ECE311 Introduction to Linear Control System

Lab 2: Control of a Magnetically Levitated Ball

Group number: Team 15
Group member: Weizhou Wang 1004421262
Shuang Wu 1004728119
Lingwei Sun 1004723276
Naize Yang 1005274579

Video link:
https://drive.google.com/file/d/1SRXAT3P004ZDmxT4eEOlzaxooyCteS1Z/view?usp=sharing

3. Theoretical Linearization

The detailed hand-written procedure for finding the following output in part3/output1 is included at the end of this report.

Output 1.1

Write in your report the equilibrium $\overline{x} \in R^3$ and the control $\overline{u}$ that you determined above. These quantities should contain $\overline{y}$, and M, L_a, R_a, g, k_m as parameters.

$$\overline{x} = \begin{bmatrix} \overline{x_1} \\ \overline{x_2} \\ \overline{x_3} \end{bmatrix} = \begin{bmatrix} \overline{y} \\ 0 \\ \sqrt{\frac{gM}{k_m}}\overline{y} \end{bmatrix}$$

$$\overline{u} = R_a\sqrt{\frac{gM}{k_m}}\overline{y}$$

The above equilibrium values were calculated by solving the equation of f($\overline{x}$, $\overline{u}$)=0.

Output 1.2

Write the linearization of (3) at $\overline{x}$ with control $\overline{u}$, defining all error variables.

Define error variables:

$$\widetilde{x} = x - \overline{x}, \widetilde{u} = u - \overline{u}, \widetilde{y} = y - \overline{x_1} = y - \overline{y}$$

The linearization of (3):

$$\dot{\widetilde{x}} = \begin{bmatrix} 0 & 1 & 0 \\ \frac{2g}{\overline{y}} & 0 & -2\frac{\sqrt{\frac{gkm}{M}}}{\overline{y}} \\ 0 & 0 & -\frac{Ra}{La} \end{bmatrix} \widetilde{x} + \begin{bmatrix} 0 \\ 0 \\ \frac{1}{La} \end{bmatrix} \widetilde{u}$$

$$\dot{\widetilde{y}} = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \widetilde{x}$$

The matrices in the linearized model above were calculated by taking the gradient of f(x,u) and h(x, u) at x = $\overline{x}$ and u = $\overline{u}$.

Output 1.3

Double-check carefully your work. Verify that f($\overline{x}$, $\overline{u}$)=0 and that the various partial derivatives were computed correctly.

We have verified f($\overline{x}$, $\overline{u}$)=0 for $\overline{x}$ and $\overline{u}$ that we calculated. Also, the partial derivatives were correctly calculated and the linearization we calculated is correct.

4. Numerical Linearization and Stability Assessment

Output 2.1

Print the numerically derived matrices A, B and their theoretically derived counterparts, A1, B1.

```
A =

         0    1.0000         0
  196.2000         0  -62.6418
         0         0  -60.0000
```

```
B =

         0
         0
   20.0000
```

```
A1 =

         0    1.0000         0
  196.2000         0  -62.6418
         0         0  -60.0000
```

```
B1 =

     0
     0
    20
```

Output 2.2

Print the errors described above, and comment on the accuracy of the numerical approximation performed by Matlab/Simulink.

The error between A and A1:

```
err_A =

   4.7481e-06
```

The error between B and B1:

```
err_B =

   1.1898e-11
```

The errors between the theoretically derived and the numerically approximated matrices are in the order of 10^{-6} and 10^{-11} respectively, which is so small that the difference between them can be ignored in practice. Therefore, the numerical approximation performed by Matlab/Simulink has really high accuracy.

Output 2.3
Print the eigenvalues of A1 and the poles of G. Note that the poles of the transfer function coincide with the eigenvalues of the matrix A1. Is the linearized model internally stable? Is it BIBO stable? Comment on how your findings of stability conform with physical intuition.

Eigenvalues of A1:

```
e =

   14.0071
  -14.0071
  -60.0000
```

Poles of G:

```
pole =

  -60.0000
   14.0071
  -14.0071
```

From the above results, the eigenvalues of A1 coincide with the poles of the transfer function G.

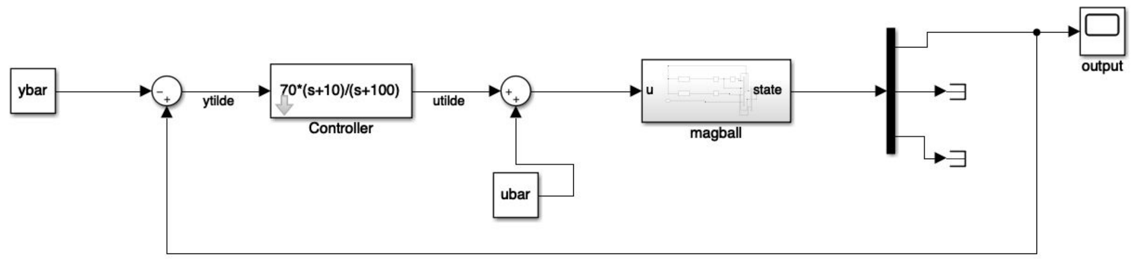
The linearized model is internally unstable, as one of the eigenvalues of A1 is 14.0071, which has a real part > 0, resulting in an unstable system.

The model is also not BIBO stable, since one of the poles of G is 14.0071, which has a real part > 0, resulting in an unstable system.

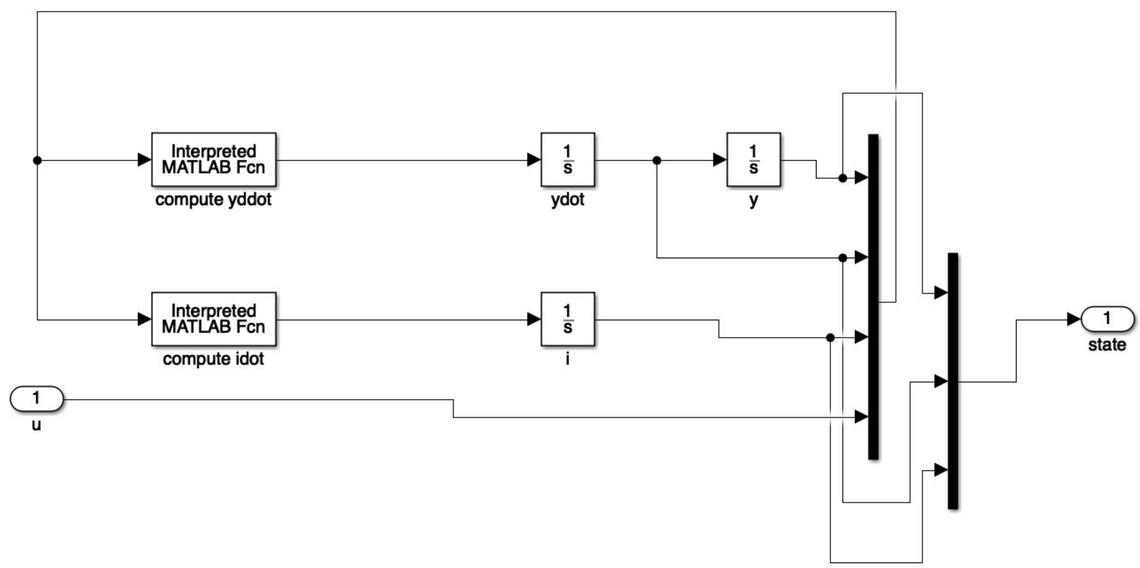
The above findings conform with the physical intuition of the problem. There are two forces acting on the steel ball, where one of them is the force imparted by the electromagnet, and the other force is the gravity of the ball itself. The ball can stay still if and only if the ball is initially static, and the electromagnetic force always cancels out with gravity. Otherwise, the ball will move. If the ball moves closer to the electromagnet, then the electromagnetic force will be greater than the gravity and the ball will be attracted by the electromagnet, resulting in a constantly decreasing y. If the ball moves away from the electromagnet, the electromagnetic force will become less than the gravity, so the ball will keep falling, resulting in a constantly increasing y. In either case, the value of y will explode and result in a system that is neither internally stable nor BIBO stable.

5. Feedback Control of the Magnetic Levitation System

The diagram below is the schematics for the block diagram of the system with the lead controller, with the initial parameters labelled as z=10, p=100 and K=70.



The diagram below is the schematics for the subsystem magball, with initialization of the ball at 15cm from the magnet face (y(0) = 0.15m).



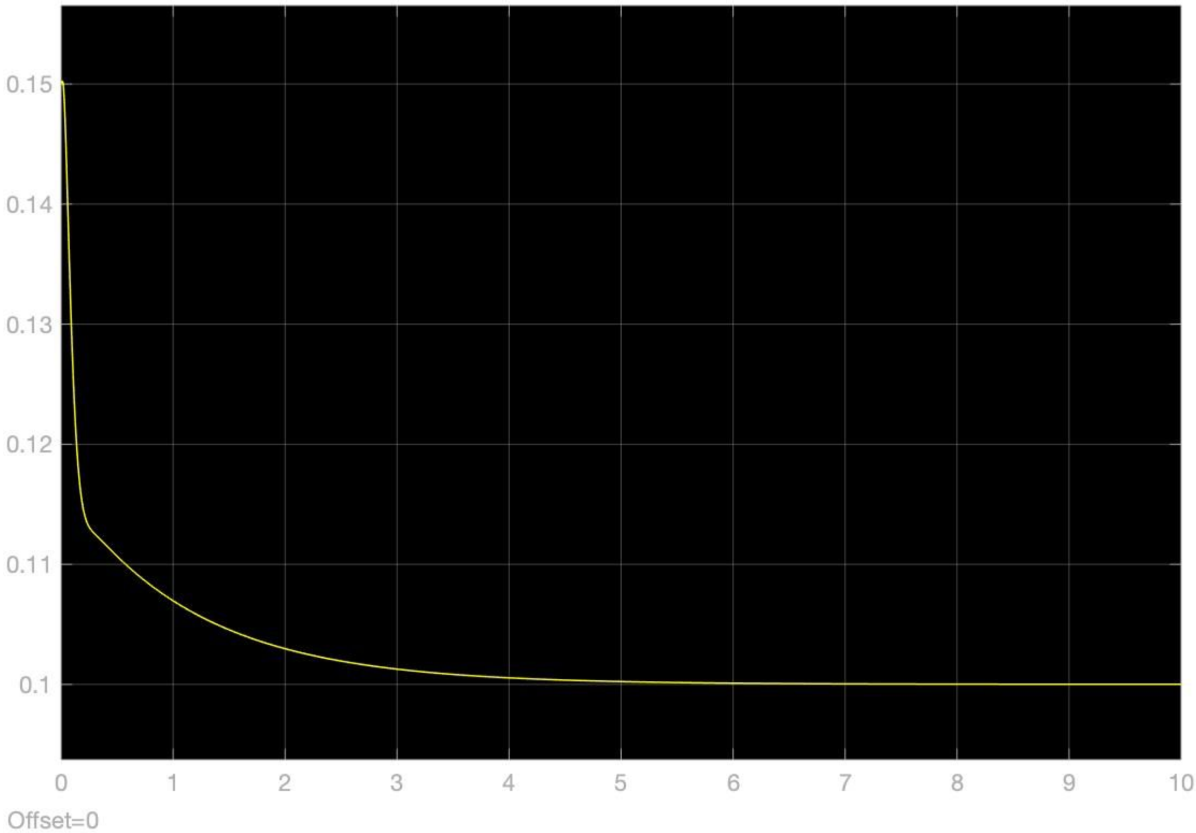
Output 3:

1.1) Print the value of K you found above that which the closed-loop system is BIBO stable.

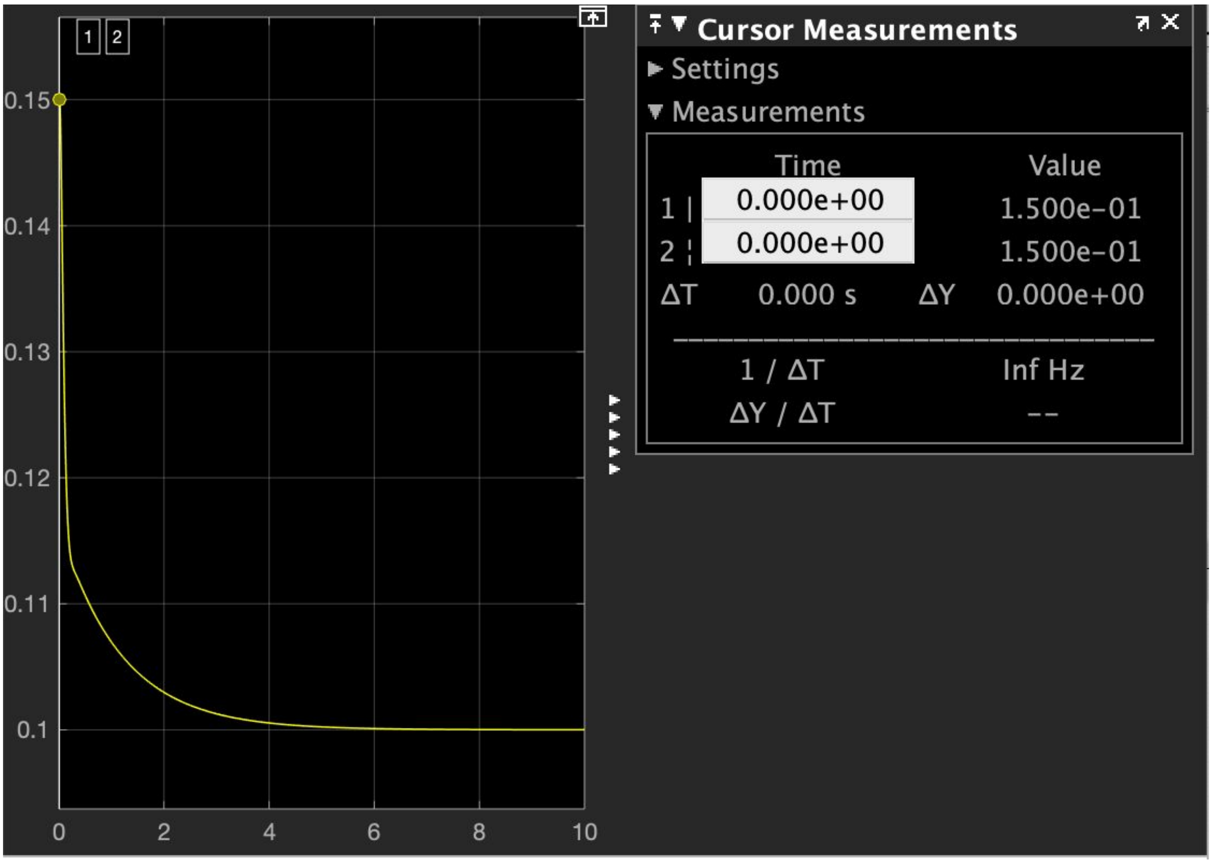
```
K =

   100
```

1.2) Produce the figure of the output y(t) when y(0) = 0.15m.

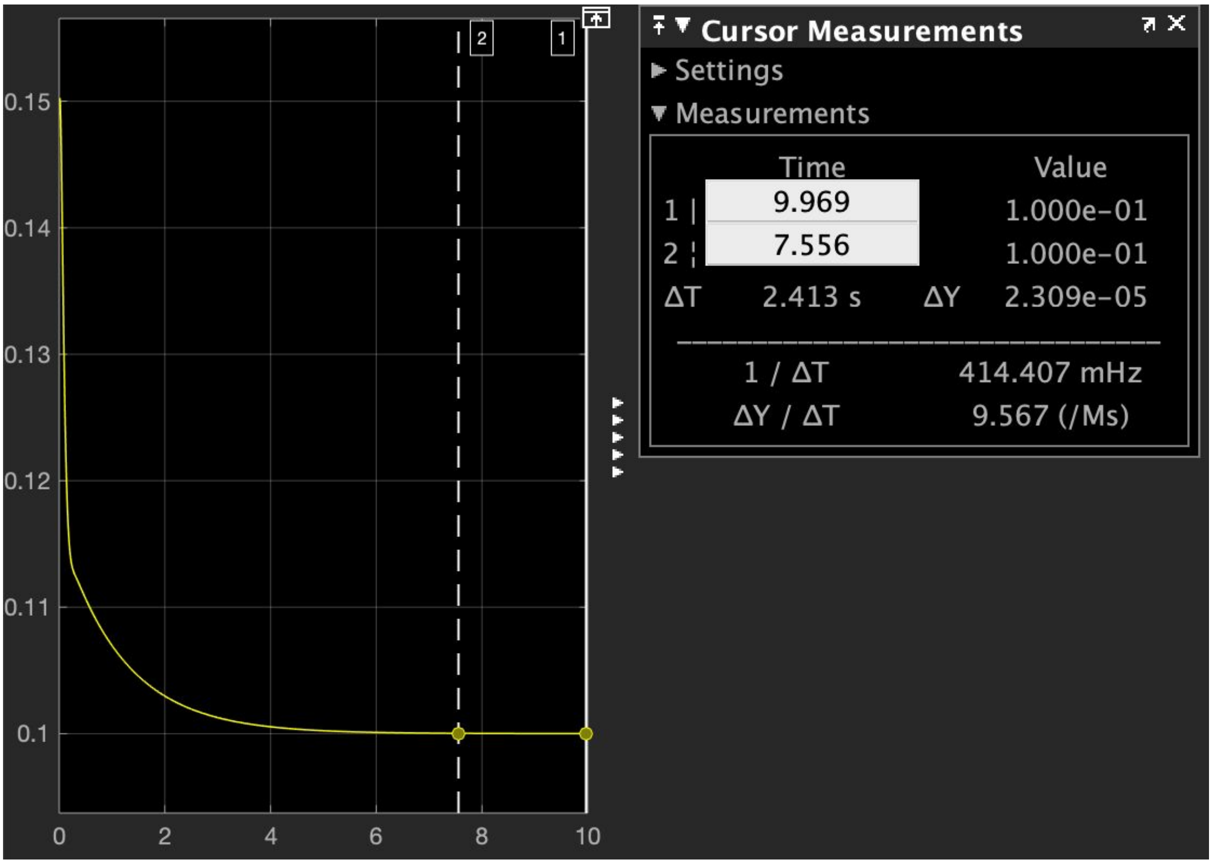
The following diagram is the output y(t) of the system with K = 100.



The following diagram with cursor at t=0, indicates that y(0) = 0.15m.



The following diagram with the cursor at t=7.556s and t=9.969s, indicates that output converges to 0.1 metres.



2) The following procedures are followed when finding K with observations provided along the way. The team first started from K = 10, increasing K by increments of 10, repeating the following procedures until we found the desired K value.

1. Constructing the controller function based on formula $K \cdot \frac{s+z}{s+p}$, where K = K, z = 10 and p = 100.
2. In order to find the poles of the closed-loop system transfer function, we compute the roots/zeros of the denominator $1 - C(s)G(s)$ using the zpkdata() function.
3. Printing the roots of the $1 - CONTROLLER \cdot G(s)$.
4. Plotting the roots using the rlocus() function, with zeros labelled as “o”.
5. Determine whether the output function at the time domain is BIBO stable or not based on THM2. If all roots are on the open left half-plane, the system is BIBO stable.

Taking K=10 as an example for the procedures and observations:

1. controller function:

```
CONTROLLER =

  10 s + 100
  ----------
   s + 100

Continuous-time transfer function.
```

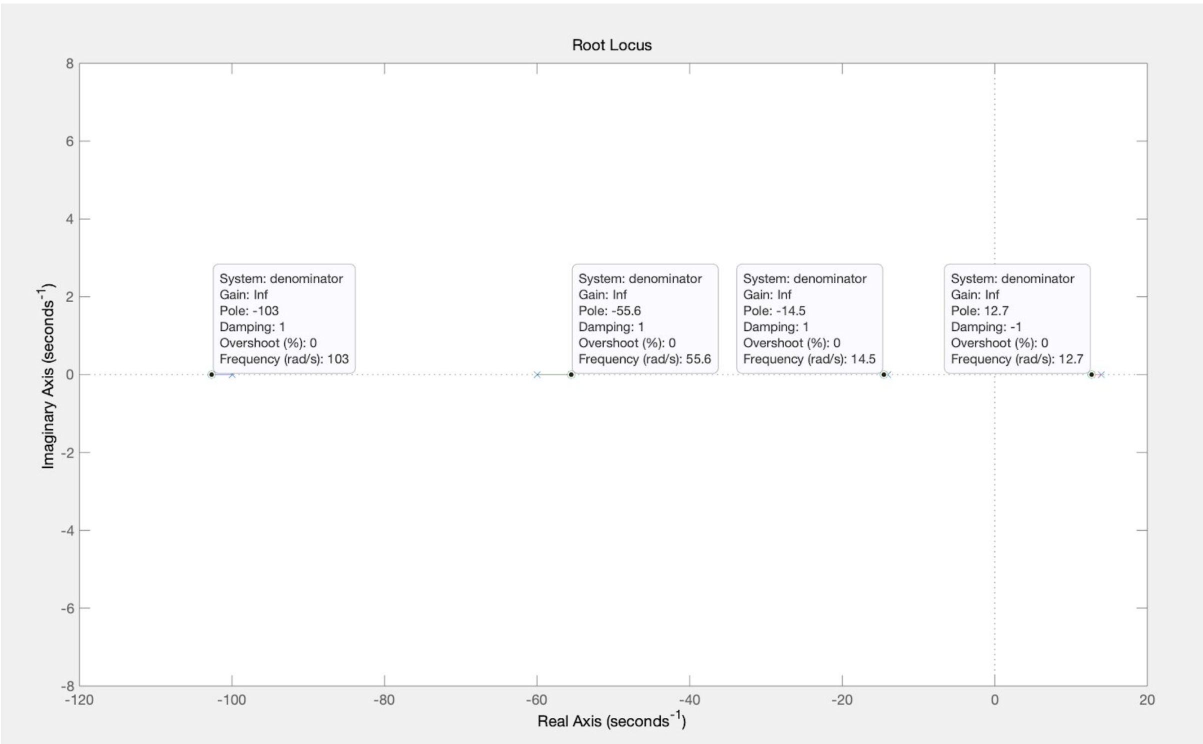
2. denominator:

```
denominator =

  s^4 + 160 s^3 + 5804 s^2 - 1.886e04 s - 1.052e06
  ------------------------------------------------
  s^4 + 160 s^3 + 5804 s^2 - 3.139e04 s - 1.177e06

Continuous-time transfer function.
```

3. zeros/roots:

```
ans =

 -102.6333
  -55.5569
  -14.5175
   12.7076
```

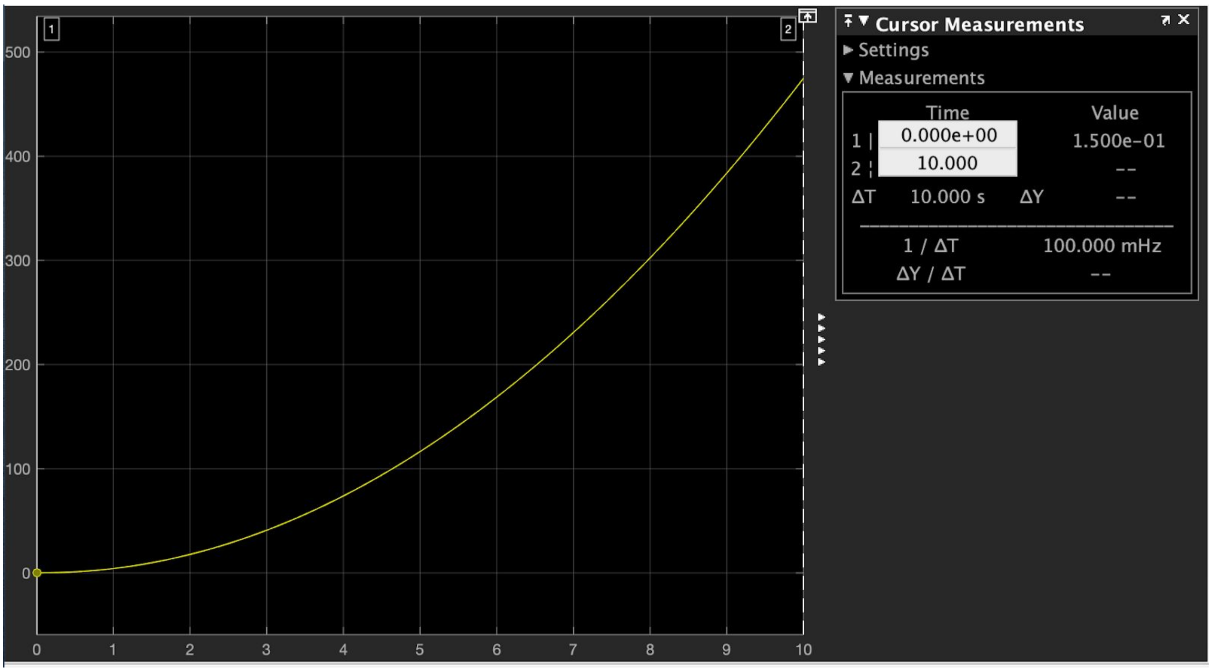
4. roots observed on the graph:



5. determination:

Since there exist a root/zero for $1 - C(s)G(s)$ that is on the right half-plane (12.7076). So, the system is BIBO unstable.

Using the Simulink to plot the output waveform in the time domain for a quick check when K = 10. The diagram below shows the output divergence when K = 10 with y(0) = 0.15m.



As we increased the K value, we discovered that the fourth root on the right half-plane, initially when K=10, is moving towards the left. When K is increased to 100, all roots are landed on the open left half-plane.

When K=100, the observations are in below:

1. controller function:

```
CONTROLLER =

  100 s + 1000
  ------------
    s + 100

Continuous-time transfer function.
```

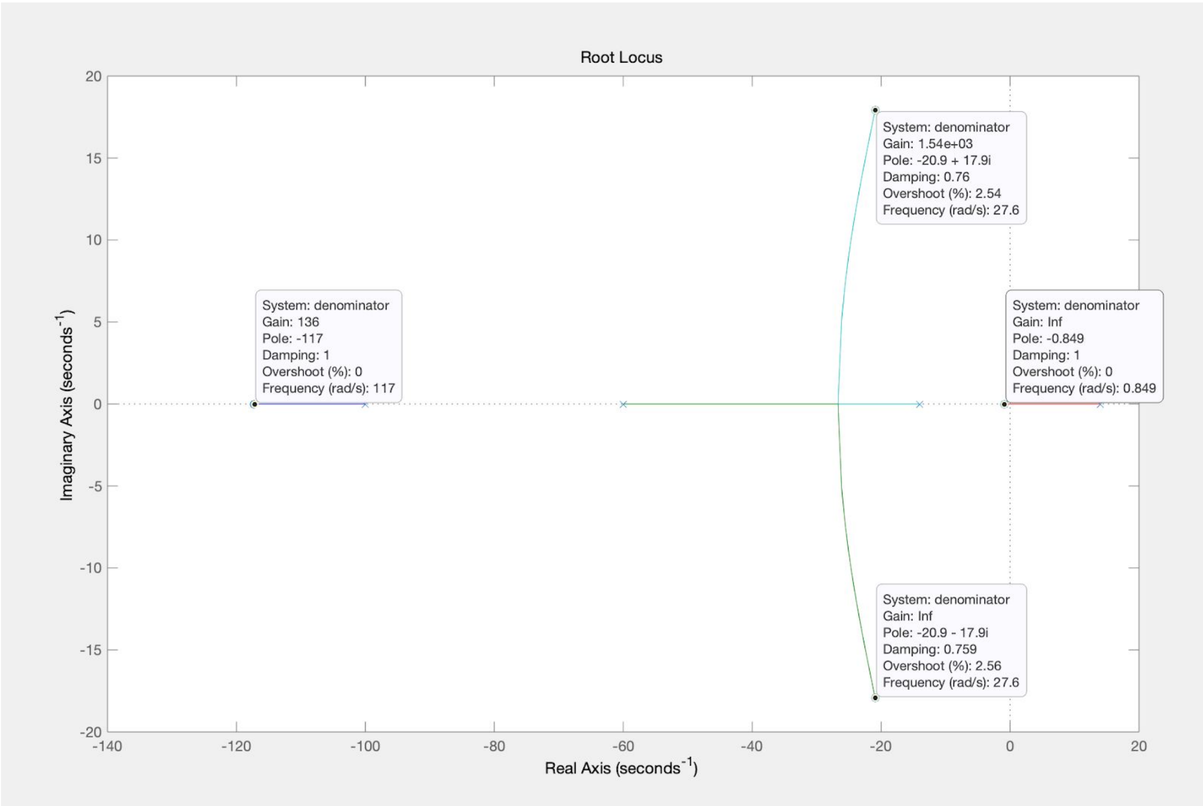
2. denominator:

```
denominator =

  s^4 + 160 s^3 + 5804 s^2 + 9.389e04 s + 7.564e04
  --------------------------------------------------
  s^4 + 160 s^3 + 5804 s^2 - 3.139e04 s - 1.177e06

Continuous-time transfer function.
```

3. zeros/roots:

```
ans =

   1.0e+02 *

  -1.1730 + 0.0000i
  -0.2093 + 0.1793i
  -0.2093 - 0.1793i
  -0.0085 + 0.0000i
```

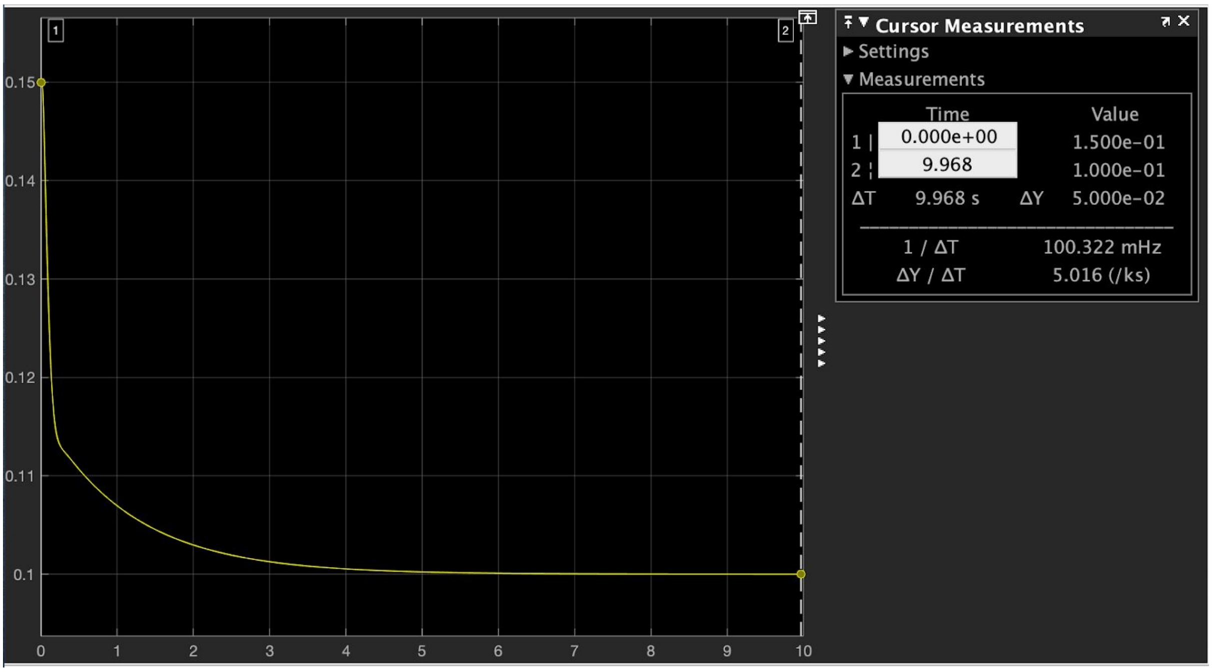
4. roots observed on the graph:



5. determination:

We observed all zeros/roots are in the open left half-plane. It indicates that the system is BIBO stable.

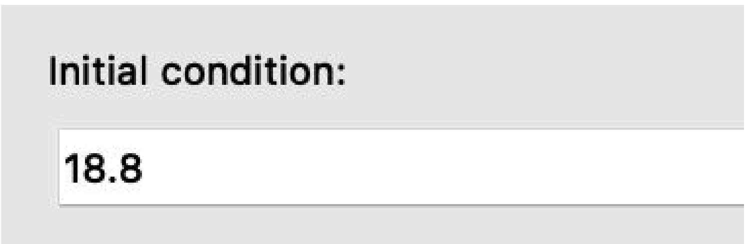
The plotted output using Simulink when K = 100 is provided in the previous part in Output3 which is also repeated below.

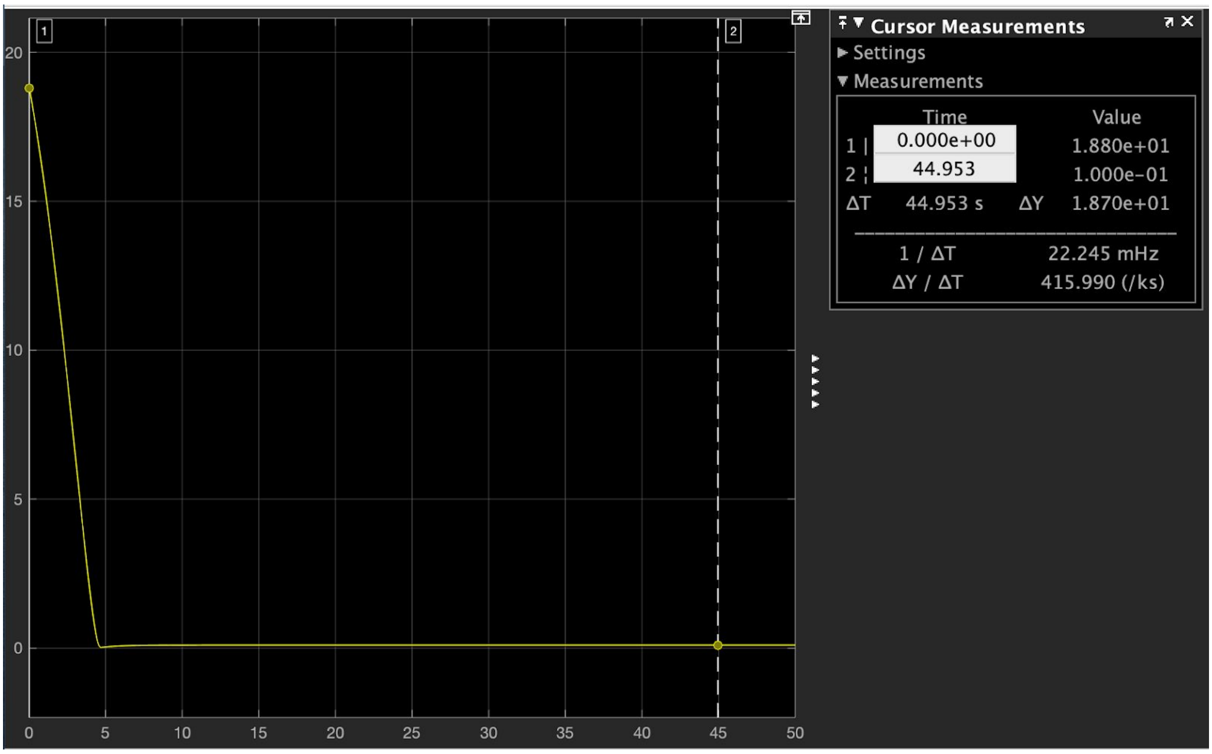


3.1) The range of initial conditions y(0) for which your controller succeeds in stabilizing the ball. We change the stop time to a larger number from 10s to 50s, to see the overall trends when the time approaches a large number (since sometimes, the y value is still increasing approaching t=10s).

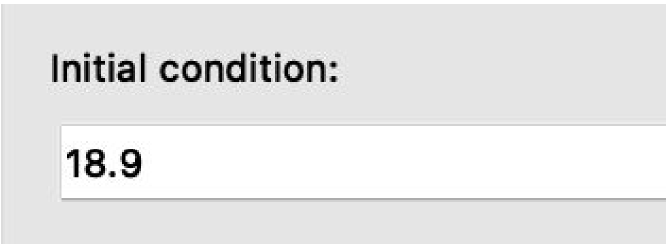
Rough upper limits = 18.8m
Rough lower limits = 0.06m

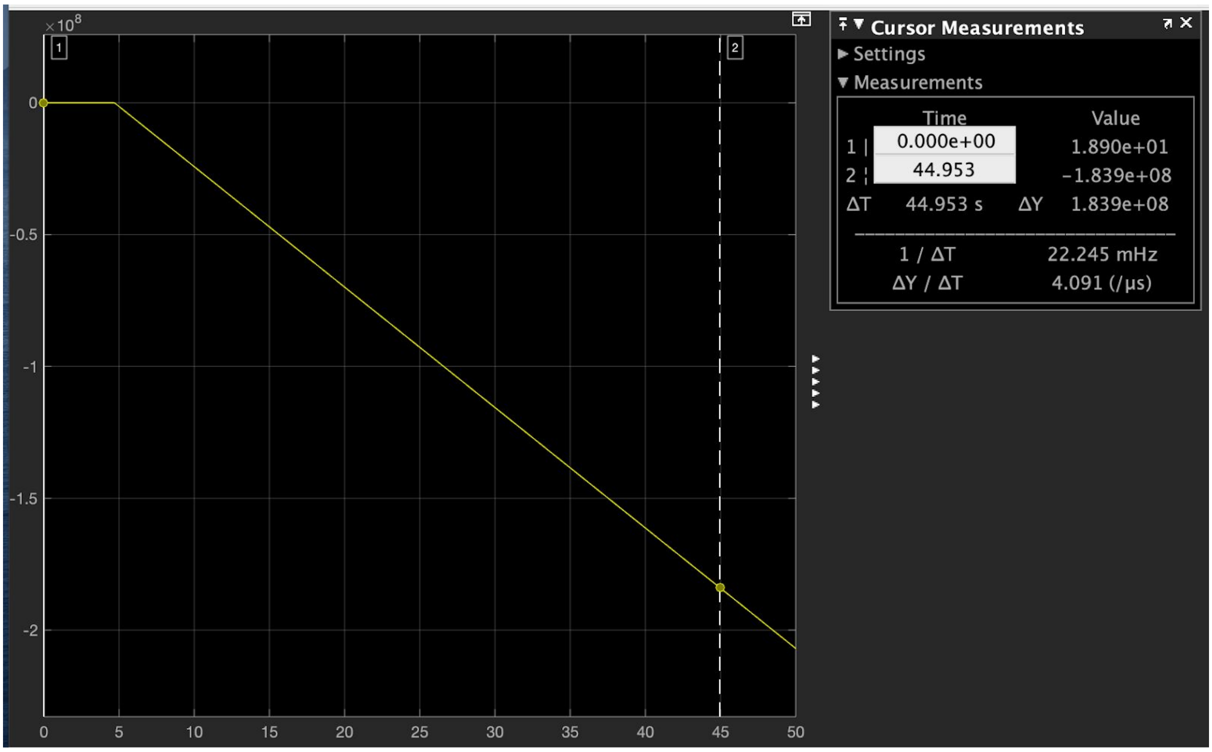
When y(0) = 18.8m, it converges:



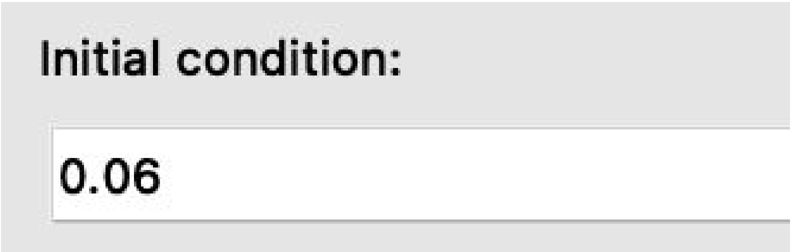


When y(0) = 18.9m, it diverges:

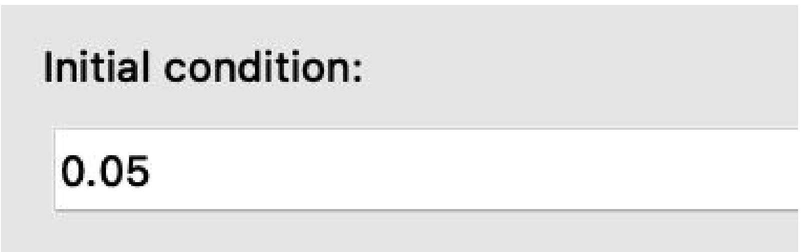


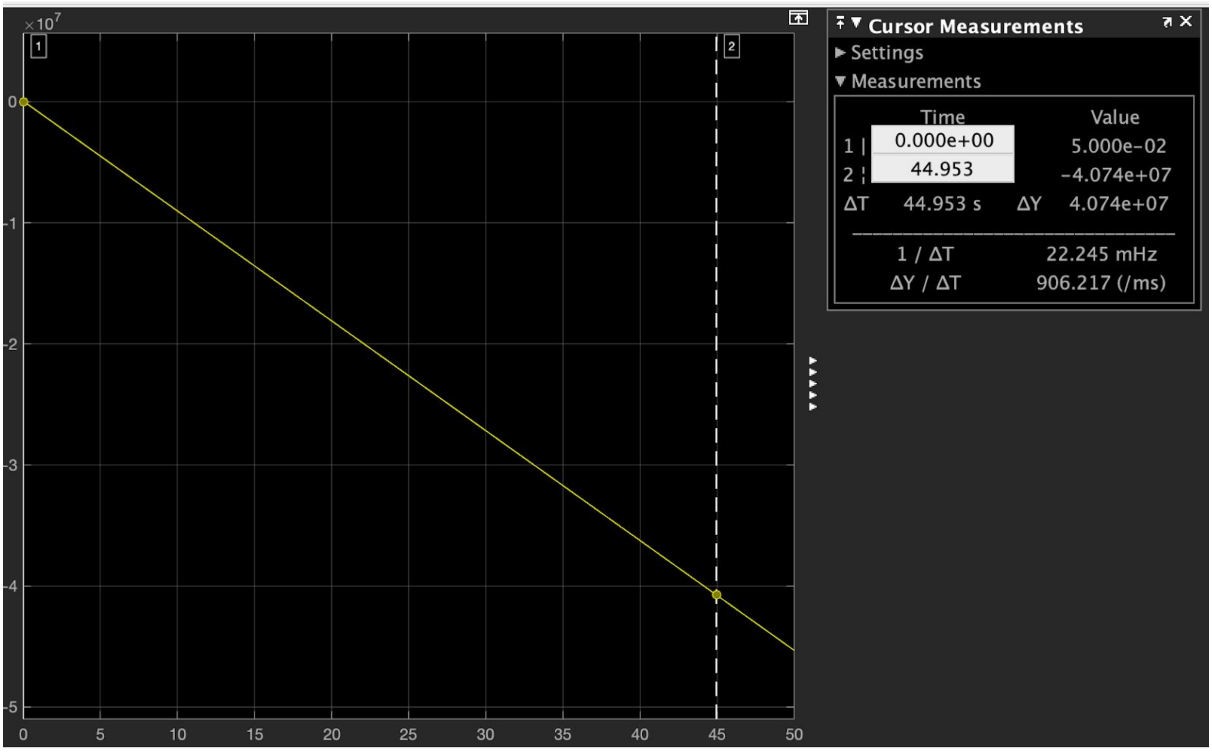


When y(0) = 0.06m, it converges:



When y(0) = 0.05m, it diverges:





3.2) Comment on why your controller does not work for initial conditions that are far from the equilibrium.

The controller is designed to stabilize the linearization and then applied to the nonlinear model. As we declared within the code previous part, $\overline{y} = 0.1m$ is the equilibrium value. The system is nonlinear, so we are applying the linearization of the model at the equilibrium point. Since the linearization method is taking the gradient of the vector function, it will only be accurate if it is near that equilibrium. So, by initializing y(0) exceeding the range from 0.06 to 18.8, it becomes far from the equilibrium point, so the linearization accuracy will be low, which leads to the instability of the output.

Detailed Handwritten procedure for finding outputs in part3:

$$f(\bar{x},\bar{u}) = \begin{bmatrix} \bar{x_2} \\ -\frac{km}{M}\frac{x_3^2}{x_1^2} + g \\ -\frac{Ra}{La}x_3 + \frac{1}{La}u \end{bmatrix} = \bar{0} \quad \therefore \quad \begin{array}{ll} \bar{x_2} = 0 & (1) \\ -\frac{km}{M}\frac{\bar{x_3}^2}{\bar{x_1}^2} + g = 0 & (2) \\ -\frac{Ra}{La}\bar{x_3} + \frac{1}{La}\bar{u} = 0 & (3) \end{array}$$

$$(3): \ \bar{x_3} = \frac{\bar{u}}{Ra}$$

since $\bar{x_1} = \bar{y}$, $(2): \ \bar{x_3} = \sqrt{\frac{gm}{km}}\,\bar{y}$

Also, $\frac{\bar{u}}{Ra} = \sqrt{\frac{gm}{km}}\,\bar{y} \quad \therefore \ \bar{u} = Ra\sqrt{\frac{gm}{km}}\,\bar{y}$

$$\therefore \ \underline{\bar{x} = \begin{bmatrix} \bar{y} \\ 0 \\ \sqrt{\frac{gm}{km}}\,\bar{y} \end{bmatrix}}, \quad \underline{\bar{u} = Ra\sqrt{\frac{gm}{km}}\,\bar{y}}$$

$$\tilde{x} = x - \bar{x}, \quad \tilde{u} = u - \bar{u}, \quad \tilde{y} = y - \bar{y}$$

$$\frac{\partial f(\bar{x},\bar{u})}{\partial x} = \begin{bmatrix} 0 & 1 & 0 \\ 2\frac{km\,\bar{x_3}^2}{M\bar{x_1}^3} & 0 & -2\frac{km\,x_3}{M x_1^2} \\ 0 & 0 & -\frac{Ra}{La} \end{bmatrix}_{\bar{x} = \begin{bmatrix} \bar{y} \\ 0 \\ \sqrt{\frac{gm}{km}}\,\bar{y} \end{bmatrix}} = \underbrace{\begin{bmatrix} 0 & 1 & 0 \\ \frac{2g}{\bar{y}} & 0 & -2\frac{\sqrt{\frac{gkm}{M}}}{\bar{y}} \\ 0 & 0 & -\frac{Ra}{La} \end{bmatrix}}_{A}$$

$$\frac{\partial f}{\partial u}(\bar{x},\bar{u}) = \begin{bmatrix} 0 \\ 0 \\ \frac{1}{La} \end{bmatrix} \leftarrow B$$

$$h(\bar{x},\bar{u}) = \bar{x_1}$$

$$\frac{\partial h(\bar{x},\bar{u})}{\partial x} = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} = C \qquad \frac{\partial h}{\partial u}(\bar{x},\bar{u}) = 0 = D$$

$$\dot{\tilde{x}} = A\tilde{x} + B\tilde{u}$$
$$\tilde{y} = C\tilde{x}$$