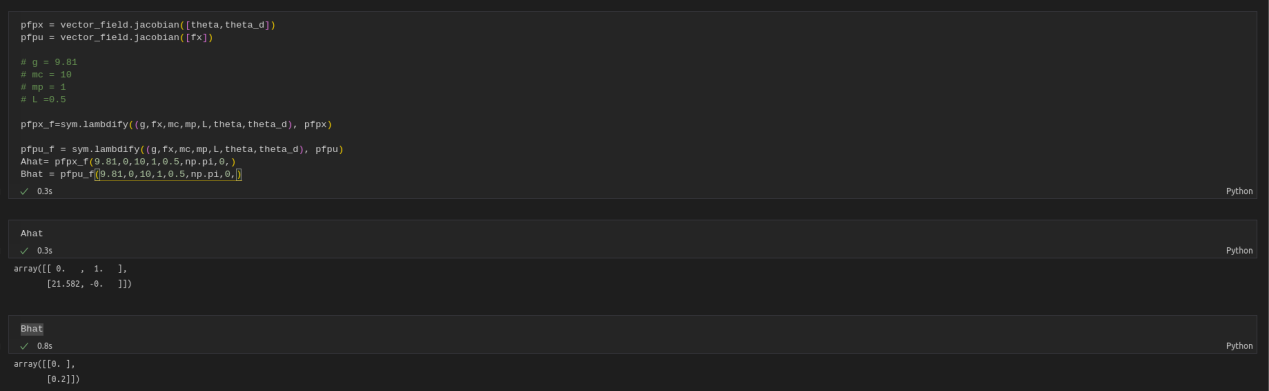
Convert function to a python code form:

The matrix shown as below is



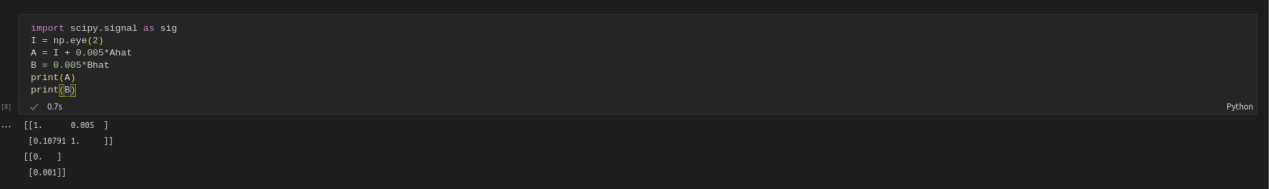
The second step:

Calculate Ahat and Bhat matrix based on Part 3 using python.



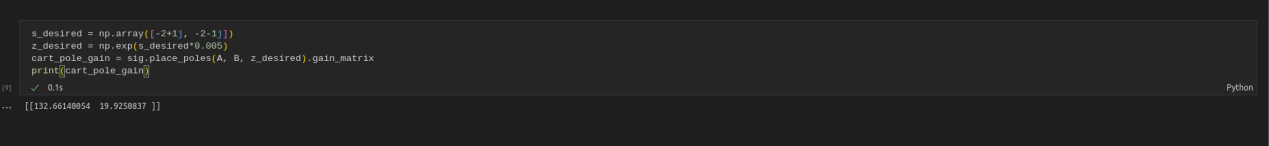
The third step:

Compute matrix A and matrix B for the discrete time angel dynamics model in Part 4 based on the continuous-time linearized model in Part 3. The result is shown below.



The fourth step:

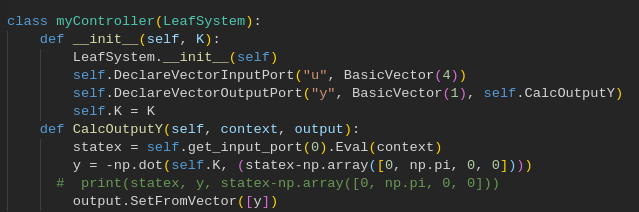
Apply the function *sig.place\_poles(matrix A, matrix B, z\_desired).gain\_matrix* to find gain K, which leads to the desired eigenvalues , where .



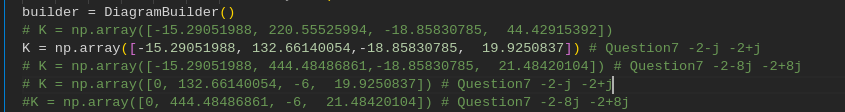
K=[132.66140054 19.9250837]

1. **Closed-loop simulation**

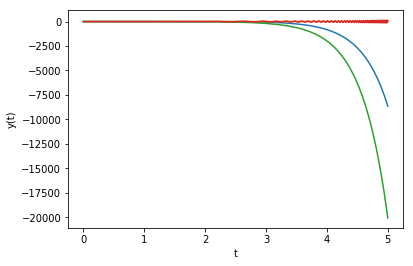
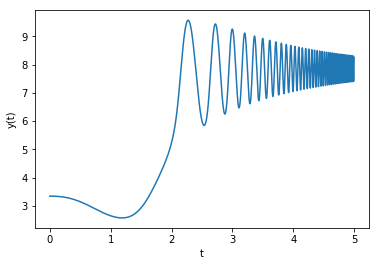
After calculating K, we should test the control is feasible or not, or the efficiency of adjustment to stable. In this part, we take the initial condition of [0, 0.2, 0, 0]T.



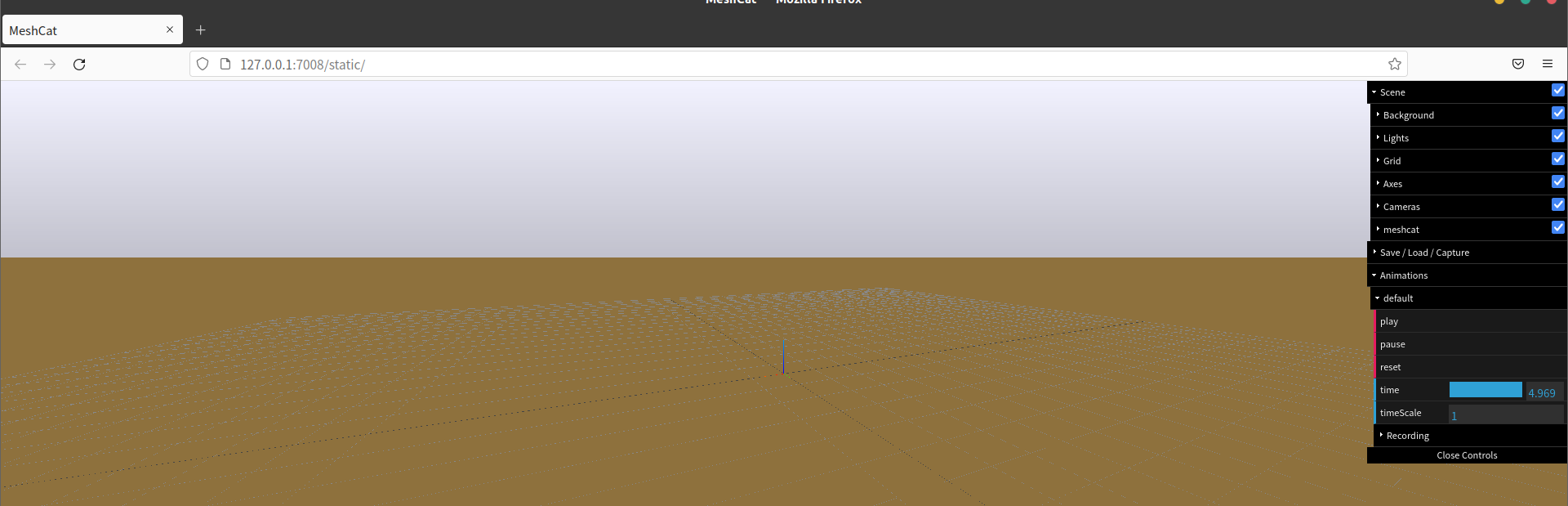
K2 and K4 have been decided in Part 6(a), which is 132.66140054 and 19.9250837. Since K1 and K3 is meant for z, and factor of z could make a difference to , so we prepared 2 sets of K with different K1 and K3.



The diagram is shown as below. The left one shows how changes with time. In the right figure, blue, yellow, green, red correspond to z, , , .

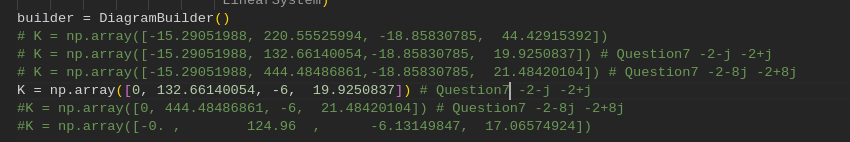


We cans see that converges to a value far more than and diverges.

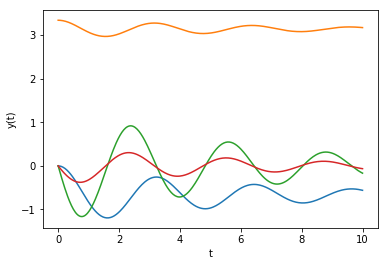
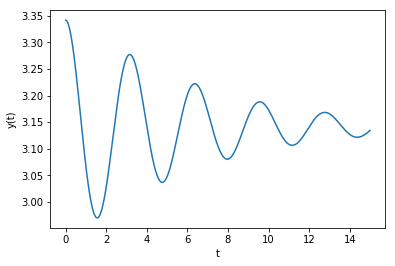


The cart is lost in the simulation, which indicates the failure (seen in q6b\_1.gif).

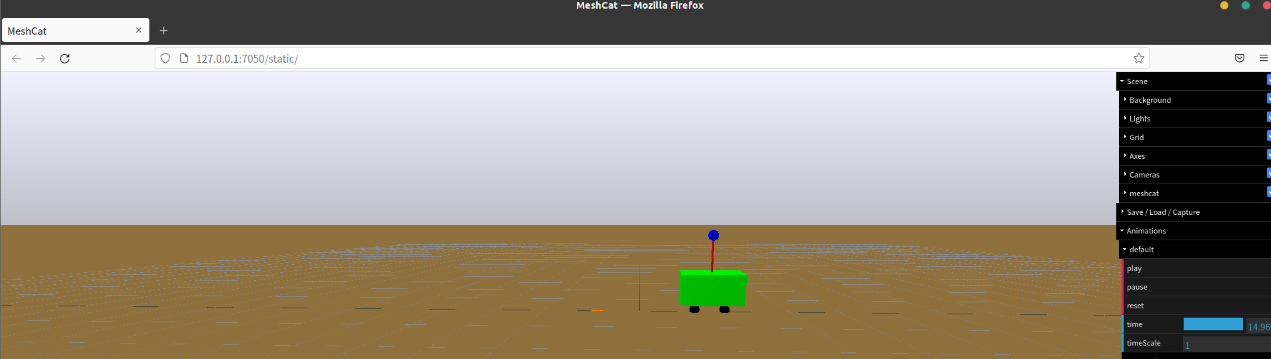
We can see that under such conditions, the cart pole is unstable. To eliminate the effect of other factor we used another set of K.



The diagram is shown as below:



We can see that and both converges, and converges to . Under this condition, the cart pole is stable, which is shown in q6b\_2.gif.

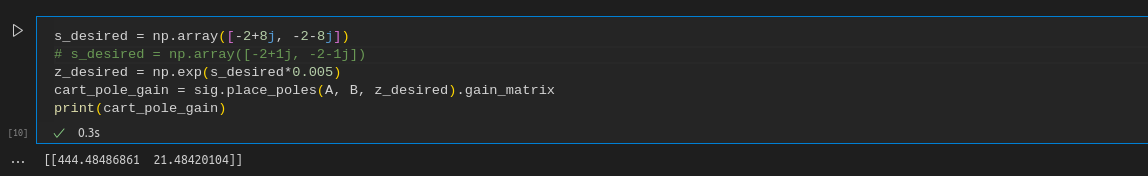


q6b\_2.gif is attached

1. **Repeat with different eigenvalues**

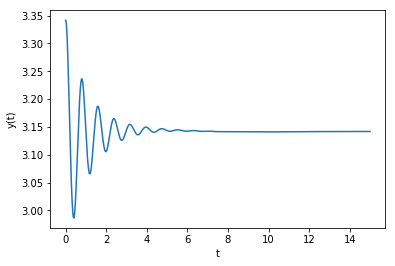
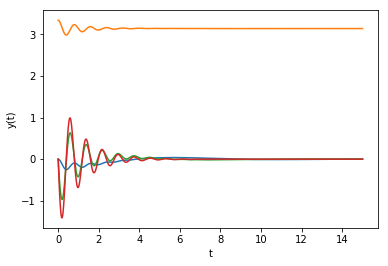
In this part, the desired eigenvalues become .

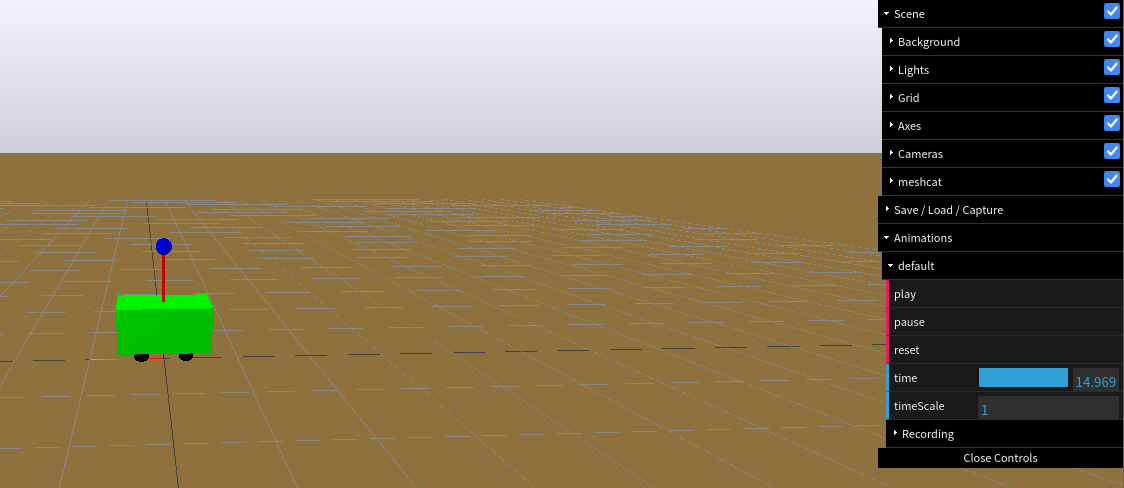
First calculate K:



As the result shows, K=[444.48486861 21.48420104]. The same K1 and K3 as Part 6 are taken in this part.

The figures are shown as follows. All of z, , , converge.



q7b\_1.gif is attached

**Analysis:**

In this part, compare

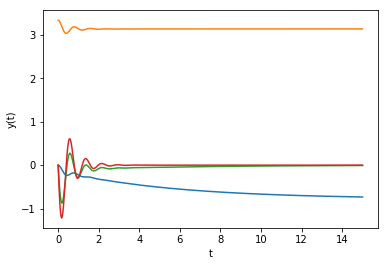
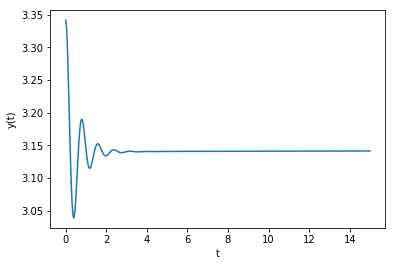
with eigenvalues of and

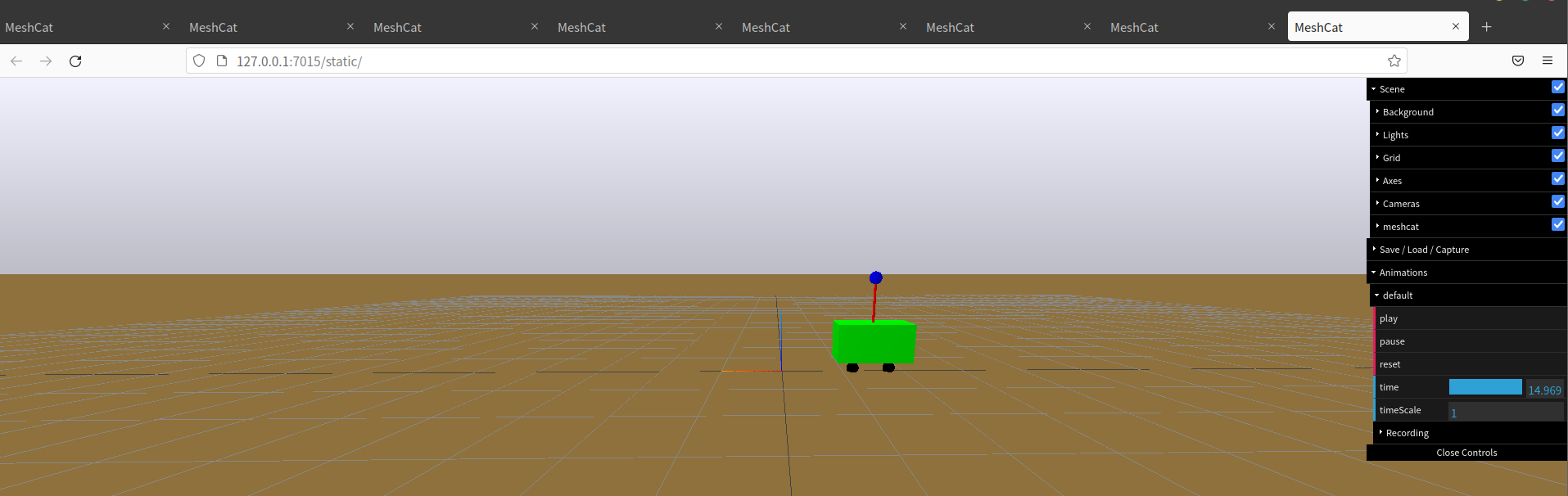
with eigenvalues of .

It can be found that the latter condition takes short time for to converge to and be in a stable state, which is around 5 second, whose respond is faster.

However, for the former one, it doesn’t show typical control plots, and converges to a value much larger than . And its ‘s converge time is much more than 5 seconds.

The figures are shown below. All of z, , , converge.





q7b\_2.gif is attached

**Analysis:**

Compared with

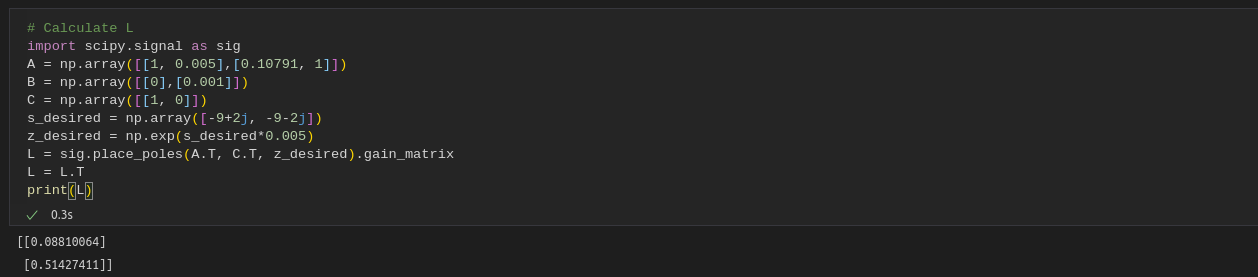
with eigenvalues of ,

with eigenvalues of takes shorter time for to converge to and be in a stable state. Both of the two conditions make convergent, and the latter one takes less than 4 seconds, while the former one needs more than 15 seconds. It is obvious that the latter one responds much faster.

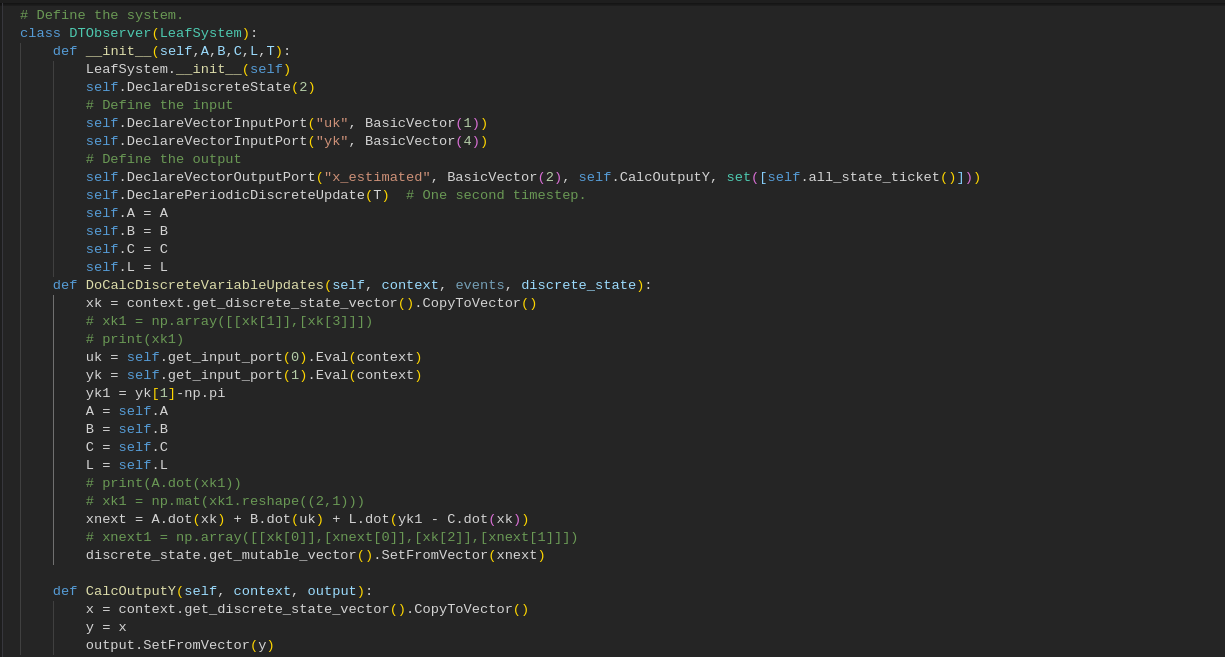
1. **Output feedback control design:**

The code of observer is shown as below.

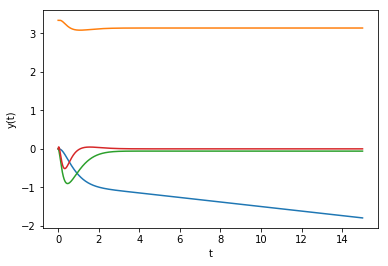
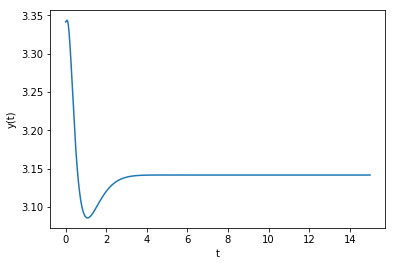
The first step is to compute L of the specific eigenvalues .

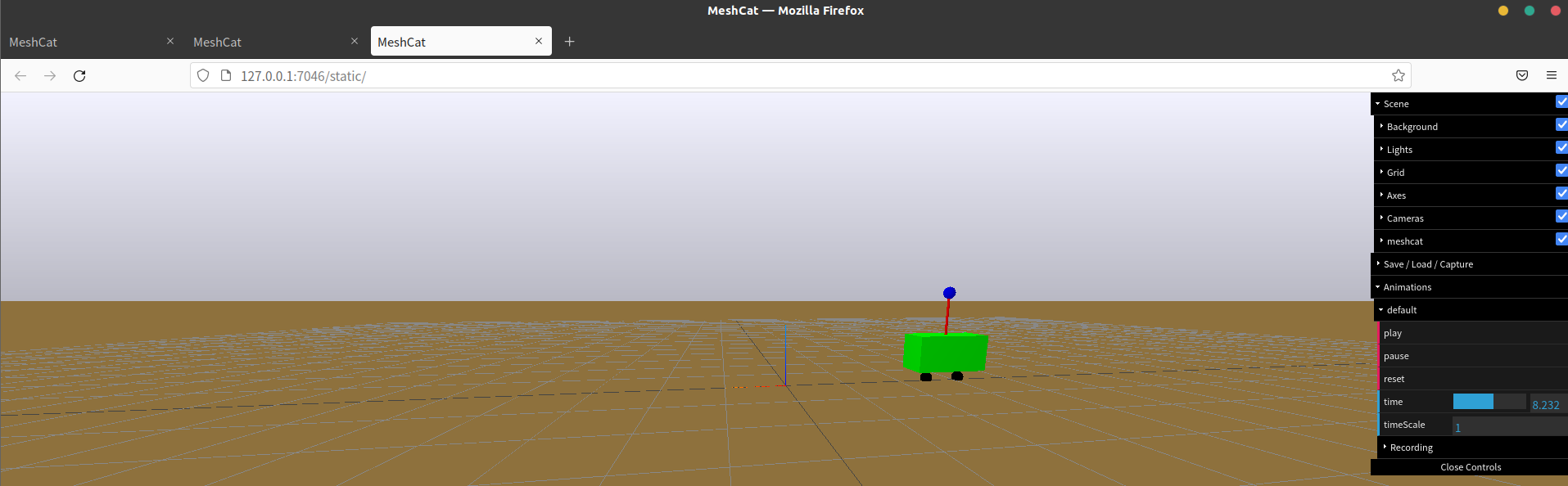


The second step is to set up the model of DT Luenberger observer.

****

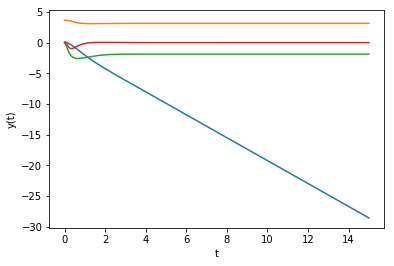
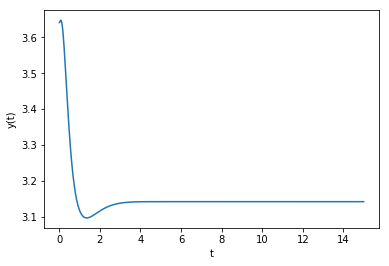
**Disturbance = 0.2**





q81.gif

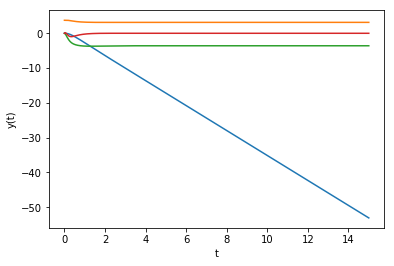
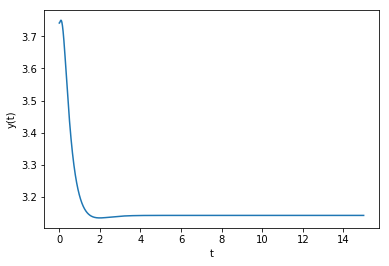
**Disturbance = 0.5**

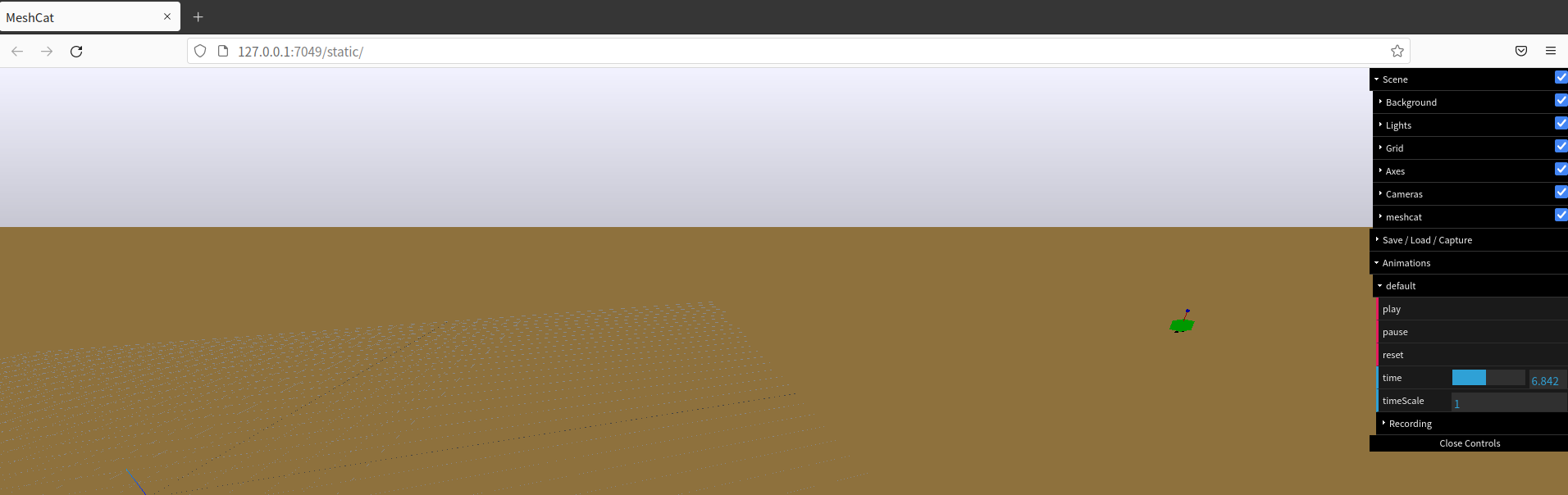




q82.gif

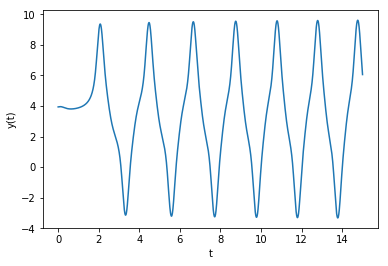
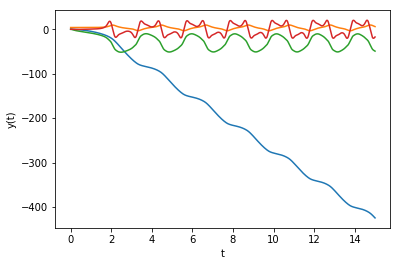
**Disturbance = 0.6**

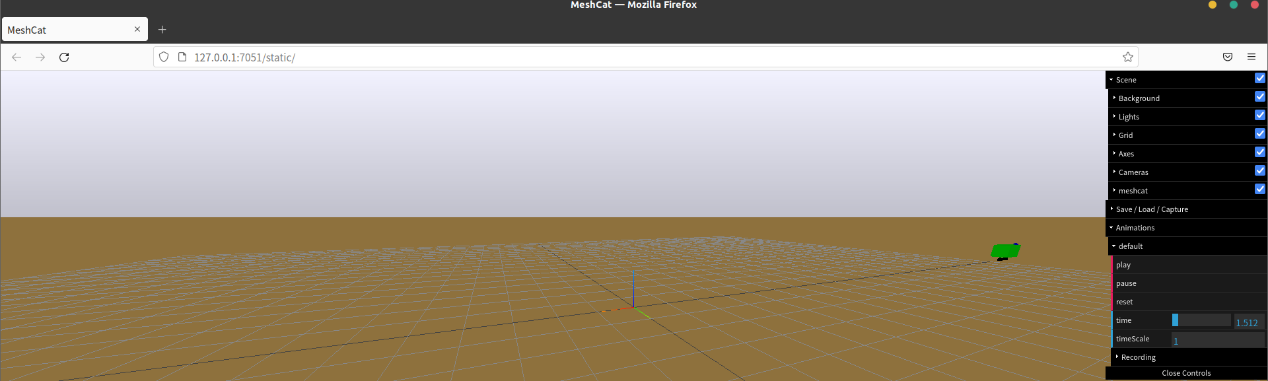




q83.gif

**Disturbance = 0.8**

****



4 different initial conditions are taken in this part.

For disturbance = 0.2, 0.4, 0.6, all of the converge to in about 3 seconds. However, the smaller the disturbance, the bigger the oscillation before getting stable. Also, the bigger the disturbance, the smaller the after getting stable, resulting in the difference between z curves.

Moreover, for disturbance = 0.8, oscillates over time, which shows it is not a stable condition. It is because the code is based on , it only works around [], which means the disturbance can’t be too big, and disturbance of 0.8 is too big to fit.