MEEN 364 ***Spring 2018***

Assignment: Lab 2 & 3
Instructor: Dr. Prabhakar Pagilla
Section: 501
Submission Date: 19 February 2018

All authors have contributed to the preparation of the report and have read the final version of the report.

On my honor, as an Aggie, I have neither given nor received unauthorized aid on this academic work.

Name of Authors	**Signature**
Andres Contreras	
Jacob Hartzer	
Perry Inhofe	

I. Abstract

In this lab, a pendulum slider system was analyzed in order to better understand the modeling of dynamic systems using MatLab and Simulink software. Nonlinear and linear models were separately developed and simulated using both MatLab and Simulink and then compared to each other as well as an experimental measured response of the system. In the first part of the lab, the nonlinear and linear equations of motions were developed and simulated using MatLab and Simulink to predict the response of the system. It was shown that the simulations are identical in either programming environment and that under the small angle assumption, the linear system is a valid model. In the next part of the lab, the slider and pendulum position potentiometers were calibrated and data was collected after initiating motion in the system. The data was analyzed to find the coefficients of dry friction and viscous damping which were 0.0732 and 0.0968 $\frac{Ns}{m}$ respectively. Using these values to simulate the motion again on MatLab, the linear model, nonlinear model, and real response motions could be directly compared to determine the accuracy of the models. It was found that the coefficient of dry friction was verified, but the viscous damping coefficient had to be changed to 0.01 $\frac{Ns}{m}$ to produce a similar model to the experimental data. Therefore, it was shown that the methods used to calculate the coefficients are good for estimating, yet are not perfect. Additionally, the nonlinear model was shown to closely model the true to life experimental data that was collected in the lab, thus verifying the simulation techniques.

II. Introduction

To be able to properly model and understand dynamic systems, it is important to be able to solve the differential equations that govern their motion. Multi-degree of freedom systems are complex and difficult to model by hand, so computers are used to provide the numerical analysis needed to simulate their behaviors. The first part of the lab modeled a two-degree of freedom pendulum mounted on a slider using computer software assuming the pendulum had viscous damping and the slider had Coulomb damping. The second part of the lab involved experimentally determining the values for the viscous and Coulomb damping coefficients and comparing the measured response to the simulated response. The objectives of this lab were to learn how to use Matlab to simulate the behavior of dynamic systems and to observe the differences between the performance of linear and nonlinear models. Additionally, this lab explores the validity of the models used to simulate multi degree of freedom systems through comparison of simulated models to real-world experimental data.

III. Theory

The system to be dynamically modeled and experimentally analyzed in this experiment is a two-degree of freedom pendulum mounted on a slider, as shown in Figure 1:

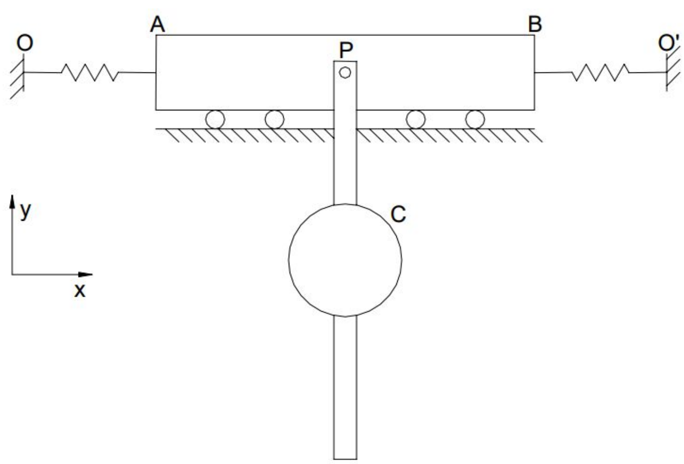

Figure 1. Pendulum-Slider System (drawn by Christopher Cullum)

The oscillations of the system are assumed to be characterized by viscous damping in the pendulum and Coulomb damping in the slider.

By constructing free body diagrams for both the pendulum and the slider, equations of motion for the system were able to be developed using kinematics and Newton's laws. The forces acting on the pendulum include gravity, x and y pin forces, and the moment caused by viscous damping. The forces acting on the slider are the reaction forces and moment form the pinned pendulum, the normal force from the track, identical spring forces, and the dry friction force opposing motion. Once the newtonian force and moment equations combined with the kinematic equations are simplified for the whole system, the resulting two equations of motion:

$$(m_1 + m_2)x + m_1 l(\cos(\theta) + \mu\sin(\theta))\theta + m_1 l(\mu\cos(\theta) - \sin(\theta))\theta + \mu(m_1 + m_2)g + 2k\Delta x = 0 \quad (1)$$

$$m_1 l\cos(\theta)\ddot{x} + (I_c + m_1 l^2)\ddot{\theta} + m_1 g l\sin(\theta) = M_0 \quad (2)$$

are observed to be nonlinear. This prompts a restructure into the state form $M\dot{X} = F$ using $x,\ \dot{x},\ \theta,\ \dot{\theta}$ as the states, which is more suitable for simulation:

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & m_1 + m_2 & 0 & m_1 l(\cos(X_3) + \mu\sin(X_3)) \\ 0 & 0 & 1 & 0 \\ 0 & m_1 l\cos(X_3) & 0 & I_c + m_1 l^2 \end{bmatrix} \begin{bmatrix} \dot{X}_1 \\ \dot{X}_2 \\ \dot{X}_3 \\ \dot{X}_4 \end{bmatrix} = \begin{bmatrix} X_2 \\ -m_1 l[\mu\cos(X_3) - \sin(X_3)]X_4^2 - \mu(m_1 + m_2)g - 2kX_1 \\ X_4 \\ M_0 - m_1 g l\sin(X_3) \end{bmatrix} \quad (3)$$

To linearize the system model, equilibrium positions must be determined and Taylor's theorem applied.

The equilibrium positions are the following:

$\dot{X} = 0;\ \dot{x} = 0,\ \dot{\theta} = 0,\ \sin(\theta) = M_0/(m_1 g l),\ \Delta x = -\frac{\mu(m_1+m_2)g}{2k}$. Using the Taylor series expansion, these equations:

$$\left.\frac{\partial g(X)}{\partial X}\right|_{X=X_0} = \left.\left(M^{-1}\frac{\partial F}{\partial X}\right)\right|_{X=X_0} \quad (4)$$

$$\frac{\partial F}{\partial X} = \left[\frac{\partial F}{\partial X_1}\frac{\partial F}{\partial X_2}\frac{\partial F}{\partial X_3}\cdots\cdots\frac{\partial F}{\partial X_n}\right] \quad (5)$$

are arrived at, which when combined with the equilibrium conditions determined previously, yields the linearized state space equations:

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & m_1+m_2 & 0 & m_1 l \\ 0 & 0 & 1 & 0 \\ 0 & m_1 l & 0 & I_c+m_1 l^2 \end{bmatrix} \begin{bmatrix} \dot{X}_1 \\ \dot{X}_2 \\ \dot{X}_3 \\ \dot{X}_4 \end{bmatrix} = \begin{bmatrix} X_2 \\ -2kX_1 \\ X_4 \\ -cX_4 - m_1 g l X_3 \end{bmatrix} \tag{6}$$

IV. Procedure

Using the equations derived in the theory section, MatLab and Simulink were used to simulate both linear and nonlinear models of the pendulum-slider system. By reducing the system of second-order differential equations to a system of first-order differential equations, MatLab is able to use numerical methods to estimate the solution to the system given initial conditions. Therefore, functions were defined that would output the mass matrices and forcing functions for the linear and nonlinear systems. These functions were used in a differential equation solver with the given initial conditions to produce the given solutions to the system of differential equations. Similarly, Simulink was used to graphically produce the results for linear and nonlinear systems. The main difference being that simulink does not require the system be reduced to a system of first-order differential equations. The results of both were then compared to show the validity of using either solver for linear or nonlinear differential equations.

After completing the simulations for the linearized and nonlinear pendulum-slider systems, the response of the systems was experimentally determined for comparison with our simulations. The slider position potentiometer and then the pendulum position potentiometer were calibrated using a linear trendline of recorded potentiometer voltage values on the experimental set-up shown in Figure 1. Once the system was calibrated, the next step was to estimate both the pendulum viscous damping coefficient and the slider dry friction coefficient simultaneously. With initial conditions of 30° for the pendulum and 3 centimeters for the slider, a simulink data acquisition program was run to gather potentiometer voltages that were then converted into positions for the pendulum and slider. Using the gathered data, it was possible to calculate the damped and undamped natural frequencies for the pendulum, and therefore the pendulum's viscous damping coefficient. It was also possible to relate the oscillation amplitudes at the initial time and end time to calculate the dry friction coefficient.

V. Results and Discussion

Part 1.

With the given values for initial conditions and system parameters, *Figure 2* shows the results of the linear simulation that was run in MatLab, and *Figure 3* shows the results of the linear simulation in Simulink. Similarly, *Figures 4 and 5* show the Matlab and Simulink simulations, respectively, for the nonlinear system.

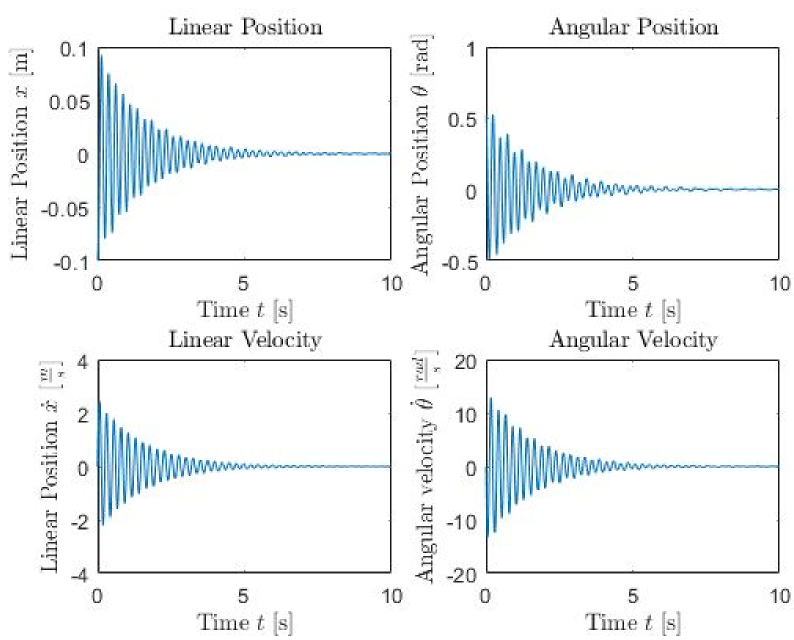

Figure 2. MatLab plots of the linear system solution

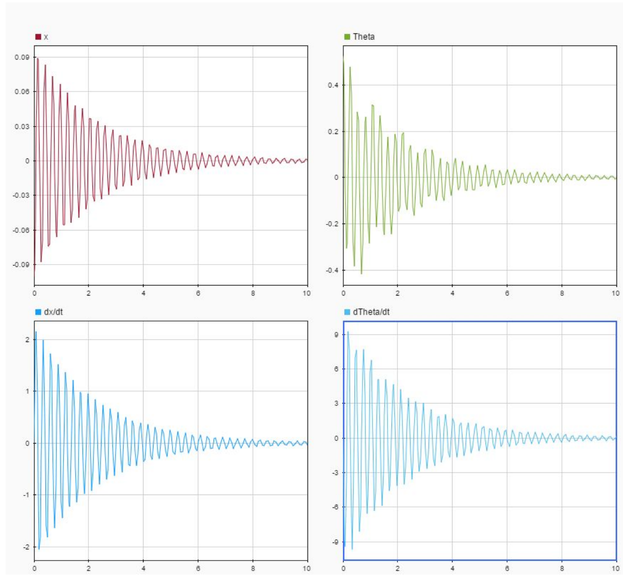

Figure 3. Simulink plots of the linear system solution

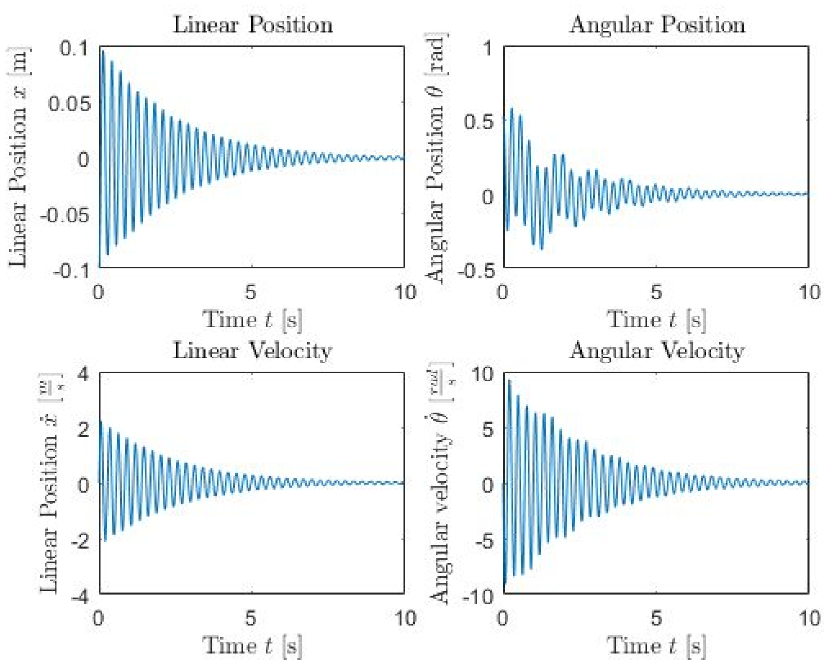

Figure 4. MatLab plots of the nonlinear system solution

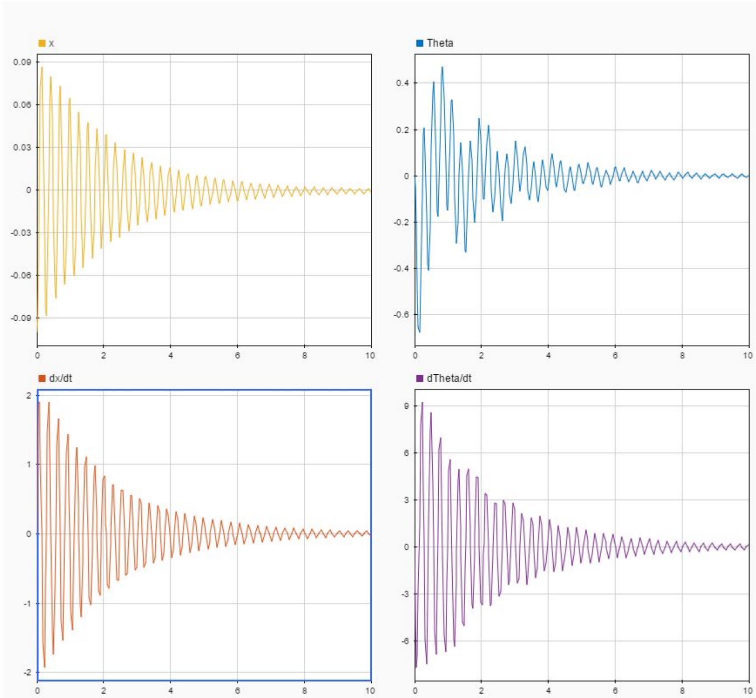

Figure 5. Simulink plots of the nonlinear system solution

As shown in the figures, the simulations run in MatLab and Simulink are identical, proving that either can be used to simulate or analyze data. Furthermore, the responses of the linear and nonlinear systems are similar, but not identical. For the initial large displacement of the system, the nonlinearity of the system is much more evident in the shown solution. However, once the system dampens to smaller displacements, it approaches the solution shown by the linear simulation. As can be seen in the figure, the linear system becomes a good approximation of the nonlinear system once the displacement of the pendulum falls below ~0.15 rad (this is consistent with the small angle assumption typically made in dynamic analysis). The reason why this is the case and why there is a difference in the linear and nonlinear systems is because there is a nonlinearity in the system due to the swinging pendulum. The restoring force on the pendulum is a nonlinear function of angle (sin). Therefore, only for small angles when $\sin(\theta) \approx \theta$ can we linearize the system.

Part 2.

Figures 6 and 7 show the calibration curves for the slider and pendulum voltages. Through the calibration procedure, the equations to transform the voltages to position and angle were as follows:

Slider Position [cm] = -1.0079 [cm/V] * Voltage [V] - 0.388 [cm] (7)

Pendulum Position [deg] = -32.138 [deg/V] * Voltage [V] - 86.101 [deg] (8)

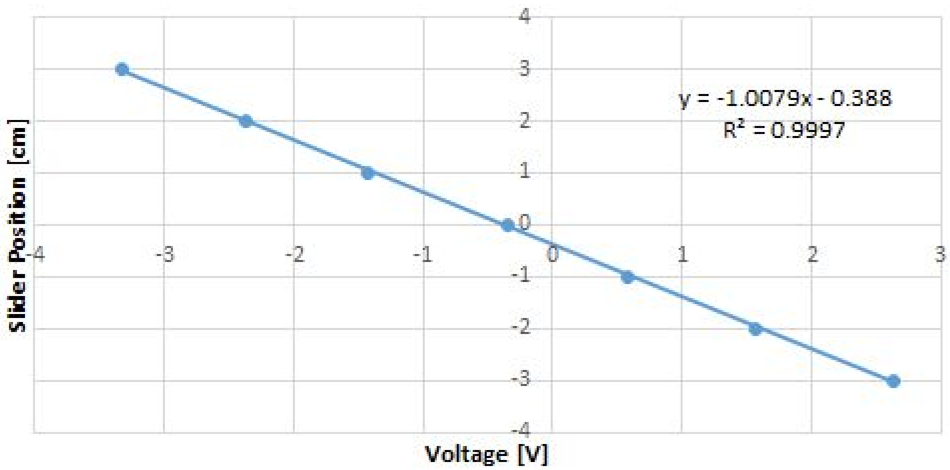

Figure 6. Calibration curve of the slider position

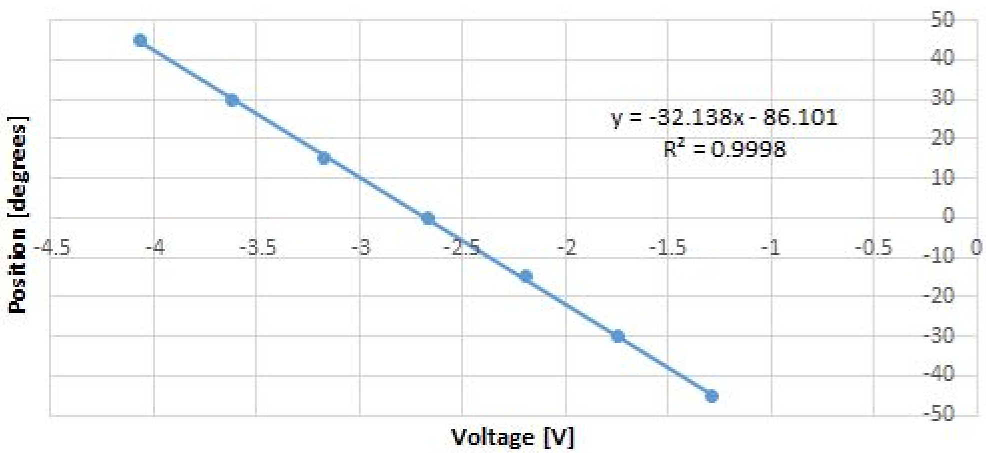

Figure 7. Calibration curve of the pendulum position

As is shown, the R-squared values are very close to 1, indicating that it is likely that the chosen regression of a line is a good fit for the data. Additionally, the transducers used in the experiment are linear, so this result is expected. However, these results only show the linearity of the transducers for the tested range. It would be possible to extrapolate these curves to predict the voltage at 60° or -4.5 centimeters, but it is unknowable without testing if the transducer is linear in that range or if there is a voltage limit that would be reached that would make such an extrapolation invalid.

Figures 8 and 9 show the curves used to estimate the viscous damping coefficient and dry-coulomb friction coefficients, respectively. By analyzing the decay of the isolated pendulum and slider motions, the viscous damping coefficient was calculated to be 0.2445 $\frac{Ns}{m}$, and the dry-coulomb friction coefficient was calculated to be 0.0732. It must be stated that the decay in the pendulum and slider motions are not due to *purely* viscous damping and dry friction respectively. There is dry friction in the axle of the pendulum and there is viscous damping as the slider moves through air. However, because these effects are much smaller than the originally stated cause of the damping, it is assumed that these minor effects are negatable.

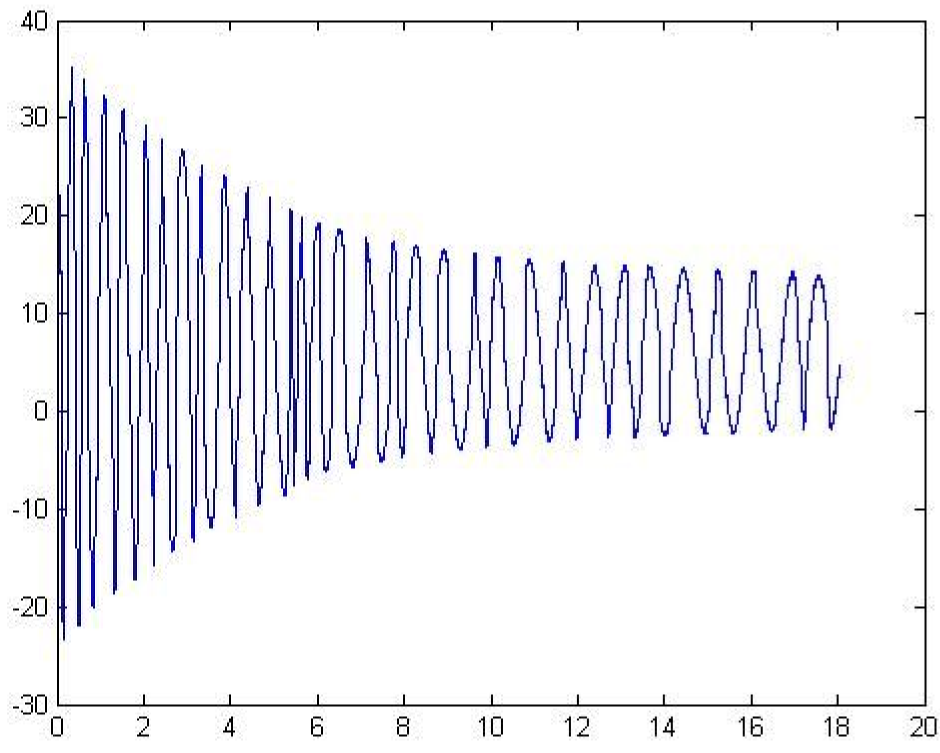

Figure 8. Isolated motion of the pendulum

$$\omega_d = \frac{2\pi}{\tau_p} = \frac{2\pi * 24}{14\,s} = 10.771\ hz \qquad (9)$$

$$\varsigma = {\frac{\left[\frac{1}{n}ln\left(\frac{x(t_0)}{x(t_n)}\right)\right]^2}{(2\pi)^2+\left[\frac{1}{n}ln\left(\frac{x(t_0)}{x(t_n)}\right)\right]^2}}^{1/2} \tag{10}$$

$$\varsigma = {\frac{\left[\frac{1}{27}ln(\frac{30}{14})\right]^2}{(2\pi)^2+\left[\frac{1}{27}ln(\frac{30}{14})\right]^2}}^{1/2} = 0.0044925 \tag{11}$$

$$\omega_{n,p} = \frac{\omega_d}{\sqrt{1-\varsigma^2}} \tag{12}$$

$$\omega_{n,p} = \frac{10.771\ hz}{\sqrt{1-0.01135^2}} = 10.772\ hz \tag{13}$$

$$C = 2\varsigma\omega_n = 2 * 0.0044925 * 10.772 = 0.0968\ \frac{Ns}{m} \tag{14}$$

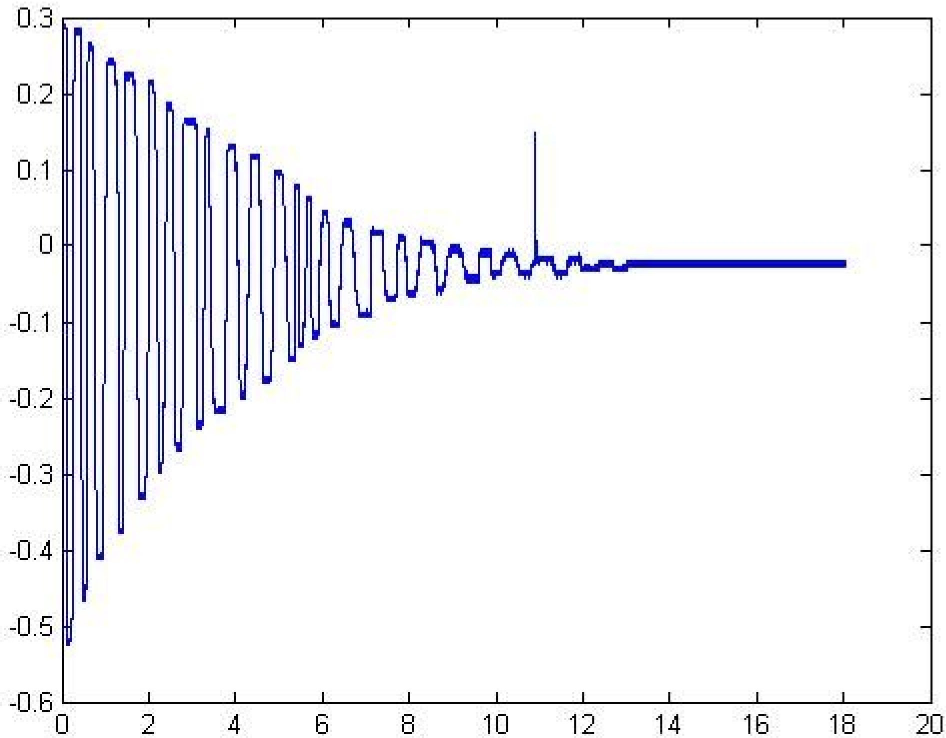

Figure 9. Isolated motion of the slider.

$$\omega_n = \frac{2\pi}{\tau_s} = \frac{2\pi*9}{4} = 14.137\ hz \tag{15}$$

$$\mu = \frac{[x(t_0)-x(t_n)]k_{eq}}{4ng(m_s+m_p)} \tag{16}$$

$$\mu = \frac{[0.24-0.07]294}{4*9g(1.15+0.783)} = 0.0732 \quad (17)$$

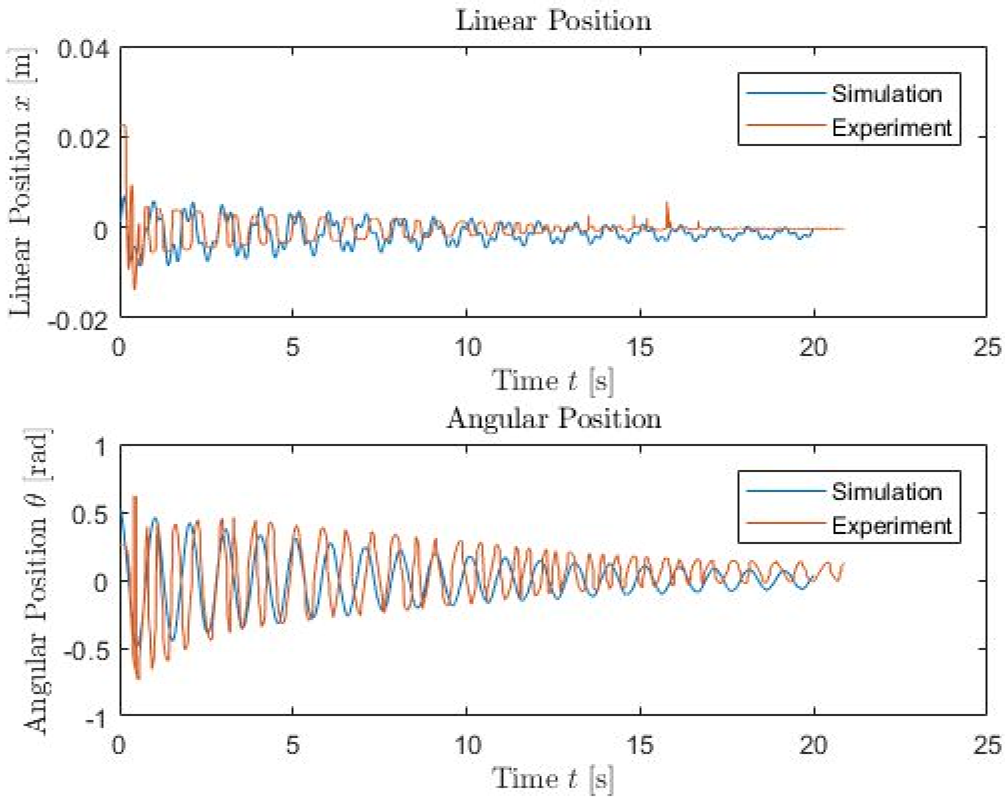

Figure 10. Comparison between experimental data and nonlinear simulation

Using the previously calculated values for the viscous damping and dry friction coefficients, the simulation produced very different results from the data that was collected. *Figure 10* shows how the value for the viscous damping had to be corrected from 0.09 $\frac{Ns}{m}$ to 0.01 $\frac{Ns}{m}$ to produce a similar graph. Additionally, the displacement of the slider was set to zero in order to simulate the effect noted in the experiment. Almost immediately after releasing the slider, it stopped vibrating freely and was only affected by the motion of the pendulum. The linear decay estimate for mu was quite accurate, without any adjustment required to produce the simulated result. Finally, the translational system is predicted to stop when the following conditions are met:

$$\dot{x} = 0 \quad (18)$$

$$x < \frac{\mu N}{k} \tag{19}$$

Moreover,

$$\frac{\mu N}{k} = \frac{0.0732 * (1.15+0.783)*9.81}{294} = 0.0047\ m \tag{20}$$

This implies that the slider should stop moving when the displacement is less than 0.47 cm and the velocity is zero. However, experimental data shows that the slider changes direction at much smaller displacements and continues to move. This is explained by the effect of the swinging pendulum. *Equations 18 - 20* assume the pendulum to not be swinging. Because that is not the case, a force from the pivot point causes the slider to continue to oscillate at displacements lower than that which was predicted to be possible.

VI. Conclusions

In the first section of this lab, the linear and nonlinear simulations through MatLab and Simulink showed that under the small angle assumption, the linear and nonlinear models are effectively the same. Additionally, the results of these simulations showed that MatLab and Simulink produce identical results for the same simulations so experiments can be run in either program. The creation and analyzing of these simulations explored the solving of differential equations using MatLab and Simulink and showed the differences in linear and nonlinear system models.

In the second portion of the lab, the calibration procedure showed that a linear regression was the best fit for the transducers used on the slider and pendulum due to the closeness of fit. This portion explored calibration techniques and issues with extrapolating curves beyond calibration regions. The comparison of the experimental data to the simulation based on calculated constants showed the discrepancies between the calculated values and the values required to produce a similar result in the simulation. The viscous damping coefficient was calculated to be 0.0968 $\frac{Ns}{m}$, but a value of 0.01 $\frac{Ns}{m}$ was used to produce the similar simulation. This indicates an issue in the methods used to calculate this value. It is likely that the coupling of the slider and pendulum cannot exactly be modeled by values calculated through isolating either system. The dry friction coefficient was calculated to be 0.0732 and it was not necessary to adjust this value to produce the similar simulation. This could indicate the system for calculating this value is very rigorous, however, the value has little effect on the simulation which is more dependent on the motion of the pendulum. Therefore, it is unknown if this value is truly accurate based on the experimental data. Finally, the experimental data disproved the results of step 25 in Lab 3 of the lab manual. This was due to the coupling of the pendulum and slider.

Overall, this lab explored the simulation of a multi-DOF system and how simulink can be used in conjunction with MatLab to collect and compare experimental data to simulation data. Within the realm of dynamical modelling, this will be invaluable in future experiments.

VII. References

Alladi, Vijay et. al. (2006) *MEEN 364 Lab Manual: Lab 2 - Pendulum-Slider System Simulation.* Texas A&M University. Web.

Alladi, Vijay et. al. (2006) *MEEN 364 Lab Manual: Lab 3 - 2 DOF Pendulum-Slider System Free Vibration Analysis and Measurements.* Texas A&M University. Web.

S.S. Rao, *Mechanical Vibrations*, 4^{th} edition, 2004.

VIII. Appendices

A. MatLab Code.

Lab2_Linear.m

```
% script for running the linear model and plotting the results
tspan = [0 10]; % simulation time
options = odeset('mass',@MMLinear);
y0 = [-0.1, 0, pi/6, 0]; % initial conditions
[t,y] = ode45(@FFLinear,tspan,y0,options); % differential equation solver

figure
subplot(2,2,1);
plot(t,y(:,1))
title('Linear Position','interpreter','latex');
xlabel('Time $t$ [s]','interpreter', 'latex');
ylabel('Linear Position $x$ [m]','interpreter','latex');

subplot(2,2,3);
plot(t,y(:,2))
title('Linear Velocity','interpreter','latex');
xlabel('Time $t$ [s]','interpreter', 'latex');
ylabel('Linear Position $\dot{x}$ [$\frac{m}{s}$]','interpreter','latex');

subplot(2,2,2);
plot(t,y(:,3))
title('Angular Position','interpreter','latex');
xlabel('Time $t$ [s]','interpreter', 'latex');
ylabel('Angular Position $\theta$ [rad]','interpreter','latex');

subplot(2,2,4);
plot(t,y(:,4))
title('Angular Velocity','interpreter','latex');
xlabel('Time $t$ [s]','interpreter', 'latex');
ylabel('Angular velocity $\dot{\theta}$ [$\frac{rad}{s}$]','interpreter','latex');
```

FFLinear.m

```
function yp=FFLinear(t,y)
   % forcing function for the linear system
   m1 = 0.78;
   k = 294;
   c = 0.05;
   L = 1;
   g = 9.81;

   yp=zeros(4,1);

   yp(1)=y(2);
   yp(2)= -2*k*y(1);
   yp(3)=y(4);
   yp(4)=-c*y(4) - m1*g*L*y(3);
end
```

MMLinear.m

```
function n = MMLinear(t,y)
   % mass matrix function for the linear system
   m1 = 0.78;
   m2 = 1.15;
   L = .23;
   Ic = 0.0014;

   n = [   1   0          0   0       ;...
        0   m1 + m2    0   m1*L    ;...
        0   0          1   0       ;...
        0   m1*L    0   Ic + m1*L^2];
end
```

Lab2_NonLinear.m

```
% script for running the nonlinear model and plotting the results
tspan = [0 10]; % experimental runtime
options = odeset('mass',@MMNonLinear);
y0 = [-0.1, 0, pi/6, 0]; % initial conditions
[t,y] = ode45(@FFNonLinear,tspan,y0,options); % differential equation solver

figure
plot(t, y(:,1));

subplot(2,2,1);
plot(t,y(:,1))
title('Linear Position','interpreter','latex');
```

```
xlabel('Time $t$ [s]','interpreter', 'latex');
ylabel('Linear Position $x$ [m]','interpreter','latex');

subplot(2,2,3);
plot(t,y(:,2))
title('Linear Velocity','interpreter','latex');
xlabel('Time $t$ [s]','interpreter', 'latex');
ylabel('Linear Position $\dot{x}$ [$\frac{m}{s}$]','interpreter','latex');

subplot(2,2,2);
plot(t,y(:,3))
title('Angular Position','interpreter','latex');
xlabel('Time $t$ [s]','interpreter', 'latex');
ylabel('Angular Position $\theta$ [rad]','interpreter','latex');

subplot(2,2,4);
plot(t,y(:,4))
title('Angular Velocity','interpreter','latex');
xlabel('Time $t$ [s]','interpreter', 'latex');
ylabel('Angular velocity $\dot{\theta}$ [$\frac{rad}{s}$]','interpreter','latex');
```

FFNonLinear.m

```
function yp=FFNonLinear(t,y)
  % forcing function for the nonlinear system
  m1 = 0.78;
  m2 = 1.15;
  k = 294;
  c = 0.05;
  L = 0.23;
  g = 9.81;
  Mo = -c*y(4);
  mu = 0.03;

  yp=zeros(4,1);

  yp(1)=  y(2);
  yp(2)=  -m1*L*(mu*cos(y(3))-sin(y(3)))*y(4)^2 - mu*(m1+m2)*g-2*k*y(1);
  yp(3)=  y(4);
  yp(4)=  Mo - m1*g*L*sin(y(3));

end
```

MMNonLinear.m

```
function n = MMNonLinear(t,y)
    % mass matrix for the nonlinear system
    m1 = 0.78;
    m2 = 1.15;
    L = .23;
    Ic = 0.0014;
    mu = 0.03;

    n = [   1   0              0   0                              ;...
            0   m1 + m2        0   m1*L*(cos(y(3)) + mu*sin(y(3))) ;...
            0   0              1   0                              ;...
            0   m1*L*cos(y(3)) 0   Ic + m1*L^2                    ];

end
```

B. Simulink Code.

Lab2_LinearSim.slx

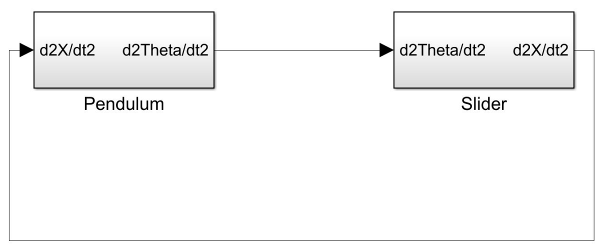

Figure 11. Simulink linear simulation subsystem view

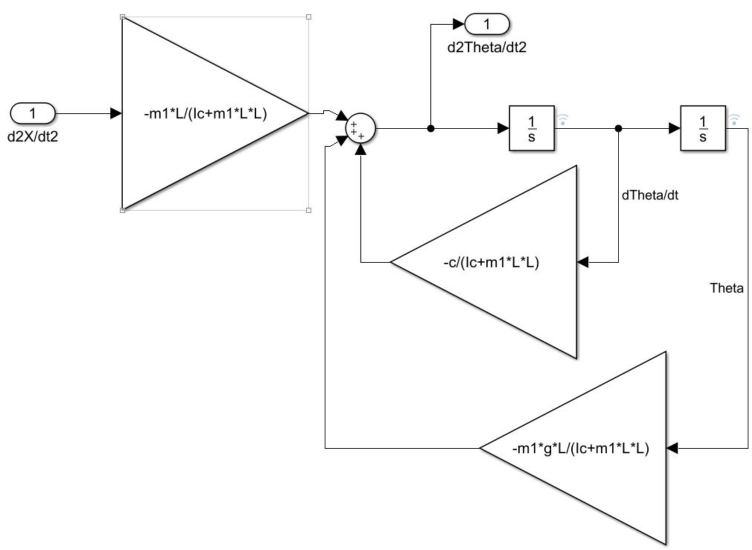

Figure 12. Pendulum subsystem

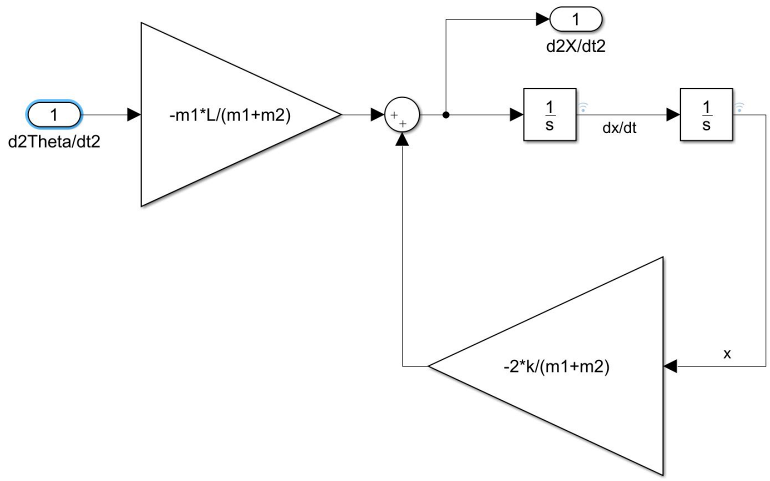

Figure 13. Slider subsystem

Lab2_NonLinearSim.slx

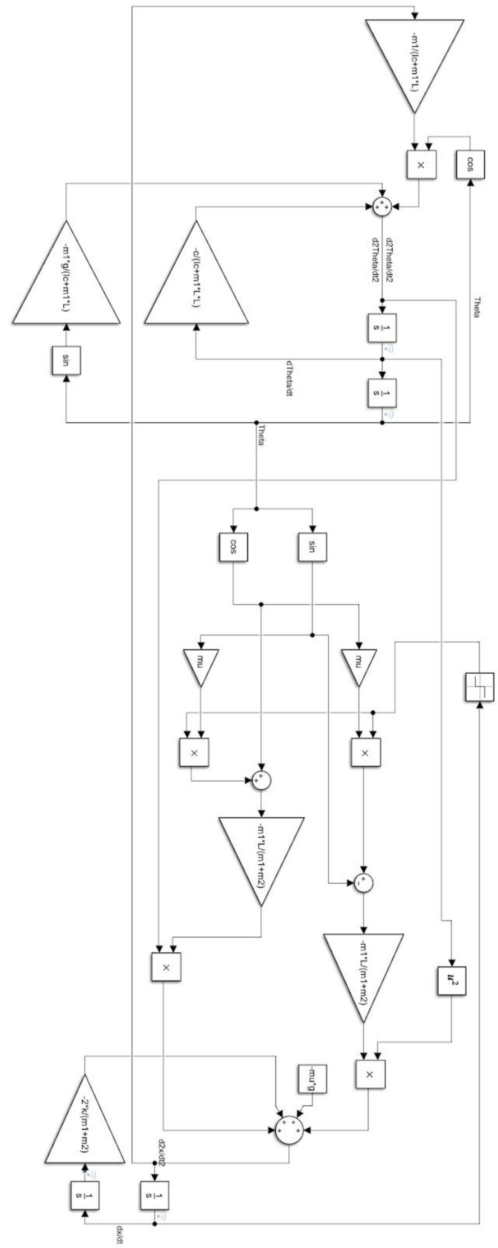

Figure 14. Nonlinear Simulink simulation

C. Raw Data.

Table 1. Selected raw data from multi-DOF experiment

Time [s]	Slider Potentiometer Voltage [V]
0	2.249863
0.001	2.244942
0.002	2.249863
0.003	2.249863
0.004	2.249863
0.005	2.249863
0.006	2.249863
0.007	2.249863
0.008	2.249863
0.009	2.249863
0.01	2.249863
0.011	2.244942
0.012	2.249863
0.013	2.249863
0.014	2.249863
0.015	2.249863
0.016	2.249863
0.017	2.249863
0.018	2.249863
0.019	2.244942