MAE 3260 Final Group Work: Exploring a System of Interest

Report

Outline (two examples given below, there could be others):

Option #1: one report, same grade for all	Option #2: separate sections/grades for each
Page 1: cover page **Pages 2-9**: any format you want in these pages, just include a few section headings in this outline) **Page 10:** References	**Page 1:** cover page **Page 2-3:** Student A technical summary **Page 4-5:** Student B technical summary **Page 6-7:** Student C technical summary **Page 8-9:** Student D technical summary **Page 10:** References

Title: Shocking Discoveries Surrounding Shocks

Topic of Interest: Modeling of a Shock as a Spring Mass Damper

Abstract: One paragraph summarizing the project. Include the system, why you decided on this project, what aspects you plan to study, etc. See the list below or the provided project description examples for some ideas of possible aspects to study. Choose the aspects that make the most sense for your system.

For our final groupwork, our group opted to dissect and analyze a shock. Over the course of the semester, we have gone over the behavior of similar systems extensively, including modeling suspension systems and highlighting the behaviors of multiple SMD systems. As a result, selecting a shock system allowed us to apply our accumulated system dynamics comprehension to a highly applicable project. Additionally, the members of our team are all extremely interested in cars and their behaviors, so selecting the shock allowed us to apply system dynamics concepts to outside interests. Because shocks are used in cars and other common systems, this project allowed us to do a deep dive into understanding the math and system topics that define such a common design, applying the lens of system dynamics to a real world, fascinating part of everyday life.

Students/Roles:

Student	Task/Role	Portfolio
Tam	Block Diagram of System and State Space Model	https://cornell-mae-ug.github.io/fa25-portfolio-tampham321/projects/Friction%20Damper%20Analysis/
Isaac	System modeling and ODEs	https://cornell-mae-ug.github.io/fa25-portfolio-IsaacNg1/projects/2025_ShockAbsorber/
Theresa	Matlab and qualitative analysis	

Nicholas	Dissection analysis	https://cornell-mae-ug.github.io/fa25-portfolio-NicholasAGavalas/projects/MAE%203260%20Design/

List of MAE 3260 concepts or skills used in this group work:

- System Modelling
- Block Diagrams
- State Space Modeling

Pages 2-9 include two pages for each student:

- Summarize technical work, including figures
- For references, use numbers/brackets (e.g. [1]) in text with list - IEEE citation format.
- Students can combine section (i.e. 4 pages for students A and B), but will get the same grade

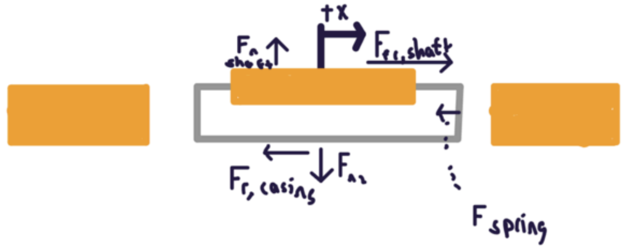
I.) Physical Dissection analysis and correlation to System Dynamics (Nicholas Nag72)

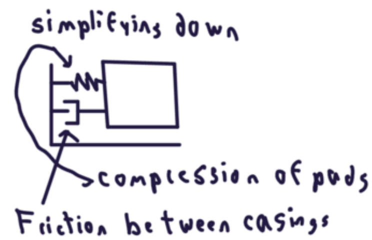
This basic sketch of our system gives us a few key components. But to start analyzing our system, we need to make some assumptions:

1.) We will be modeling all our system components in 1 dimension
    a.) No poissons ratio
    b.) No off axis motion
    c.) No moment contributions
2.) We will be using a Coulomb friction model
3.) We will be treating the sliding casing and outer casing as completely rigid materials.
4.) We will be ignoring finite effects (i.e. any effective ranges of motion)

This gives us the following within our operating range of acceleration, i.e., we will not be exceeding the coefficient of friction between the friction pad and the sliding metal rod (or we would have just a friction damper system.

This means that we can do a sum of the forces on the sliding plastic casing instead of the metal rod (we are assuming they are moving together)
This gives us the following model:





We get the following system of equations
We start with our sum of forces in the y to find the normal forces contributing to our frictional component

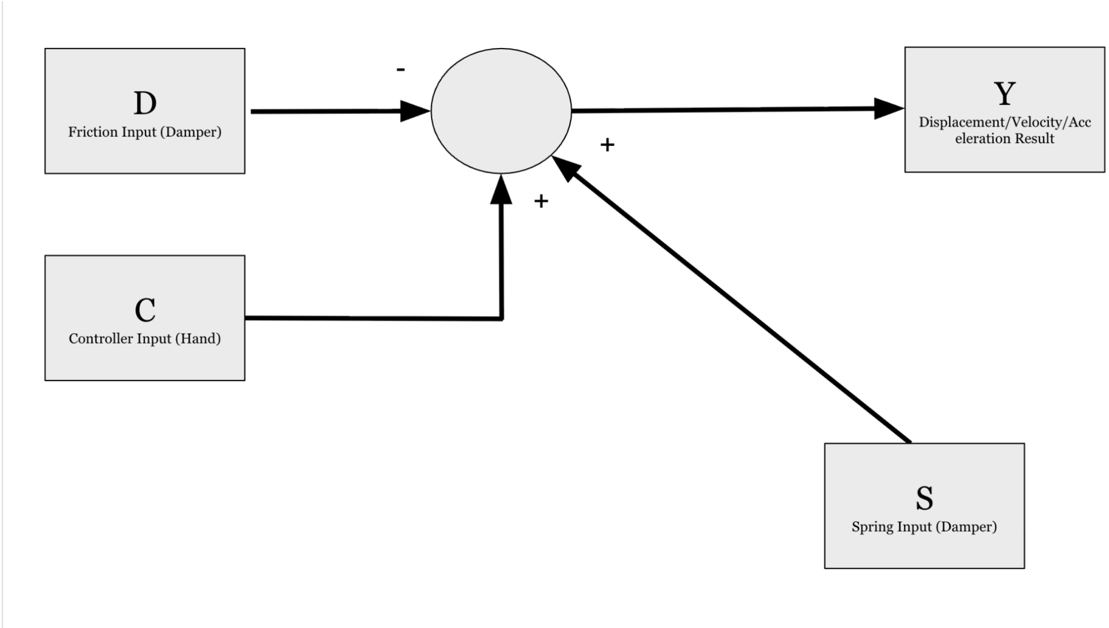
$$\Sigma F_y = 0$$
$$F_{N, shaft} = F_{N, Casing}$$
$$F_{N, casing} = E \times A_H \times (\frac{\delta}{T})$$

We can now take a sum of the forces in the x direction,

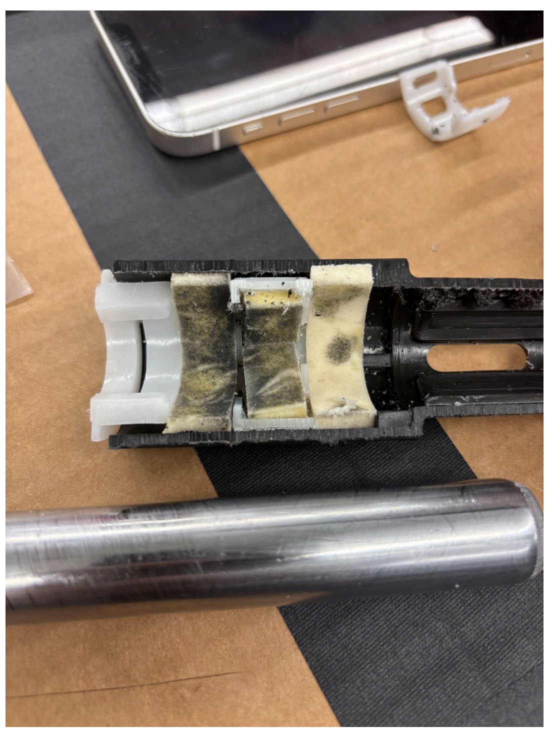
$$\Sigma F_x = (M_{shaft} + M_{slider})x''$$
$$\Sigma F_x = F_{spring} + F_{Friction}$$
$$F_{spring} = E \times A_V \times (\frac{X}{L})$$
$$F_{friction} = \mu * F_{N, casing}$$
$$F_{friction} = \mu \times E \times A_H \times (\frac{\delta}{T})$$
$$\Sigma F_x = E \times A \times (\frac{x}{L}) + \mu \times E \times A_H \times (\frac{\delta}{T})$$
$$(M_{shaft} + M_{slider})x'' = E \times A \times (\frac{x}{L}) + \mu \times E \times A_H \times (\frac{\delta}{T})$$
$$x'' = E \times A \times (\frac{x}{L}) + \mu \times E \times A_H \times (\frac{\delta}{T})/(M_{shaft} + M_{slider})$$

As for our mechanical parameters, we have:
We have the following parameters that we assume:
E = 120 x 10^3
μ = .289
We have the following parameters that we can measure:
δ = 0.005
A_V = 0.00005 m^2
A_H = 0.0001 m^2
M_{shaft} = .200
M_{slider} = .018
L = 0.02m
T = 0.01 m

2) Block Diagram & State Space Model of System (Tam Pham thp28)



Although the friction damper does not contain a visible spring, the rubbery foam that rests on the surface of the hydraulic cylinder acts as both a damper and the spring. The foam in a bent position provides elastic potential energy that allows the spring-mass-damper system to return to its original position. The plastic casing (1) has a slightly raised rubbery foam that provides compression for elasticity and increased normal force for friction.



The surface of the rubbery foam has a high coefficient of friction which serves as a damper, slowing down its acceleration via the controller input. In the photo, as the cylinder moves from right to left, the middle rubber foam gets **compressed axially**, resulting in a spring force that points left to right, which allows the cylinder to return to its original position because of a restoring force.

The system is best **modeled as an open-loop system**, there is no feedback mechanism that allows you to control or know how far or fast the system can be actuated. Breaking down the shock absorber down into a spring-mass-damper system allows us to use a substantial amount of class material in order to create a relatively simple state space model to understand our inputs and outputs.

2) State Space Model of a Spring-Mass-Damper (Tam Pham thp28)

In order to model a state space, we need to figure out x(t), y(t) and u(t).

x(t), How many states to model the spring-mass damper?
Answer: Position and Velocity, x_1 and x_2 respectively.
y(t), What can we quantify and measure?*
Answer: Position only.
*y(t) is subject to change if we had not dissected the friction absorber, as we could use a force reader to calculate the acceleration, which will allow us to get velocity.
u(t), What are your inputs?
Answer: Force via hand is the only input.

$$\dot{x} = Ax + Bu$$
$$y = Cx + Du$$

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

$$\dot{x} = \begin{bmatrix} \dot{x} \\ \ddot{x} \end{bmatrix} = \begin{bmatrix} x_2 \\ \dot{x}_2 \end{bmatrix}$$

$$\begin{bmatrix} x_2 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ \frac{k}{m} & -\frac{b}{m} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1/m \end{bmatrix} \begin{bmatrix} u \end{bmatrix}$$

$$\begin{bmatrix} y \end{bmatrix} = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix} \begin{bmatrix} u \end{bmatrix}$$

3) System Modelling with Washing Machine Mass and Forcing Included (Isaac Ng in63)

Components of a Washing Machine Suspension:

1. The effective mass from the drum and wet clothes
2. The Shock Absorbers acting as spring-damper with friction
   a. Spring force from the compressed foam expanding
   b. Coulomb friction force from the slider rubbing on the foam
3. Disturbance input from the unbalanced clothes or controller input from applied force

Modelling the components:

1. m = mass from drum, wet clothes, and water
2.
   a. $F_{spring} = EA_V(\frac{X}{L})$
   b. $F_{friction} = \mu EA_H(\frac{\delta}{T})$
3.
   a. $F_d = m_u r\omega^2 \cos(\omega t)$ where m_u is the unbalanced mass
   b. $F_c = F_{Applied}$

Final ODE:

$$m\ddot{x} + F_s + F_f = F_d(t)$$

$$m\ddot{x} + EA_V(\frac{X}{L}) + \mu EA_H(\frac{\delta}{T}) = m_u r\omega^2 cos(\omega t)$$

The difference between the two ODEs comes from what each model includes. The section 1 shock absorber model only considers the shaft and slider, so the mass is $M_{shaft} + M_{slider}$, and the only forces acting on the system are the spring force and the friction force. This equation describes how the shock absorber behaves by itself.

In my model, the shock absorber is included as part of the full washing machine system. Because of this, the total mass also includes the drum, the clothes, and the water. The unbalanced rotation of the drum during the spin cycle also adds an external forcing term. As a result, the spring and friction act as resisting forces, while the unbalanced force acts as the input.

State-Space Modelling of the Shock Absorber with the Washing Machine included:

ODE:

$$m\ddot{x} + EA_v\left(\frac{x}{L}\right) + \mu EA_H\left(\frac{\delta}{T}\right) = m_u r \omega^2 \cos(\omega t)$$

$$\ddot{x} = -\frac{EA_v}{mL}x - \frac{\mu EA_H}{m}\left(\frac{\delta}{T}\right) + \frac{m_u r \omega^2}{m}\cos(\omega t)$$

State Variables:

$$x_1 = x \;,\; x_2 = \dot{x}$$

$$\dot{x}_1 = x_2$$

$$\dot{x}_2 = -\frac{EA_v}{mL}x_1 - \frac{\mu EA_H}{m}\left(\frac{\delta}{T}\right) + \frac{m_u r \omega^2}{m}\cos(\omega t)$$

Input Vector:

$$u(t) = \begin{bmatrix} \cos(\omega t) \\ 1 \end{bmatrix}$$

State Space:

$$\dot{x} = \begin{bmatrix} 0 & 1 \\ -\frac{EA_v}{mL} & 0 \end{bmatrix} x + \begin{bmatrix} 0 & 0 \\ \frac{m_u r \omega^2}{m} & -\frac{\mu EA_H}{m}\left(\frac{\delta}{T}\right) \end{bmatrix} u$$

Output:

$$y = \begin{bmatrix} 1 & 0 \end{bmatrix} x + \begin{bmatrix} 0 & 0 \end{bmatrix} u$$

The two state-space models are different because they come from different physical assumptions about the system. The standard form shown in the second section represents a classic SMD system with viscous damping.

My state space model is based on the washing machine shock absorber with friction and forcing from the unbalanced drum.

4) Matlab and analysis: Theresa Franklin

I chose to approach the modeling of the system using an in class matlab script as a reference (specifically the frequency_response simulation), my own matlab experience, and Google Gemini to supplement and correct any gaps in comprehension.

The matlab model is somewhat similar to the final animated spring-mass-damper system in the frequency_response assignment from earlier in the semester. As such, we used this as a guideline for the majority of this design, including how to animate and visualize the system and how to define operation. That being said, while the frequency_response simulation is qualitatively similar, there are inherent limitations to the applicability of the model. Using Nicholas Gavalas' mathematical representation from section one to model is the main inherent difference, as the governing equations that model our system are inherently different, even though they are both generally representing SMD systems. Additionally, the frequency_response simulation does not allow for modeling constant force applications such as the one we chose to model.

We chose to model a constant force application due to the way that we qualitatively tested during our section. While we had initially thought it would be productive to measure oscillation response (eg. with a shock dyno), due to time constraints, we instead tested by applying a vertical force with our hands and noting the response. Since we used this technique to get a quantitative response, we chose to match that in the model. That being said, the model is also able to handle sinusoidal inputs.

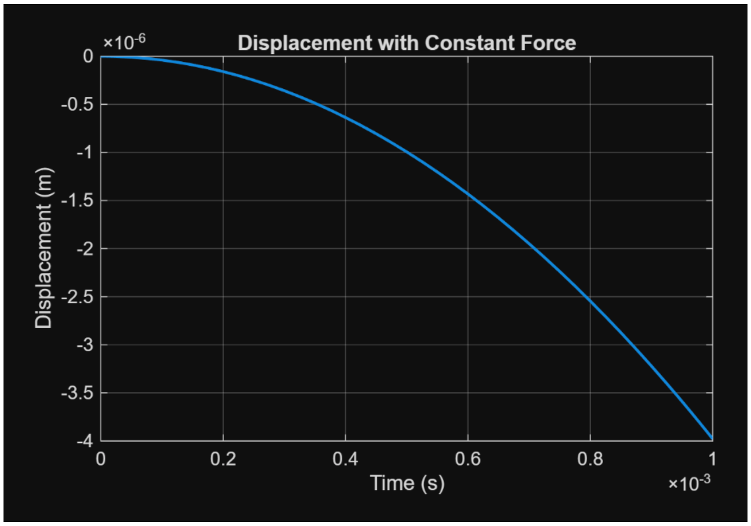
Overall, the matlab system is a simplified model of the physical system. It only allows for movement in the y direction, which assumes that the shaft is constrained to exclusively vertical displacement, which is not inherently true due to the soft, springy inner compartment of the damper (hence, it could move slightly left and right, thus receiving differing reaction forces to keep it aligned). Additionally, it does not include any aspects that limit range of motion, meaning that it does not accurately output responses that mimic the actual physical system if the inputs into the model result in movement past the limits of the physical range.

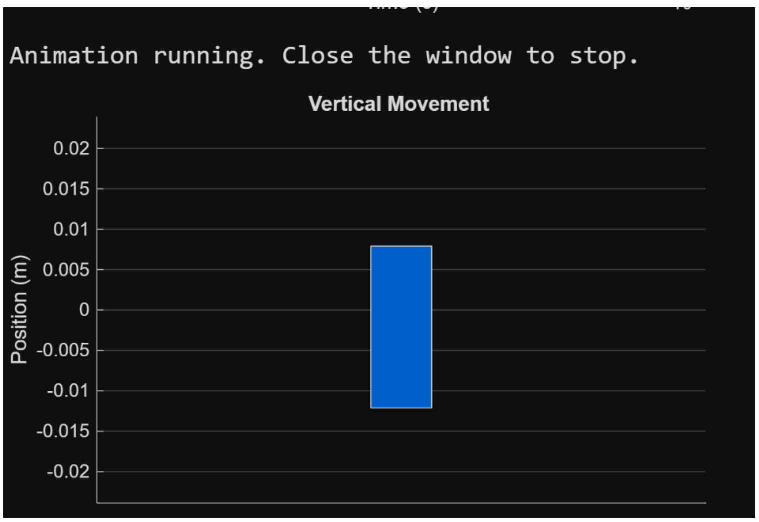
An example output is as follows, using the parameters calculated in Nicholas Gavalas's section, as well as his representation of the system acceleration:

$$x'' = E \times A \times (\frac{x}{L}) + \mu \times E \times A_H \times (\frac{\delta}{T})/(M_{shaft} + M_{slider})$$

Assumed parameters:
E = 120 x 103
μ = .289
Measured parameters:
δ = 0.005
AV = 0.00005 m^2
AH = 0.0001 m^2
Mshaft = .200

Mslider = .018
L = 0.02m
T = 0.01 m





Note that the second image is a screenshot of the animation - feel free to run the code to watch the animation move. Do note that it is programmed to repeat the animation over and over: do not confuse the looped animation with the behavior of the system. Overall, the output from the matlab closely resembles our quantitative analysis and experimentation from in classroom working with the physical structure, insinuating both the ODE and the numbers represented in Nicholas's section are accurate to the system dynamics.

```
clear; clc; close all;


%% PARAMETERS
```

```
E = 120e3;
mu = 0.289;

delta = 0.005;
L = 0.02;
T = 0.01;
Mshaft = 0.200;
Mslider = 0.018;
A_H = 0.0001;

M_total = Mshaft + Mslider;
A = A_H;

k_axial = (E * A) / L;
F_constant = -(mu * E * A_H * delta) / T;

tEnd = 0.001;
x0 = [0; 0];

massSize = 0.02;
scale = 5000;
dt_anim = 0.0001;

slowdownFactor = 10000;
pauseTime = dt_anim * slowdownFactor;

%% SOLUTION
[t,x] = ode45(@(t,x)
stateEq_AxialStiffness(t,x,M_total,k_axial,F_constant), [0 tEnd], x0);
```

```
xDisp = x(:,1);


%% PLOTTING

figure(1)

plot(t, xDisp, 'LineWidth', 1.5)

xlabel('Time (s)'); ylabel('Displacement (m)')

title('Displacement with Constant Force')

grid on


%% ANIMATION

figure(2)

clf; hold on; grid on

title('Vertical Movement ')

ylabel('Position (m)'); set(gca,'XTick',[])


max_disp_actual = max(abs(xDisp));

MAX_SCALED_MOVEMENT = 1.2 * scale * max_disp_actual;

range = min(MAX_SCALED_MOVEMENT, 2.0);

axis([-0.1 0.1 -range range])


mass = rectangle('Position',[-massSize/2,-massSize/2,massSize,massSize],
'FaceColor',[0.1 0.4 0.8]);


dt_sim = t(2)-t(1);

frameStep = max(1, round(dt_anim/dt_sim));


disp('Animation running. Close the window to stop.')


while ishandle(mass)
```

```
    for ti = 1:frameStep:length(t)

        ypos = scale * xDisp(ti);


        set(mass,'Position',[-massSize/2, ypos-massSize/2, massSize, 
massSize]);


        drawnow;

        pause(pauseTime);


        if ~ishandle(mass); break; end

    end

end

disp('Animation stopped.')


%% STATE EQUATIONS

function dx = stateEq_AxialStiffness(t,x,M,k,F_const)

    dx = zeros(2,1);

    dx(1) = x(2);


    dx(2) = (k*x(1) + F_const) / M;

end
```