ECE356
Lab 2
Juho Kim (1005955085)
Hshmat Sahak (1005903710)
Mar 4, 2022

1. Introduction

In this lab, we will investigate the magnetic ball-suspension system. We will compute steady state conditions and use linearization to compute the state space model near equilibrium. We first do this by hand; then we use MATLAB to verify our results and familiarize ourselves with using SIMULINK for performing linearization.

The magnetic ball-suspension system, as well as its block representation, is depicted below. Two forces act on the ball. The first is gravitational force which follows the law of gravitation (i.e., it is equal to Mg). In our problem, $M = 1Kg$ and $g = 9.8m/s^2$. The second is force exerted by the electromagnet on the ball, given as $F = i^2/y^2$.In our problem, the electromagnet is represented as a series RL circuit with winding resistance $R = 3\Omega$ and winding inductance $L = 1H$.

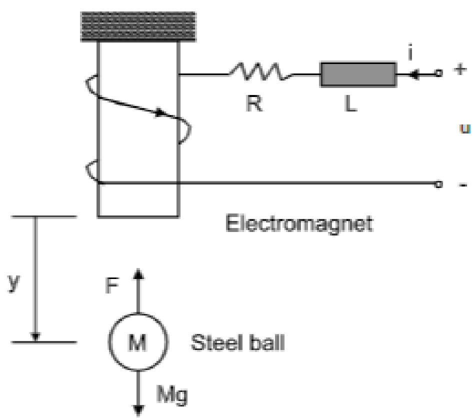
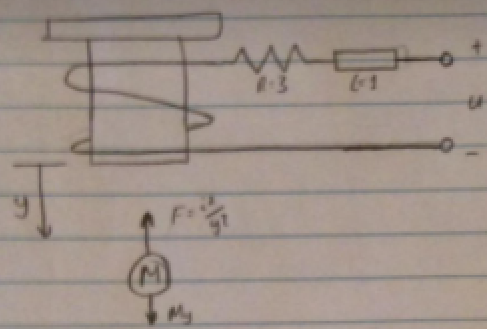


Figure 1: Block diagram of magnetic ball-suspension system

We first present individual lab preparation documents. Next, we provide explanations and results from both the SIMULINK model and MATLAB code.

2. Lab Preparation

Lab 2 Prep



$g = 9.8\,m/s^2$

$M = 1\,kg$

$R = 3\,\Omega$

$L = 1\,H$

1)

$$x = \begin{bmatrix} y \\ \dot{y} \\ i \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \Rightarrow \dot{x} = \begin{bmatrix} \dot{y} \\ \ddot{y} \\ i' \end{bmatrix}$$

RL Circuit equation: $u(t) = iR + L\,{}^{di}/_{dt}$

$$= iR + Li'$$

$$i' = \frac{u(t) - iR}{L}$$

$$i' = {}^{1}/_{L}\, u(t) - {}^{R}/_{L}\, i$$

Newton's Second Law: $M\ddot{y} = Mg - F$

$$\ddot{y} = g - {}^{F}/_{M}$$

$$\ddot{y} = g - \frac{i^2}{My^2}$$

$$\dot{x} = f(x,u) = \begin{bmatrix} x_2 \\ g - \frac{x_3^2}{Mx_1^2} \\ -{}^{R}/_{L}\, x_3 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \frac{u(t)}{L} \end{bmatrix}$$

$$y = h(x,u) = x_1$$

Now, substitute values for M, R, and L to get:

$$\boxed{\dot{x} = \begin{bmatrix} x_2 \\ g - \frac{x_3^2}{x_1^2} \\ -3x_3 + u(t) \end{bmatrix} \quad y = x_1}$$

2) At equilibrium, $\dot{x}^* = 0$

$\dot{x}^* = 0 \Rightarrow \quad x_2 = 0 \qquad \frac{x_3^2}{Mx_1^2} = g \qquad u(t) = Rx_3$

Given $y = x_1^* = y^*$:

$x_2 = 0 \qquad (x_3^*)^2 = Mg(y^*)^2 \qquad u^*(t) = Rx_3^*$

$x_3^* = \pm\sqrt{Mg}\, y^* \qquad = \pm R\sqrt{Mg}\, y^*$

Substitute $M = 1kg, R = 3, L = 1$:

$x_2 = 0 \qquad x_3^* = \pm\sqrt{g}\, y^* \qquad u^*(t) = \pm 3\sqrt{g}\, y^*$

Pick positive solution: $x_2 = 0 \quad x_3^* = \sqrt{g}\, y^* \quad u^*(t) = 3\sqrt{g}\, y^*$

$(x^*, u^*) = ([y^*, \dot{y}^*, i^*], u^*) = ([y^*, 0, \sqrt{g}\, y^*], 3\sqrt{g}\, y^*)$

$$\boxed{x^* = \begin{bmatrix} y^* \\ 0 \\ y^*\sqrt{g} \end{bmatrix} \quad u^* = 3y^*\sqrt{g}}$$

Linearize model:

$$\dot{x} = f(x,u) \qquad y = h(x,u)$$

$$A = \frac{\partial f}{\partial x}\bigg|_{(x^*,u^*)} \quad B = \frac{\partial f}{\partial u}\bigg|_{(x^*,u^*)} \quad C = \frac{\partial h}{\partial x}\bigg|_{(x^*,u^*)} \quad D = \frac{\partial h}{\partial u}\bigg|_{(x^*,u^*)}$$

$$f(x,u) = \begin{bmatrix} x_2 \\ g - \frac{x_3^2}{x_1^2} \\ -3x_3 + u(t) \end{bmatrix} \qquad h(x,u) = x_1$$

$$\frac{\partial f}{\partial x} = \begin{bmatrix} 0 & 1 & 0 \\ \frac{2x_3^2}{x_1^3} & 0 & -\frac{2x_3}{x_1^2} \\ 0 & 0 & -3 \end{bmatrix} \qquad \frac{\partial f}{\partial u} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \qquad \frac{\partial h}{\partial x} = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \qquad \frac{\partial h}{\partial u} = [0]$$

$$\frac{\partial f}{\partial x}\bigg|_{(x^*,u^*)} = \begin{bmatrix} 0 & 1 & 0 \\ \frac{2(g)(y^*)^2}{(y^*)^3} & 0 & -\frac{2y^*\sqrt{g}}{(y^*)^2} \\ 0 & 0 & -3 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ \frac{2g}{y^*} & 0 & -\frac{2\sqrt{g}}{y^*} \\ 0 & 0 & -3 \end{bmatrix}$$

$$\boxed{\text{So: } A = \begin{bmatrix} 0 & 1 & 0 \\ \frac{2g}{y^*} & 0 & \frac{-2\sqrt{g}}{y^*} \\ 0 & 0 & -3 \end{bmatrix} \quad B = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad C = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \quad D = [0]}$$

As usual in linearization:

$$\boxed{\delta x = x - x^* \quad \delta u = u - u^* \quad \delta y = h(x,u) - h(x^*,u^*)}$$

3) From $\frac{d}{dt}(\delta x) = A\delta x + B\delta u$

$$\delta y = C\delta x + D\delta u$$

We get:

$$G(s) = \frac{\delta Y(s)}{\delta U(s)} = C(sI-A)^{-1}B + D$$

$$= \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} s & -1 & 0 \\ -2g & s & 2\sqrt{g} \\ 0 & 0 & s+3 \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} + [0]$$

First row, last col of inverse matrix:

$$G(s) = \frac{1}{\det(sI-A)}\left(-2\sqrt{g}\right)$$

$$\det(sI-A) = (s+3)(s^2-2g)$$

$$\boxed{G(s) = \frac{-2\sqrt{g}}{(s^2-2g)(s+3)}}$$

4) Use residue formula

$$G(s) = \frac{-2\sqrt{g}}{(s^2-7g)(s+3)}$$

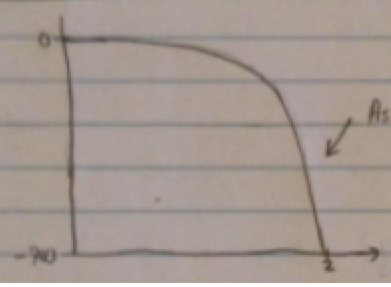
$$G(s)e^{st} = \frac{-2\sqrt{g}}{(s^2-7g)(s+3)} e^{st}$$

$$= \frac{-2\sqrt{g}}{(s-\sqrt{7g})(s+\sqrt{7g})(s+3)} e^{st}$$

$$\mathcal{L}^{-1}\{G(s)\} = -2\sqrt{g}\left[\frac{e^{st}}{(s-\sqrt{7g})(s+3)}\Big|_{s=-\sqrt{7g}} + \frac{e^{st}}{(s+\sqrt{7g})(s+3)}\Big|_{s=\sqrt{7g}} + \frac{e^{st}}{(s+\sqrt{7g})(s-\sqrt{7g})}\Big|_{s=-3} \right]$$

$$\boxed{g(t) = -2\sqrt{g}\left[\frac{e^{-\sqrt{7g}\,t}}{(-2\sqrt{7g})(-\sqrt{7g}+3)} + \frac{e^{\sqrt{7g}\,t}}{(2\sqrt{7g})(\sqrt{7g}+3)} + \frac{e^{-3t}}{(-3+\sqrt{7g})(-3-\sqrt{7g})} \right]}$$

Plot of solution:



As $t \to \infty$, approach $\frac{-2\sqrt{g}}{(2\sqrt{7g})(\sqrt{7g}+3)} e^{\sqrt{7g}\,t}$

since $e^{-\sqrt{7g}\,t}$ and $e^{-3t} \to 0$ as $t \to \infty$

3. Experiment

3.1. Building the SIMULINK model

Question: Write down the two Fcn functions in the space below:

Answer: From our lab preparation, we have:

$$y'' = g - (x_3/x_1)^2$$

and

$$i' = u - 3 \cdot x_3$$

In our SIMULINK model, it follows that our two Fcn functions are:

y_dot_dot = 9.8 - u(3) ^ 2 / u(1) ^ 2
i_dot = u(4) - 3 * u(3)

Images of the SIMULINK model are shown below.

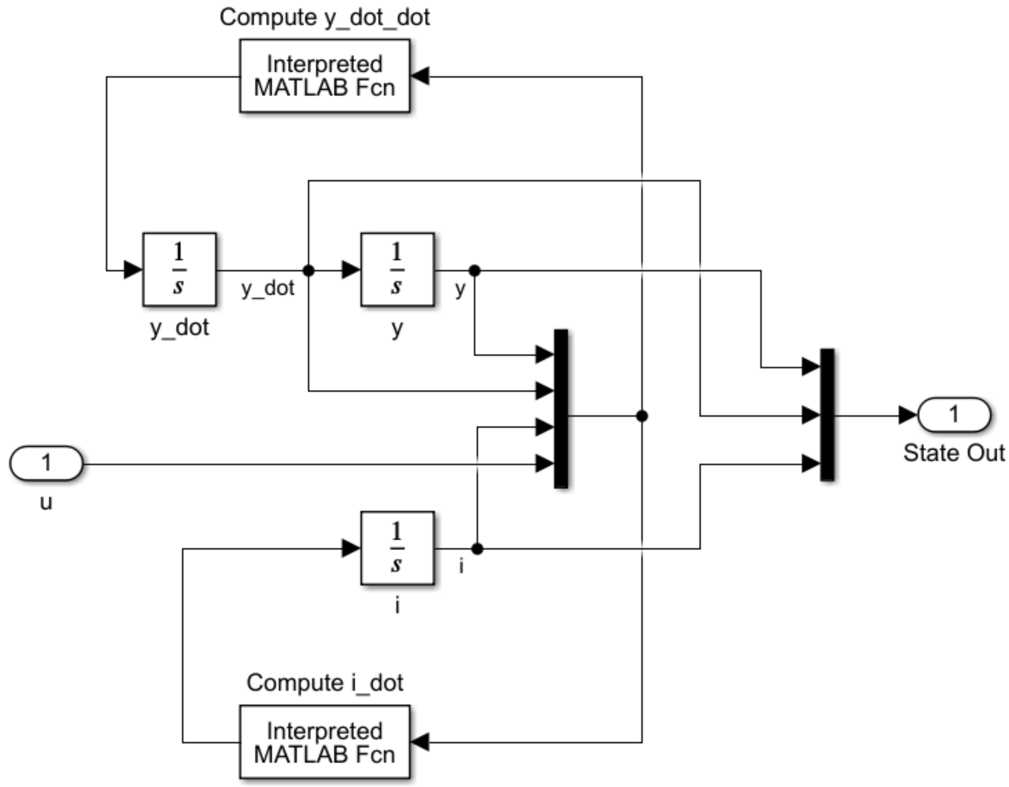


Figure 2: High-level SIMULINK model

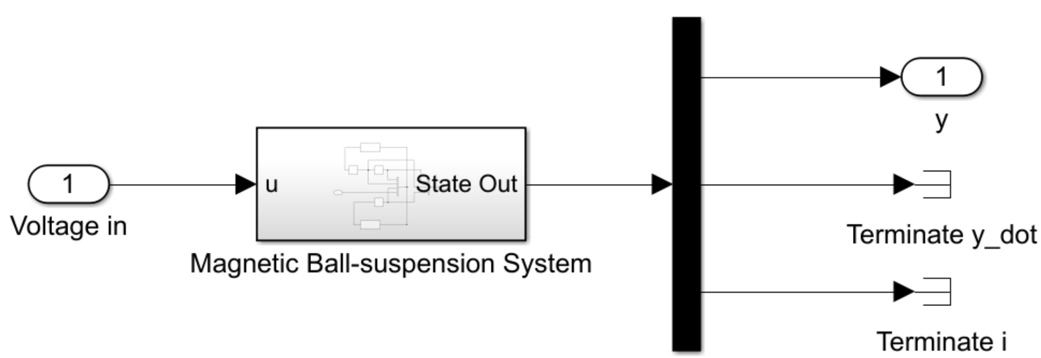


Figure 3: Subsystem block

3.2. Linearizing the Model in MATLAB

Question: Write down numerical values of the equilibrium point (x * , u *), corresponding to the constant position y * = 1.

Answer: Our MATLAB code output gives:

```
Part 2


yy0 =

0
0


i0 =

-(7*5^(1/2)*y)/5
 (7*5^(1/2)*y)/5


u0 =

-(21*5^(1/2)*y)/5
 (21*5^(1/2)*y)/5

yy0, i0, and u0:
yy0 =

0


i0 =

(7*5^(1/2)*y)/5


u0 =

(21*5^(1/2)*y)/5
```

So,

$$x^{\cdot} = \begin{bmatrix} y^{\cdot} \\ 0 \\ \frac{7\cdot\sqrt{(5)}\cdot y^{\cdot}}{5} \end{bmatrix}, u^{\cdot} = \frac{21\cdot\sqrt{(5)}\cdot y^{\cdot}}{5}$$

Setting y^* = 1, we obtain:

$$x^{\cdot} = \begin{bmatrix} 1 \\ 0 \\ \frac{7\cdot\sqrt{(5)}}{5} \end{bmatrix}, u^{\cdot} = \frac{21\cdot\sqrt{(5)}}{5}$$

Question: Write down A, B and the eigenvalues of A in the space below. Is the linearized system stable or unstable?

Answer: Our MATLAB code gives:

```
A =

[       0, 1,                     0]
[98/(5*y), 0, -(14*5^(1/2))/(5*y)]
[       0, 0,                    -3]


B =

0
0
1


C =

[1, 0, 0]


D =

0
```

Setting y^*=1, we get:

A =

```
[   0, 1,                 0]
[98/5, 0, -(14*5^(1/2))/5]
[   0, 0,                -3]
```

B =

```
0
0
1
```

C =

```
[1, 0, 0]
```

D =

```
0
```

Eigenvalues of matrix A

ans =

```
                  -3
-(7*2^(1/2)*5^(1/2))/5
 (7*2^(1/2)*5^(1/2))/5
```

Summarizing our results:

$$A = \begin{bmatrix} 0 & 1 & 0 \\ \frac{98}{5} & 0 & \frac{-14\cdot\sqrt{(5)}}{5} \\ 0 & 0 & -3 \end{bmatrix}, B = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

Eigenvalues of A are:

$\lambda_1 = \; -\; 3$

$\lambda_2 = \frac{-7*\sqrt{10}}{5}$

$\lambda_3 = \frac{7*\sqrt{10}}{5}$

From this, we deduce that the linearized system is unstable, because matrix A has a positive eigenvalue. We saw in lecture that a positive eigenvalue for matrix A implies unstable system.
Question: Give the physical intuition behind your finding that the magnetic levitation system is stable or unstable

Answer:
Intuitively, we expect that the system should be stable, given the model. First, observe that downwards force due to gravity is constant, so it suffices to only consider upwards force F, the squared ratio between current i and output y. When we run an impulse function as u, we get a sudden jump in the current i, but y is constant. This means that F becomes larger than at equilibrium, so the ball moves towards the magnet. As this happens y will accelerate as magnitude of y^2 is less than at equilibrium and since we can assume that eventually, current reaches steady state again. Eventually, we would have the ball attached to the magnet, ie y = 0.

If we were to allow y to become negative, which our model allows, then once the magnitude of y becomes large enough again, the force F becomes smaller and smaller as F is inversely proportional to square of y. So, gravity will win this “tug of war” and bring it back down. However, when we linearized our system, we obtained the following formula for y’’:

$$y'' = 2g(y - y^{\cdot}) - 2 \cdot \sqrt{(g)} \cdot i$$

For small perturbations of x or u, this makes sense. For example, increasing i while keeping y would mean y’’ < 0, so we accelerate in the negative direction (i.e move towards magnet). This makes perfect sense, as i^2 is in the numerator of expression for F.

But, in our model, as y decreases (while i goes back to steady state, say), then y-y^* makes y’’<0, so we move towards the magnet. As we move towards the magnet, y becomes more negative, so y-y^* becomes more and more negative and thus upwards force F becomes stronger and stronger (since F = Mg - My’’, and Mg is constant). So, we end up in an endless loop because our linearization of the model doesn’t capture that this “loop” only holds for values of y that are positive.

From our graph, we see we reach values of y like -600. In reality, when this happens, F should be very small because F is proportional to the inverse square of magnitude of y, which is huge in this case. But for constant i in our linear model, upwards force F depends on the difference between equilibrium y and y, so we accelerate towards negative infinity.

Question: Write down the transfer function G(s) and the pole(s) and zero(s).

Answer: Our MATLAB code gives the following as the transfer function, poles and zeros:

Transfer function G(s):
G =
    -6.261

----
s^3 + 3 s^2 - 19.6 s - 58.8
Continuous-time transfer function.

Zeros (z) and poles (p) of G(s):
z =

0×1 empty double column vector

p =

4.4272
-4.4272
-3.0000

The zeros being a 0x1 empty double column vector means there are no zeros.

Question: Plot the impulse response and discuss whether it is the same one you are expected in your prep.

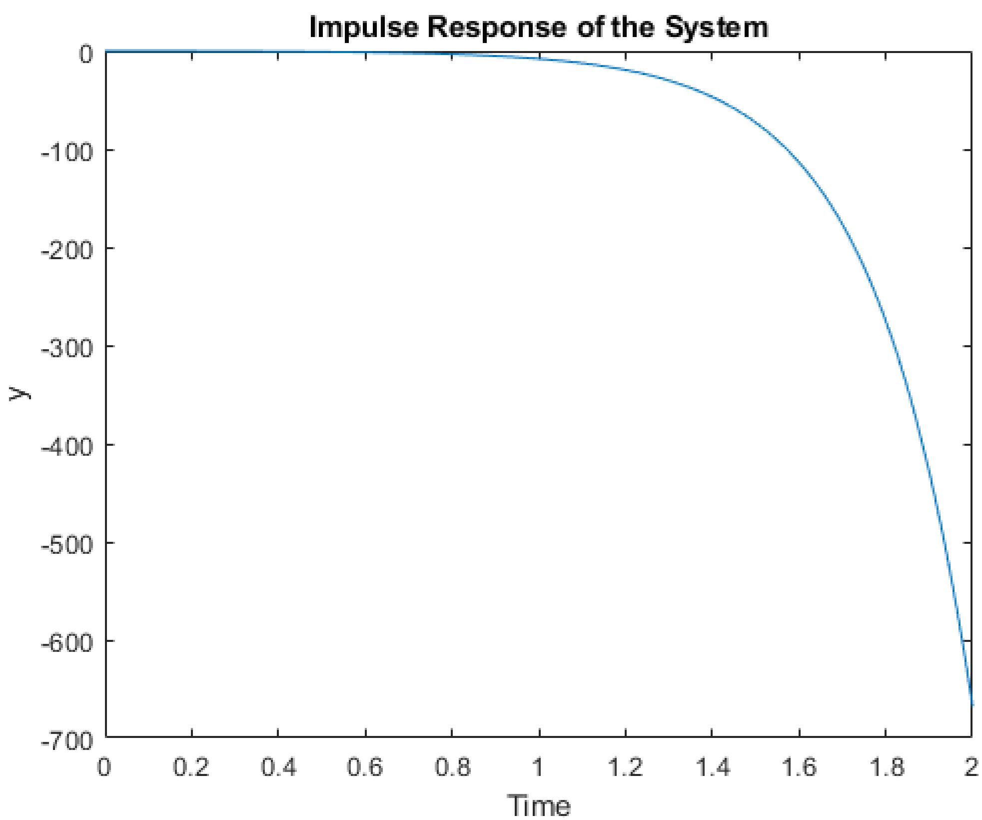


Figure 4: Impulse response of the linearized system

Answer: This is the same as the one we expected in our prep. As time goes to infinity, our output goes towards negative infinity.

Question: Plot the step response by using the second way (Part (b)). Discuss if the result is the same as the one obtained earlier in Part (a).

Answer:
The output of MATLAB step command is:

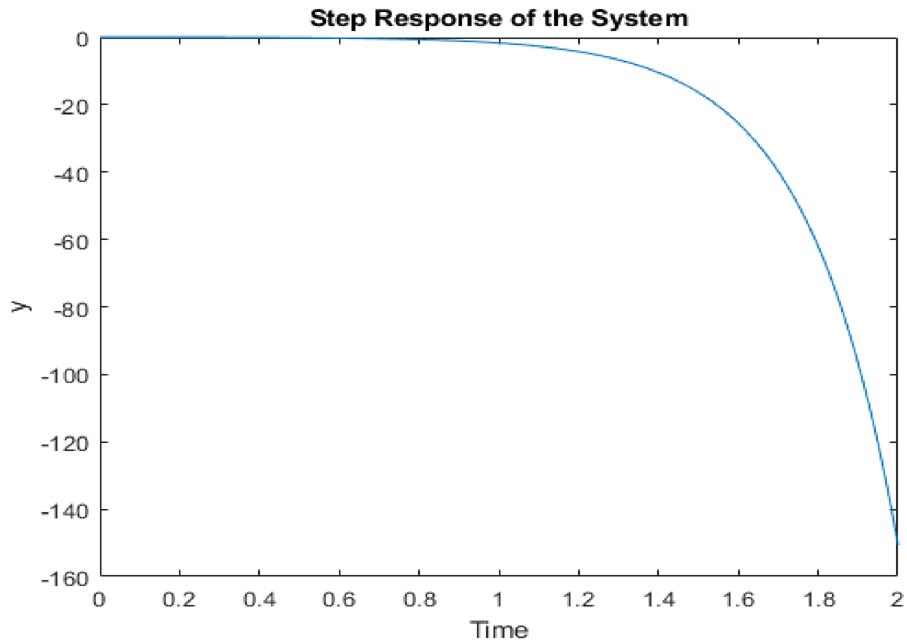


Figure 5: Step response of the linearized system

The SIMULINK model constructed as described in part b is shown below:

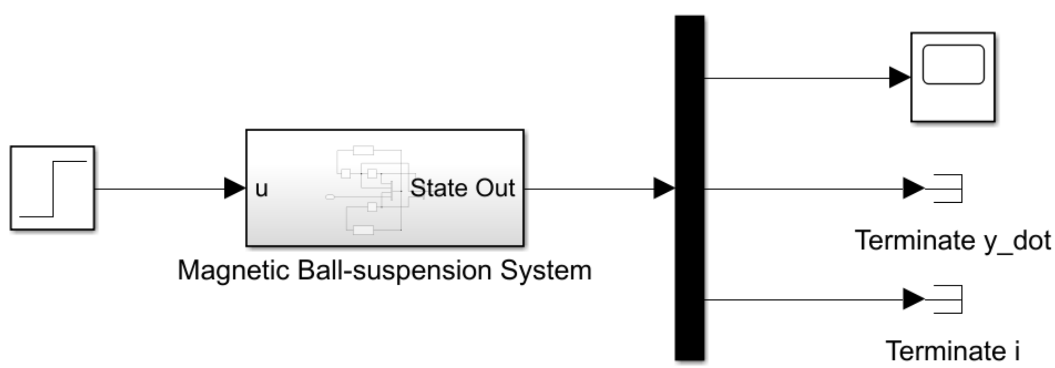


Figure 6: High-level linearized model

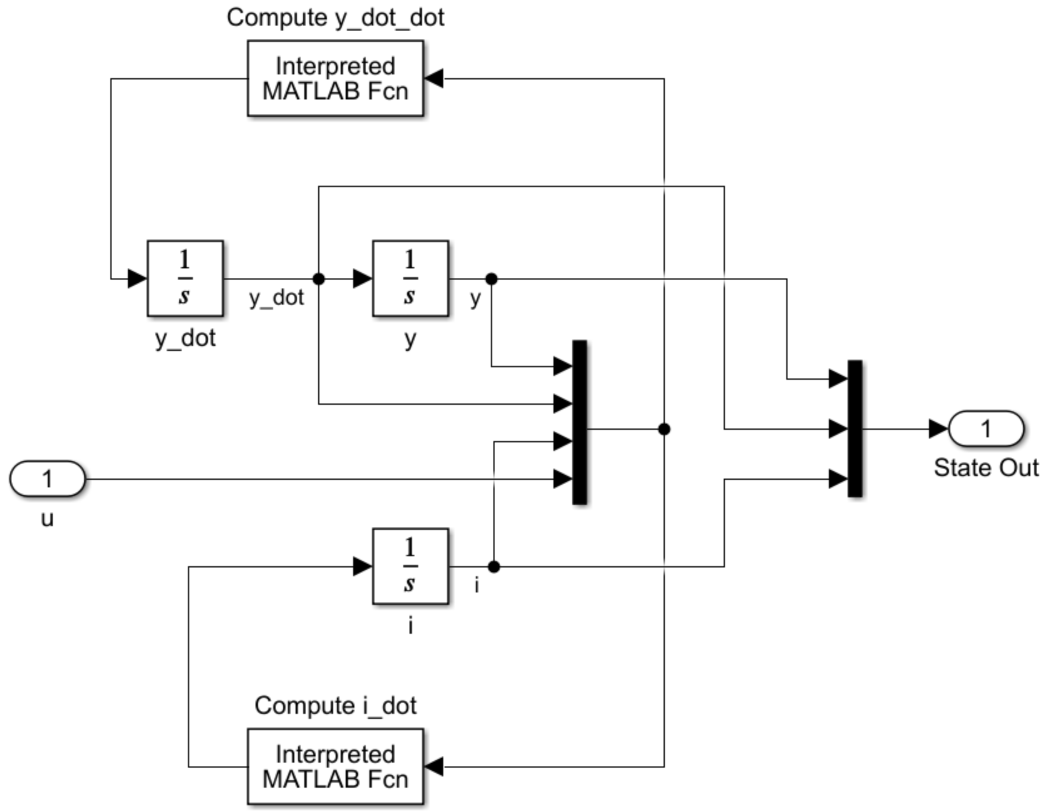


Figure 7: Sublevel system for linearized model

The output of the scope is shown below:

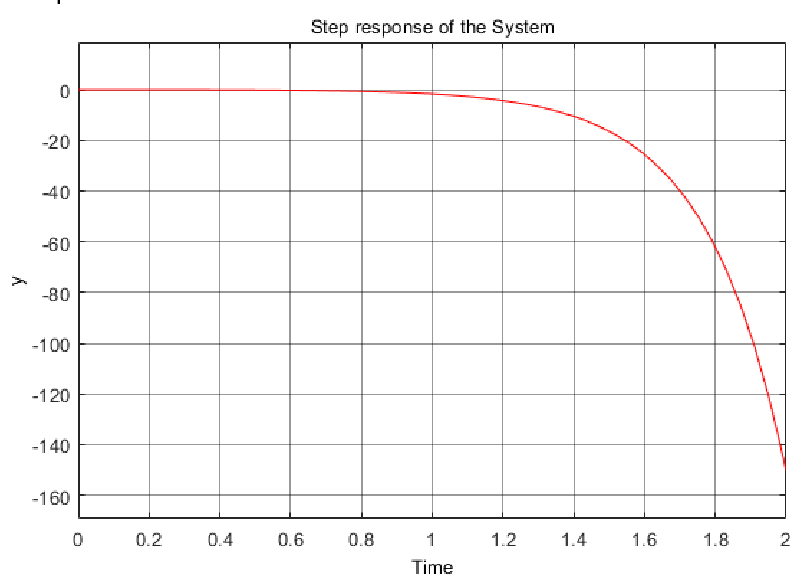


Figure 8: Step response of the linearized system

We can verify visually that the results for part b (SIMULINK) are the same as part a (MATLAB).

4. Summary or Conclusion

In this lab, we got to familiarize ourselves with the magnetic ball-suspension system. During the preparation, we were able to apply knowledge from class regarding linearization, state-space to transfer function and definition of stability to find equilibrium points of the magnetic ball system, and subsequently linearize the system and find a transfer function representation for it. This allowed us to determine the impulse response and make definitive conclusions on the stability of the linearized system.

We used MATLAB and SIMULINK models to verify our lab preparation. All our results were verified by the MATLAB code, and the outputs of our SIMULINK models also satisfied our results from preparation. Intuitive analysis of the stability of the magnetic-ball system exposed limitations of linearization in this problem, and linearization in general. Further, we realized the importance of model assumptions. For example, in this case, imposing y greater or equal to 0 would be useful to impose to our otherwise unaware mathematical model that the ball cannot go through the electromagnet.

Overall, this lab was a good way to apply concepts taught in lectures, as well as build essential skills in popular industry software, namely MATLAB and simulink. This lab in particular demonstrated the usefulness of the two software in *linearization*, which has always been presented as something that is done very often in practice.