Date: 2/27/2024
Submission Date:
Samson Wu
Joaquin Arcilla

# ECE356S
# Lab 2 Report PRA 2, Group 3

## Introduction:

The purpose of this lab is to simulate a magnetic ball-suspension system using MATLAB and SIMULINK. We first form the equations of motion for the model, giving a non-linear model. We then linearize the model to get the transfer function. The impulse and step responses are simulated in MATLAB and SIMULINK, and illustrate that the linearized system is unstable.

This lab will help in later more complicated projects where doing the math by hand is impractical and tedious. By practicing simulation on a smaller model, we ensure accuracy for the more complicated examples and projects.

Lab Preparation:

Lab 2 Pre-lab Group 3 Feb 27 Samson Wu

1.

$\downarrow y$ +

Newton's 2nd Law

$$Mg - F = Ma \Rightarrow Mg - \frac{i^2}{y^2} = M\ddot{y} \Rightarrow \ddot{y} = g - \frac{i^2}{My^2} = \frac{My^2 g - i^2}{My^2}$$

Circuit:

KVL: $u - Ri - L\frac{di}{dt} = 0$

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

$$\dot{x} = \begin{bmatrix} \dot{y} \\ \ddot{y} \\ di/dt \end{bmatrix} = \begin{bmatrix} x_2 \\ g - i^2/My^2 \\ \frac{u}{L} - \frac{Ri}{L} \end{bmatrix}$$

$$y = [1 \ 0 \ 0]x$$

$\boxed{2}$ equilibrium conditions:

$$f(x^*, u^*) = 0$$

① $\dot{y} = 0$

② $g - \frac{i^2}{My^2} = 0 \Rightarrow g = \frac{i^2}{My^2} \Rightarrow Mgy^2 = i^2 \Rightarrow \boxed{i^* = \pm\sqrt{Mg}\,y^*}$

③ $\frac{u}{L} - \frac{Ri}{L} = 0 \Rightarrow u^* = Ri^* \Rightarrow u^* = R(\sqrt{Mg}\,y^*)$

positive root $\boxed{i^* = \sqrt{Mg}\,y^*}$

$$x^* = \begin{bmatrix} 1 \\ 0 \\ \sqrt{Mg} \end{bmatrix} \qquad u^* = R\sqrt{Mg}\,y^*$$

$$\begin{aligned} \frac{d}{dt}(\delta x) &= A\delta x + B\delta u \\ \delta y &= C\delta x + D\delta u. \end{aligned}$$

let $\delta x = x - x^*$

$\delta y = y - y^*$

$\delta u = u - u^*$

$$\delta \dot{x} \approx \frac{\partial f}{\partial x}(x^*, u^*)\,\delta x + \frac{\partial f}{\partial u}(x^*, u^*)\,\delta u$$

$$\delta y \approx \frac{\partial h}{\partial x}(x^*, u^*) + \frac{\partial h}{\partial u}(x^*, u^*)\,\delta u$$

$$f(x,u) = \begin{bmatrix} x_2 \\ g - \frac{x_3^2}{M x_1^2} \\ \frac{u}{L} - \frac{R x_3}{L} \end{bmatrix} \qquad -c\,x_1^{-2} \Rightarrow 2c\,x_1^{-3}$$

$$\frac{\partial f}{\partial x}(x^*, u^*) = \begin{bmatrix} 0 & 1 & 0 \\ \frac{2x_3^2}{M x_1^3} & 0 & -\frac{2x_3}{M x_1^2} \\ 0 & 0 & -\frac{R}{L} \end{bmatrix} = A$$

$$\frac{\partial f}{\partial u}(x^*, u^*) = \begin{bmatrix} 0 \\ 0 \\ 1/L \end{bmatrix} = B \qquad h(x,u) = x_1$$

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

$$\frac{\partial h}{\partial u}(x^*, u^*) = 0 = D$$

3. Set $y^* = 1$, and derive the open-loop transfer function $G(s) = \delta Y(s)/\delta U(s)$ from the input $\delta u$ to the output $\delta y$ of this linearized system.

4. Find the corresponding impulse-response function $g(t) = \mathcal{L}^{-1}(G(s))$ of this linearized system and plot it.

$\boxed{3}$ $y^* = 1 \longrightarrow x_1^* = y^* = 1$

$$\frac{d}{dt}\delta x = A\delta x + B\delta u \quad , \quad \delta y = y - y^* = C\delta x + D\delta u$$

$$\frac{d}{dt}(\delta x) = \begin{bmatrix} 0 & 1 & 0 \\ \frac{2x_3^{*2}}{M} & 0 & \frac{-2x_3^*}{M} \\ 0 & 0 & -\frac{R}{L} \end{bmatrix} \delta x + \begin{bmatrix} 0 \\ 0 \\ 1/L \end{bmatrix} \delta u$$

$$\delta x = x - x^* = \begin{bmatrix} y - 1 \\ \dot{y} - \dot{y}^* \\ i - i^* \end{bmatrix} \qquad \frac{di}{dt} = \frac{u}{L} - \frac{Ri}{L}$$

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

$$sI - A = \begin{bmatrix} s & -1 & 0 \\ -\frac{2x_3^{*2}}{M} & s & \frac{2x_3^*}{M} \\ 0 & 0 & s + \frac{R}{L} \end{bmatrix}$$

$$\downarrow$$

$$G(s) = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} s & -1 & 0 \\ -\frac{2x_3^{*2}}{M} & s & \frac{2x_3^*}{M} \\ 0 & 0 & s + \frac{R}{L} \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ 0 \\ 1/L \end{bmatrix}$$

sub in:

$$x_3^{*2} = i^{*2} = Mgy^{*2} = 9.8$$

$R = 3$
$L = 1$
$M = 1$

$$= \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} s & -1 & 0 \\ -\frac{2\cdot 9.8}{1} & s & \frac{2\sqrt{9.8}}{1} \\ 0 & 0 & s + \frac{3}{1} \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ 0 \\ 1/1 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} s & -1 & 0 \\ -19.6 & s & 2\sqrt{9.8} \\ 0 & 0 & s+3 \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

$$\det(sI-A) = s\begin{vmatrix} s & 19.6 \\ 0 & s+3 \end{vmatrix} - (-1)\begin{vmatrix} -19.6 & 19.6 \\ 0 & s+3 \end{vmatrix} = s^2(s+3) + (-19.6(s+3))$$
$$= (s+3)(s^2-19.6)$$

$$\operatorname{adj}(sI-A) = \begin{bmatrix} s^2+3s & -(-19.6(s+3)) & -2\sqrt{9.8} \\ -(s+3) & s^2+3s & 0 \\ 2 & -(2s) & s^2-19.6 \end{bmatrix}$$

$$\Rightarrow G(s) = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \frac{1}{(s+3)(s^2-19.6)} \begin{bmatrix} s^2+3s & 19.6(s+3) & -2\sqrt{9.8} \\ -s-3 & s^2+3s & 0 \\ 2 & -2s & s^2-19.6 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

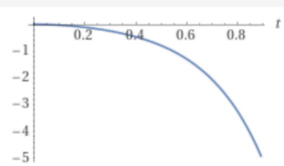
$$= \frac{1}{(s+3)(s^2-19.6)} \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} -2\sqrt{9.8} \\ 0 \\ s^2+19.6 \end{bmatrix} = \frac{-6.26}{(s+3)(s^2-19.6)}$$

$\boxed{4}$ $g(t) = \mathcal{L}^{-1}\{G(s)\} =$

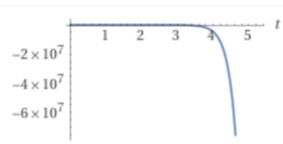
Result

$$-6.26\left(0.0791335\, e^{-4.42719\, t} - 0.0943396\, e^{-3\, t} + 0.0152061\, e^{4.42719\, t}\right)$$

Plots




(t from 0 to 0.9)




(t from 0 to 5.4)

Alternate forms

Experiment:

**3.1 Building the SIMULINK Model**

Write down the two Fcn functions in the space below:

$\frac{d^2y}{dt^2} = g - \frac{i^2}{M*y^2}$ OR y_dot_dot = G - u(3)^2/(M * u(1)^2);

$\frac{di}{dt} = \frac{u}{L} - \frac{iR}{L}$ OR di_dt = u(4)/L -(u(3)*R)/L;

**3.2 Linearizing the Model in MATLAB**

*Write down numerical values of the equilibrium point (, ), corresponding to the constant position set f(x*, u*) = 0 to find equilibrium points:*

x* = [1; 0; sqrt(9.8)]
u* = [3*sqrt(9.8)]

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

A = [0 1 0; 19.6 0 -6.26; 0 0 -3];
B = [0;0;1];

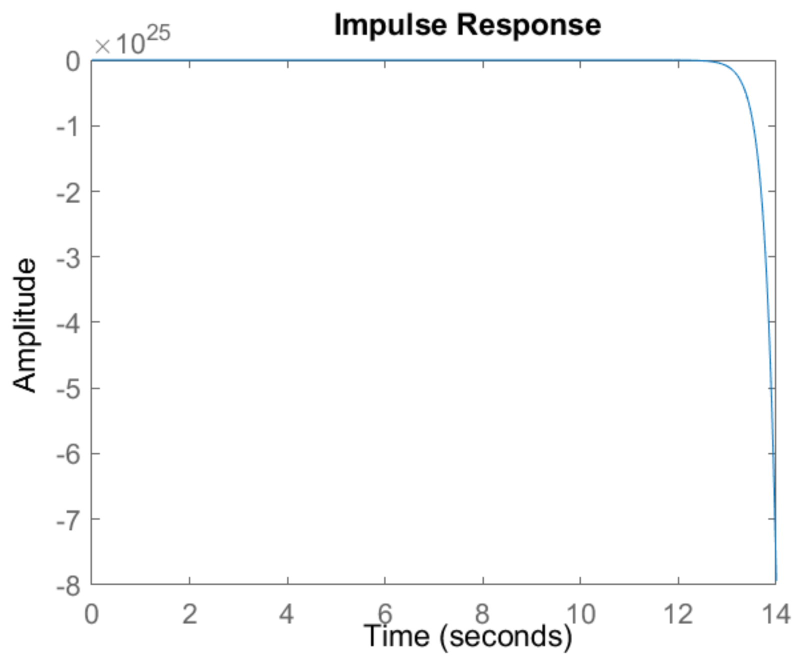
Eigenvalues of A are:
4.4272, -4.4272, -3

*Is the system stable or unstable based on the eigenvalues?*
Since one of the eigenvalues is positive, the maglev system is unstable.

*Give the physical intuition behind your finding that the magnetic levitation system is stable or unstable.*
This is intuitively known because the magnetic force is constant because the inductor acts like a short circuit, assuming u is a step function. If the ball is in a state of equilibrium, one little nudge should either send it up or down.

*Write down the transfer function G(s) and the pole(s) and zero(s).*
G(s) = - 6.26 / (s+3)(s^2 - 19.6) or -6.26 / (s^3 + 3*s^2 -19.6*s - 58)
There are no zeros.
There are 3 poles, one positive and 2 negative: -3, 4.42, 4.42.

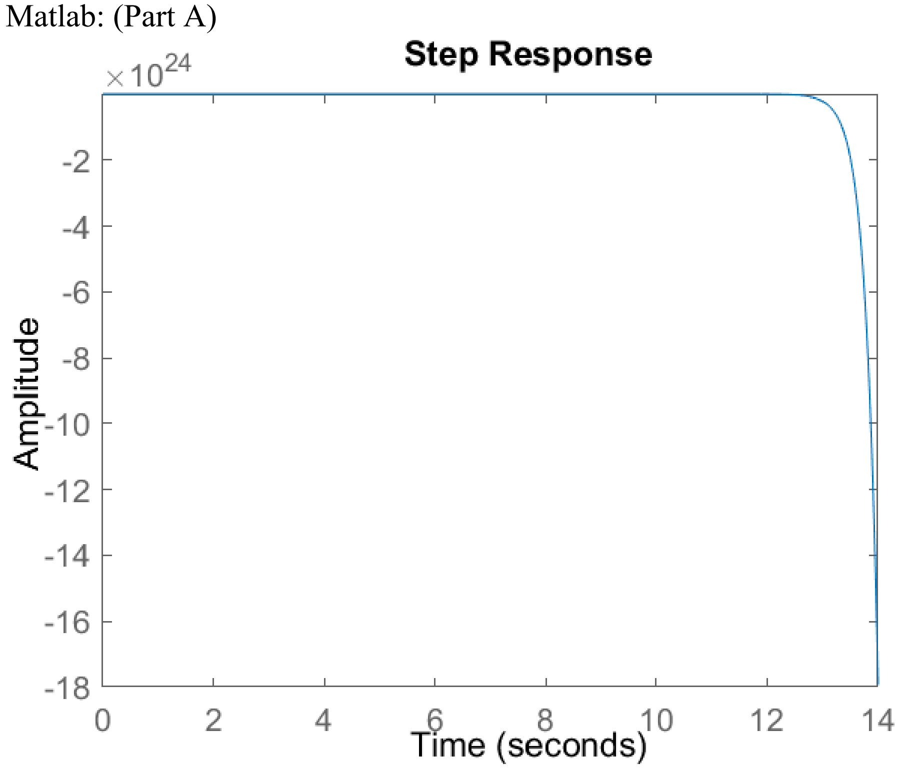
*Plot the impulse response and discuss whether it is the same one you are expecting in your prep.*



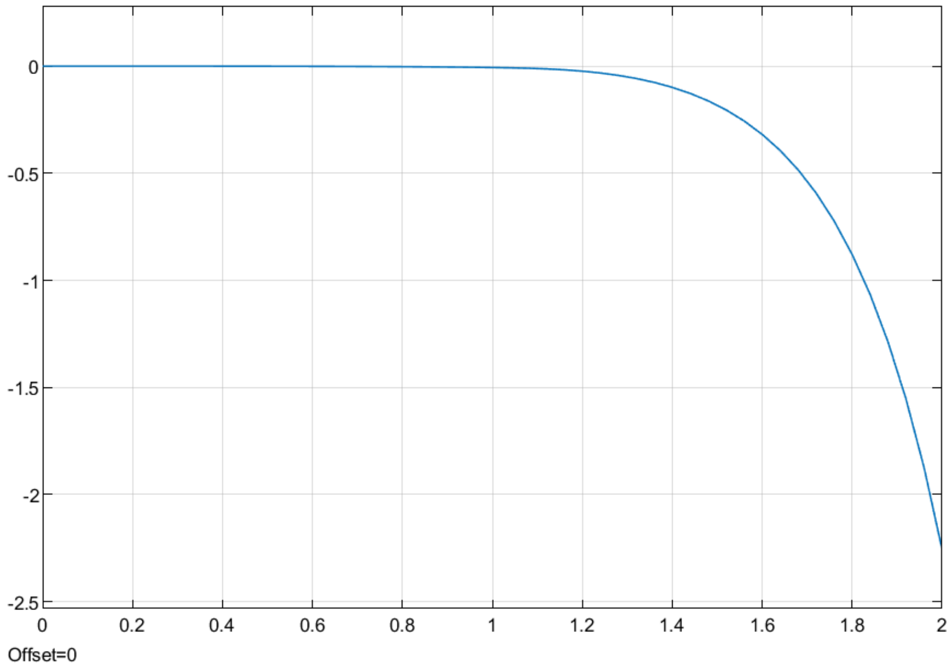

This is the same thing that we are expecting based on the pre-lab work.

*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).*

Matlab: (Part A)



Simulink: (Part B)




Both displays of the step response are the same and quickly fall soon after the impulse. This makes sense since both methods are basically doing the exact same thing, just one with the Matlab model and one with the Simulink model.

**Conclusion**

In this lab, we successfully simulated a magnetic ball-suspension system using MATLAB and SIMULINK. We derived the non-linear equations of motion and then linearized the model to obtain the transfer function. Through simulations, we observed that the linearized system exhibits instability, evident in the impulse and step responses.

This exercise serves as a valuable stepping stone for future projects involving complex systems. By practicing simulation on a manageable model like this, we gain the necessary skills and confidence to tackle intricate projects where manual calculations become impractical and error-prone. This lab, therefore, equips us with the foundation to approach complex systems with analytical and simulation-based methods.

Signatures:

Their input is $\ddot{y}$, and their output is $y$. The input of these two integrators is computed using a function block $f(\cdot)$, found in the `User-Defined Functions` library. This function takes as input the output of the multiplexer, which is the vector $[y \;\; \dot{y} \;\; i \;\; u]\top = (x, u)$. Summarizing, the upper part of the block diagram in Figure 3 implements an equation $\ddot{y} = f(x, u)$. you will have to insert the correct function $f$ according to your mathematical model. Now look at the single integrator block in the lower part of the diagram. Its output is $i$, and its input is $di/dt$. What feeds this integrator is the output of another function block $f(x, u)$ (this $f$ is different than then one above). Again, you will have to put the appropriate function in this block. Save your complete model in SIMULINK as `magball`.

Have the TA sign here if your model is correct. JM

### 3.2 Linearizing the Model in MATLAB

1. For the real magnetic ball-suspension system, we have the following physical parameters:

   - g = 9.8 $m/s^2$
   - M = 1 Kg
   - R = 3 Ohm
   - L = 1 H

   Substituting in the values of the physical parameters, setting the equilibrium position $y^\star$ to be $y^\star = 1$, and based on the linearized state equation obtained in your preparation, enter the linearized system matrices $A$ and $B$ in the Matlab workspace.

2. Using the Matlab command `eig`, determine the numerical values of the eigenvalues of $A$. We will see later in this course that if all eigenvalues of $A$ have negative real part, then the linearized system is *stable*. For the linearized maglev system, stability means that for any initial condition, $(y(t), \dot{y}(t), i(t)) \to (y^\star, \dot{y}^\star, i^\star)$ as $t \to \infty$. Vice versa, if $A$ has at least one eigenvalue with positive real part, then the linearized maglev system is *unstable*, meaning that there are initial conditions such that the triple $(y(t), \dot{y}(t), i(t))$ does not converge to the equilibrium $(y^\star, \dot{y}^\star, i^\star)$.

   Look at the eigenvalues you just found: is the linearized system stable or unstable? Discuss with your lab partner the physical intuition behind the result (i.e., why the system is stable or unstable). You'll include a discussion on this in your lab report.

   Using the Matlab command `ss2tf`, use your matrices $(A, B, C, D)$ to find the transfer function of the linearized model. Compare it with the transfer function you found in your preparation and check that they coincide.

   Find the poles and zeros of $G(s)$.

   Have the TA check your work and sign here. JM

3. Now you will learn how to use Simulink to automatically linearize nonlinear models, and you will compare the result to your own computations. Using the Matlab command `linmod`, you will linearize your system at the equilibrium point $(x^\star, u^\star)$. First you need to understand the ordering of states in Simulink. For that, issue the command

   ```
   >> [sizes, x0, states]=magball
   ```

   Look at the `states` array. The ordering of the strings in this array coincides with the ordering of states used by Simulink. The string names are the labels you have used under each integrator block $(1/s)$. Now use a command of the form `[A,B,C,D]=linmod('magball',xstar,ustar)` to generate the $(A, B, C, D)$ matrices of the linearization at the equilibrium condition $(x^\star, u^\star)$ found in part 2 of your prep, setting $y^\star = 1$. Check that the matrices $(A, B, C, D)$, other than a possible re-ordering of the states, are the same as those obtained earlier in Step 1.

   Have the TA check your work and sign here. JM

4. Use the MATLAB command `impulse` to obtain the impulse response and check its appearance vs the result in your preparation.

5. Use the following two ways to obtain the step response (that is, the plot of the output signal when the input fed to the system is the unit step).

   (a) Use the MATLAB command **step**.

   (b) Modify the Fcn blocks in Fig.3 to get the linearized model (as in Fig.4). The two equations in Fcn blocks are corresponding to the $A$ and $B$ matrices you have obtained in Step 3. Save the new SIMULINK model as `magball_linear`. Then use Step block as the input source and place a Scope block to display the position $y$ during the simulation. Set the simulation stop

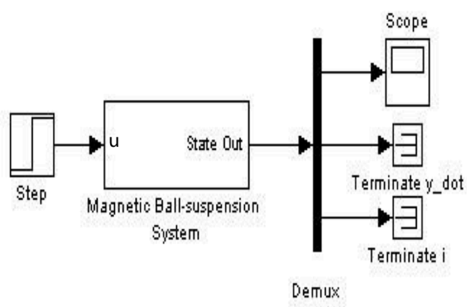



Figure 4: Linearized SIMULINK Model

   time to be 2(sec) and run the simulation.

   Check that the result is the same as the one obtained earlier in part (a).

   Have the TA check your work and sign here.

## 4 Report

Please submit this document with all TA signatures stapled together with your lab report. For the lab report, please follow the format instructions for Lab 1 provided on the course website.

## Appendix

The following is a list of MATLAB commands which may be useful for completing this lab. You may also refer to the MATLAB Tutorial Handout on the course website (under the folder "general handouts") or the reference library on the Mathworks website for more MATLAB functions and the SIMULINK usage.

- **eig** - Determines the eigenvalues of a matrix
- **help** - Access to help on MATLAB commands. e.g. **help plot**
- **impulse** - Simulates an impulse response for a LTI system. e.g. **impulse(sys)**
- **legend** - Creates a legend for a plot. e.g. **legend('Reference Step','Position')**