Elec Eng 4CL4
Lab 1 – Phase 0: Modeling and Simulation of State-Space Control Systems in MATLAB/SIMULINK Environment

Instructor: Dr. T. Kirubarajan (Kiruba)
George Gill – L03 Group 1 –Gillg62- 400327563
Richa Rani – L03 Group 1 –Ranir- 400311161
TA:

October 4th, 2024

As a future member of the engineering profession, the student is responsible for performing the required work in an honest manner, without plagiarism and cheating. Submitting this work with my name and student number is a statement and understanding that this work is my own and adheres to the Academic Integrity Policy of McMaster University and the Code of Conduct of the Professional Engineers of Ontario. Submitted by [George Gill, Gillg62, 400327563]

As a future member of the engineering profession, the student is responsible for performing the required work in an honest manner, without plagiarism and cheating. Submitting this work with my name and student number is a statement and understanding that this work is my own and adheres to the Academic Integrity Policy of McMaster University and the Code of Conduct of the Professional Engineers of Ontario. Submitted by [Richa Rani, Ranir, 400311161]

Table of Contents

Table of Figures

Abstract

This experiment acts as an introduction to modeling dynamical systems in MATLAB/Simulink. The objective of this laboratory was to get used to the environment and gain knowledge on the different tools, as a first step to ultimately model and simulate an inverted pendulum mechanism. Given two dynamical systems, a single pendulum and a double mass-spring-damper system, three different methods of modeling and simulation were used. The first approach consisted of using multiple integrator blocks to model the single pendulum. The integrator block offered by MATLAB serves as an integration block hence the name integrator, whose purpose is to take in an input and output the respective integration. In our case, we input a state derivative and get an output as a state. By adjoining these blocks together, we can model a simple LTI system. The second method uses a state-space block, this block represents the state-space equations. We used this method to model the mass-spring-damper system by inputting the respective matrices and initial values. Lastly, Simulink offers s-functions. This block takes in an m-file which in the background computes the output of the system given an input and initial conditions. An m-file was provided, which was manipulated to model the double mass-spring-damper system. This m-file consists of multiple functions which work together to reach an output. Each of these methods allows for modeling an LTI system, however, we noticed that the integrator method seems to quickly become hard to follow with the numerous blocks. As we move forward with simulating complex systems, the s-function method seems optimal and efficient. Overall, this experiment provided a better understanding of the different modeling methods provided by MATLAB/Simulink through two different applications.

State-Space Model of a Single Pendulum

Task 1

Let the vector of states x = [θ θ]T. With this choice of states, write the system dynamics in state-space form.

Results and Discussions

The vector states for the single pendulum system were found mathematically as shown in Appendix A Figure 15, where the states were derived by using the provided equations and system diagram from the lab manual. Originally, we approximated the value of $\dot{x}2$ by removing the sine function as seen in Figure 16. This changes the system to be a linear system and thus allows it to be written into matrix form. However, from our conversation with the TA, this was adjusted back to include the sine function as a state space model can be non-linear.

Task 2

Use two integrators to implement the system dynamics in Simulink.

Results and Discussions

The single pendulum system was implemented on Simulink using two integrator blocks, two gain blocks, a sine block, a subtractor, and a scope block. As discussed before, each integrator block outputs the integral of its input. In the first instance, the angular acceleration (from the subtractor) is inputted into to integrator block which outputs a state variable $\dot{x}2$ (angular velocity). In the second instance, this angular velocity is fed into the integrator block returning the angular position. The angular velocity and angular position are fed back into the subtractor after being manipulated by the gain blocks. The scope block allows us to visually depict the angular velocity and angular position of the system. The approximated approach was also implemented and differs slightly as it removes the sine function after the second integrator.

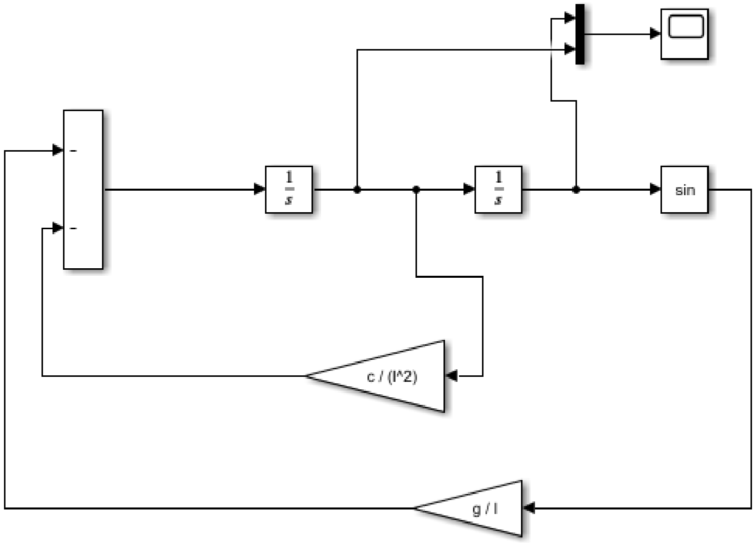


Figure 1 Single Pendulum Simulink Implementation

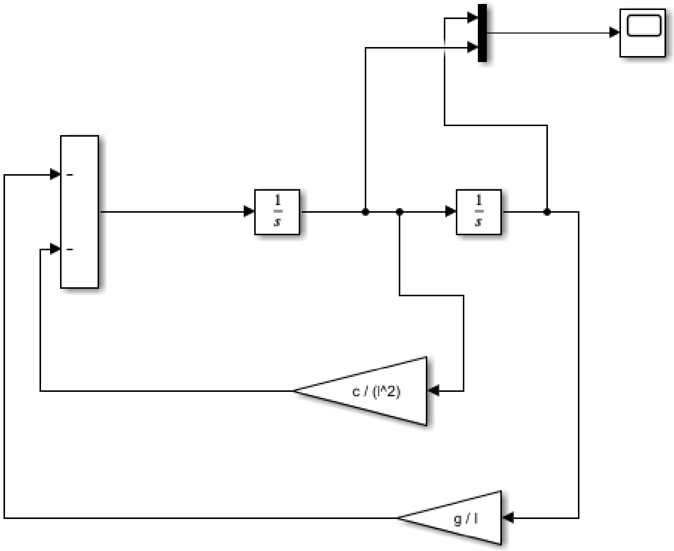


Figure 2 Single Pendulum Simulink Implementation With Approximation

Task 3

Assuming that m = 1, c = 0.8, l = 0.75 and g = 9.81, simulate the state response with an initial state of x0 = [π/4 0]T.

Results and Discussions

The simulation provided the plots shown below in Figures 3 and 4. Automatically we can see that the approximation has very little effect on the output of the plots, as both the Position and Velocity plots in both instances are almost identical. Furthermore, in both plots, we see the same trend. For the position, the pendulum starts at π/4 (initial condition) and oscillates until it settles at a final value of zero. Note the peak positional value is equal to the initial starting value. The velocity (same as the position) also oscillates until it reaches a resting value of zero. Its peak value occurs on the "first swing" and gradually decreases as the pendulum comes to a rest. This is due to the dampener located at the base of the pendulum which slows down the pendulum over time, ultimately reaching a settling value of 0 at a settling time of 7 seconds.

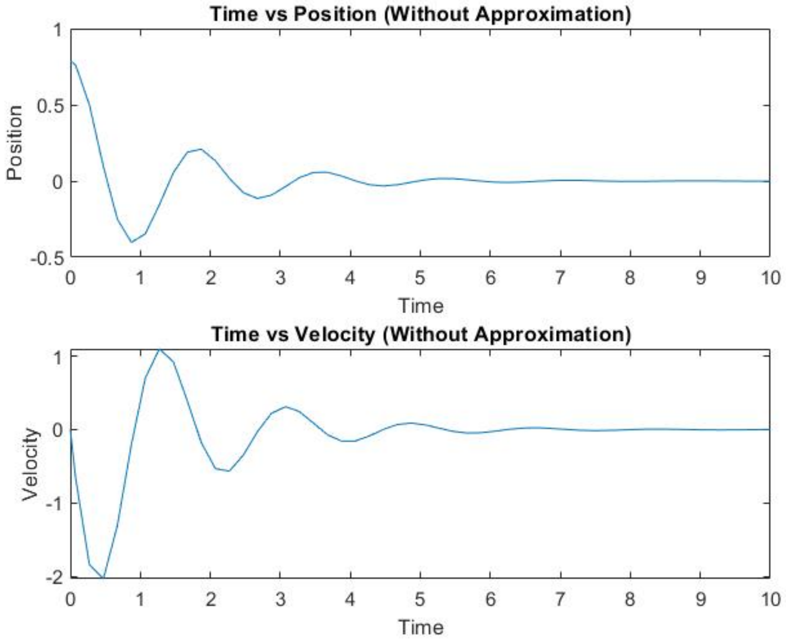


Figure 3 Time vs Position and Time vs Velocity

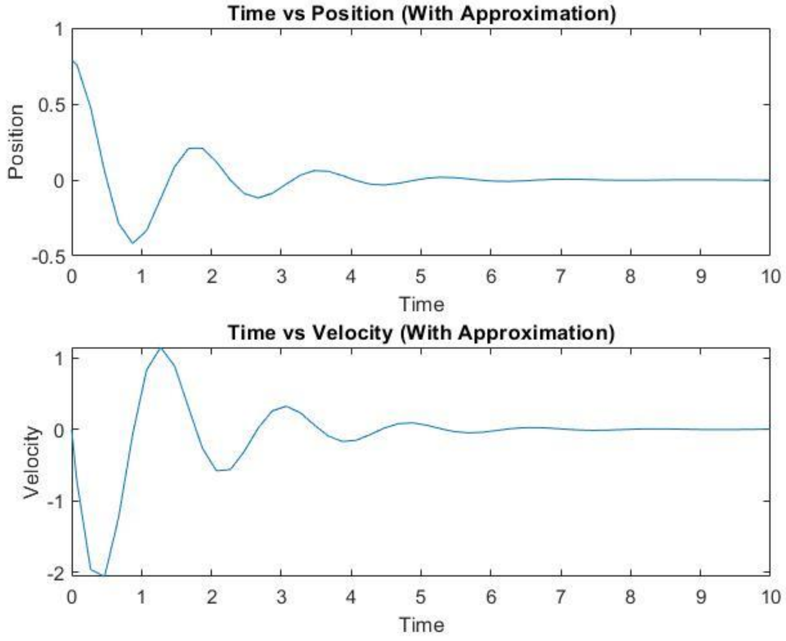


Figure 4 Time vs Position and Time vs Velocity with Approximation

Task 4

Repeat the simulation, this time by setting c = 0. How do you interpret the results?

Results and Discussions

The C constant in this application acts as the dampening force. Theoretically, we can predict that if this value is set to zero, then the system is ideal in the context that no forces are acting against the movement of this system. Thus, we can assume that the pendulum will oscillate forever at the same magnitude. This was confirmed by the simulations that are plotted below in Figures 5 and 6. Again, in both cases, we see identical results, as the pendulum seems to oscillate forever without changing its magnitude in either position or velocity. In conclusion, we can interpret this result as a case where the pendulum is an ideal environment.

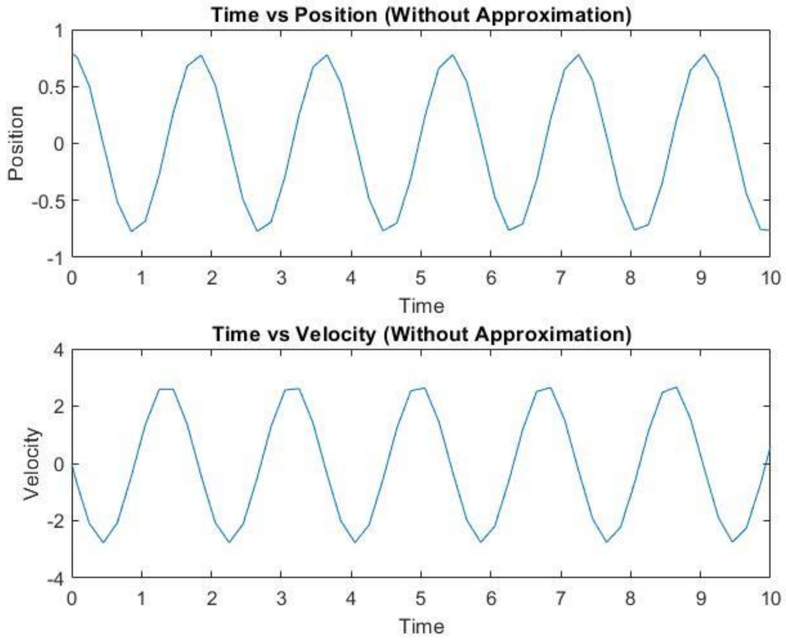


Figure 5 Time vs Position and Time vs Velocity with C = 0.

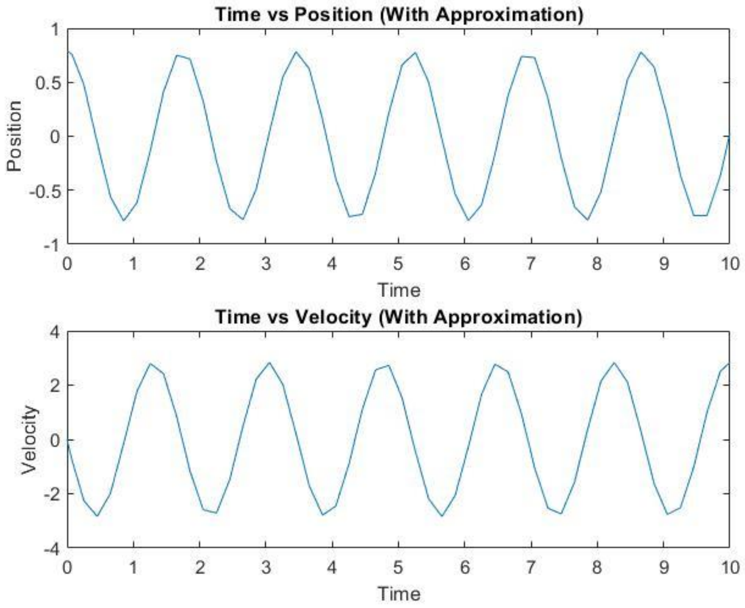


Figure 6 Time vs Position and Time vs Velocity with Approximation and C = 0.

Bonus Task

How does the approximation affect the system?

Results and Discussions

In both Tasks 3 and 4, we added the case of the approximation and no approximation. We got almost identical results in both cases, so we asked, "Does the approximation matter?”. This question was answered by running another simulation.

By looking at the dampener, we see it only acts on the system when the pendulum is moving (due to its placement), if the pendulum is stationary (located at position 0) then the dampener has no effect. Similarly, if the pendulum was inverted and positioned at position pi theoretically the pendulum should stay vertical forever.

After running the simulation, we get the following Figures 7 and 8 below. We now see that both instances differ significantly as compared to the last two tasks. As we theoretically discussed, the instance without approximation shows the pendulum being at a vertical position and staying in that position for the whole 10 seconds. Whereas in the approximation case, we see the same trend as we saw in Task 2 where the position and velocity both oscillate until the system settles at zero at around five seconds.

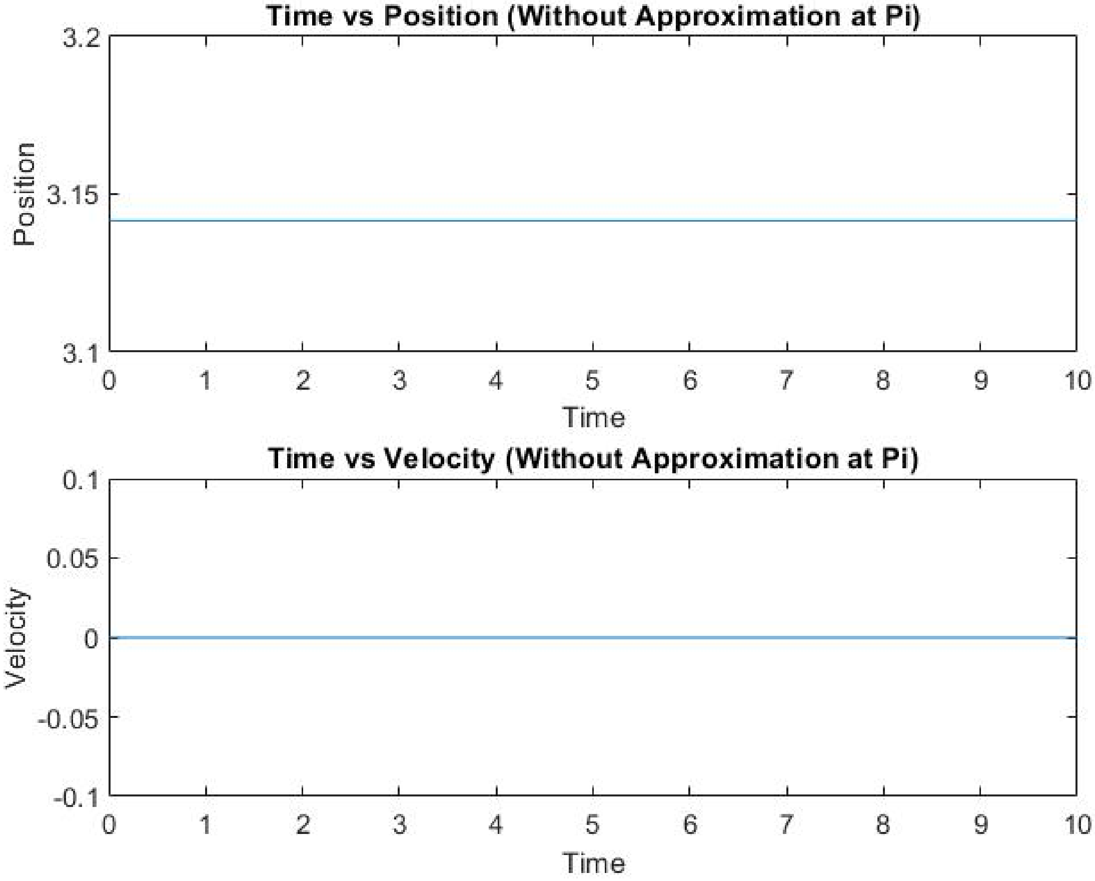


Figure 7 Time vs Position and Time vs Velocity with IC at Pi

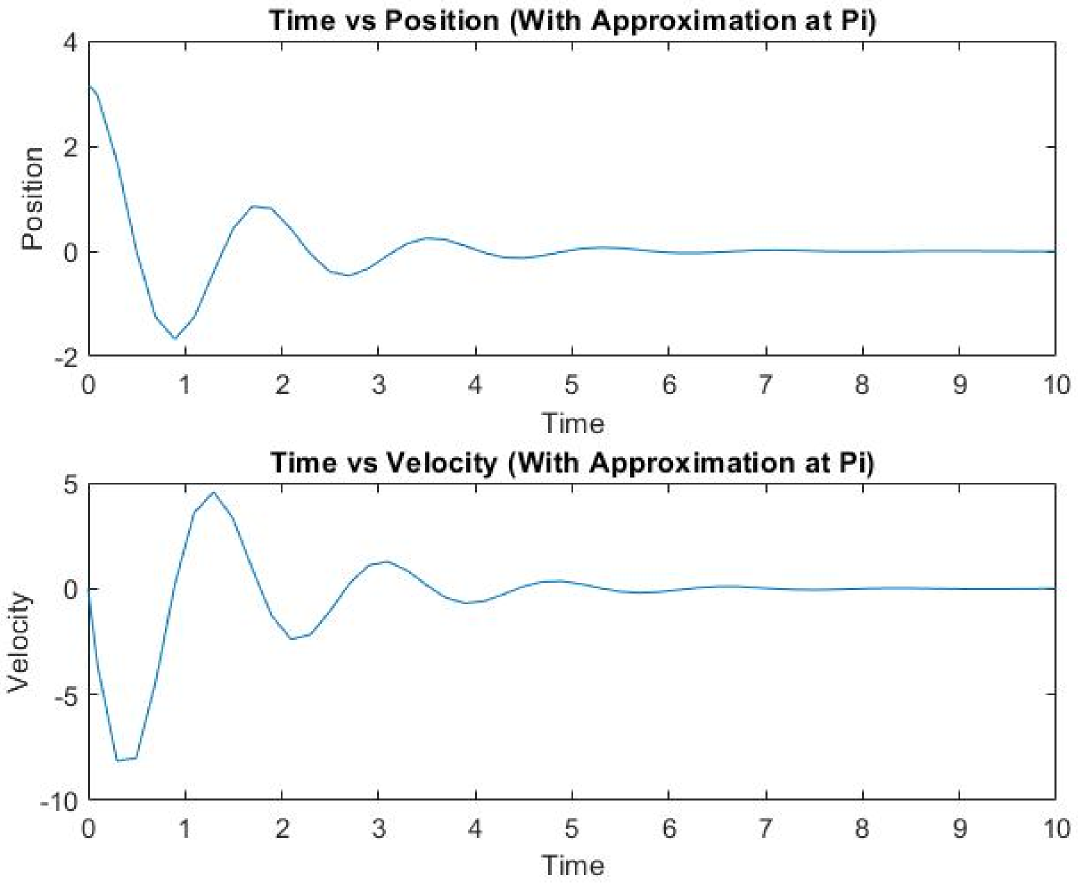


Figure 8 Time vs Position and Time vs Velocity with Approximation and IC at Pi.

State-Space Model of a Double Mass-Spring-Damper System

Task 5

Implement the state-space model of the double mass-spring-damper using the state-space block of Simulink. To accomplish this task, you need to write an m-script to compute the matrices A, B, C, and D. You need to define the outputs of the block as all of the states, i.e., $y = x$ so you can access the states outside the block.

The matrices of A, B, C, and D were determined mathematically by hand in *Figure 1.A* as well as in the m-script . These matrices were implemented into the state-space model simulated in Simulink.. As shown in Figure 5.1, a step function that goes to 50 at 2 was inputted into the state space block in Simulink. The step function is to simulate the 50N force which is applied to the system at 2s as outlined in the question in Task 7.

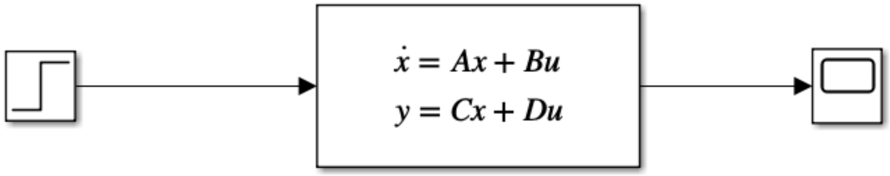


Figure 9 Simulink State Space Model for Mass Spring Damper System

Results and Discussions

Figure 5.2 below shows the resulting graph when this simulation is observed through the scope. This graph shows the Time against the position of the mass m1 and m2. The initial conditions for this system are $\boldsymbol{x}$**0 = [1 0.2 −0.2 0]** . This means the Initial position of mass 1 (yellow line) is 1, the initial velocity for mass 1 (blue) is 0.2, the initial position of mass 2 (red) is -0.2, the initial velocity of mass 2 (green) is 0.

Before t=2s

This graph shows that prior to the application of the force, mass 1 is moving to the right with the velocity of 0.2m/s. It moves to the right but the force of k1 to the left causes the velocity to deaccelerate until at 0.5 s where mass 1 momentarily comes to a halt and the force of the spring k1 causes It to accelerate in the opposite direction and mass 1 accelerates to the left. The amplitudes of mass 1 and velocity 1 are more than the amplitude of mass 2 and velocity 2 since mass 2 started off as stationary and is just responding to the forces caused by mass 1. Mass 2 moves to the right as a response of mass 1 moving to the left and the velocity increases. Eventually, after both masses oscillate due to the initial velocity and effects of the springs, they reach near an equilibrium state at around 1.1s.

At 2s:

At 2s, a 50 N force is applied to mass 2 which can be shows as the position and the velocity of m2 (green and blue) get a higher amplitude. The 2 masses oscillate until they reach an equilibrium.

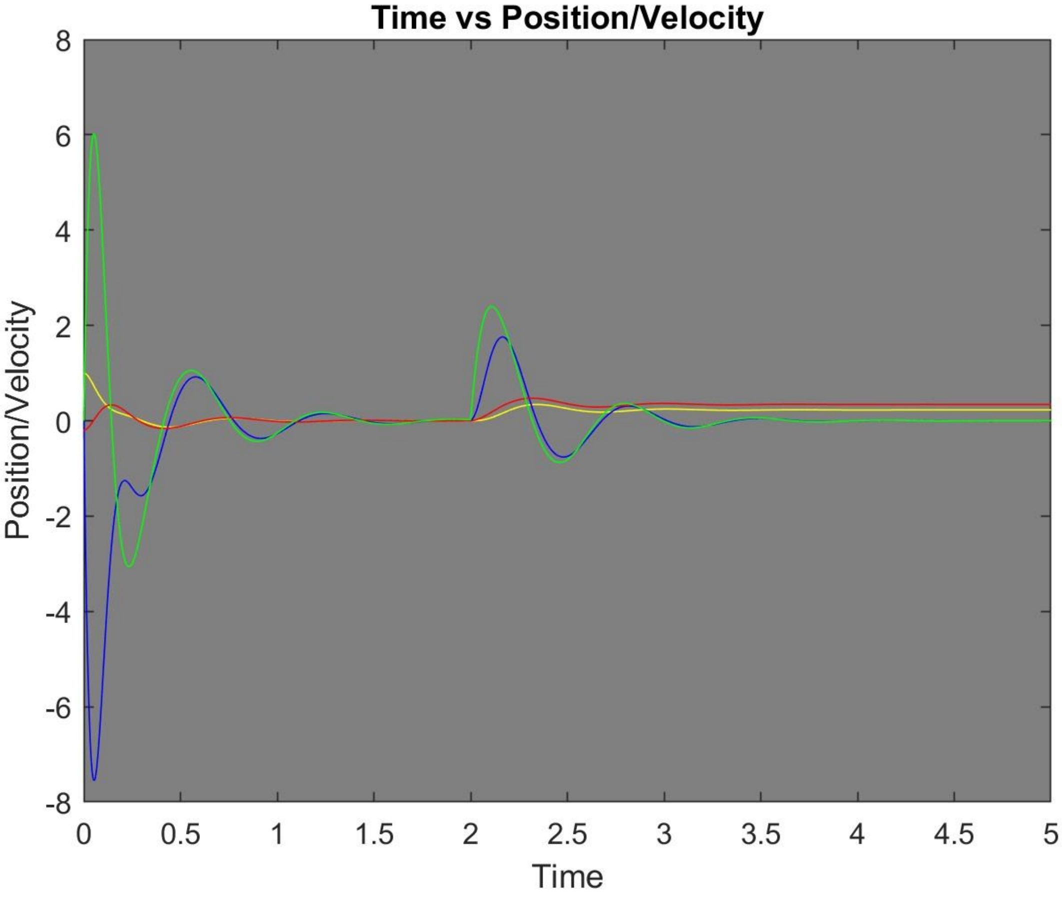


Figure 10 Mass Spring Damper System Simulation using State Space Model

Task 6 and 7

Implement the state-space model of the double mass-spring-damper by modifying sfuntmpl.m. This file, which is part of MATLAB/Simulink installed package, is also posted on "avenue to learn" for your reference. To accomplish this task, you need to modify the content of the file to set up your system configuration parameters and implement the state dynamics and output equations

Assume that $m1$ = 1, $b1$ = 10, $k1$ = 100, $m2$ = 1, $b2$ = 10, $k2$ = 200, $b3$ = 1, $k3$ = 80. Simulate the total response of the system to a step input force at t = 2, i.e., f = 50N, t ≥ 2 and the initial state of $x0$ = [1 0.2 −0.2 0]Tusing each of the above implementations and compare the results. Use a final time of tf = 5. Investigate the effect of the choice of maximum step size by using two values of 0.001 and 0.01, respectively. Set all the damping values to zero and repeat your simulations. What is your interpretation of the results?

The parameters that were outlined in the question above were implemented in the sfuntmpl.m file These values are then inputted into the A,B and C matrices that were determined in order to calculate the curve for time against position and velocity of the 2 masses. A step function that simulates the 50 N at 2 s is inputted into an s-function block as shown in *Figure 2* and measured with a scope

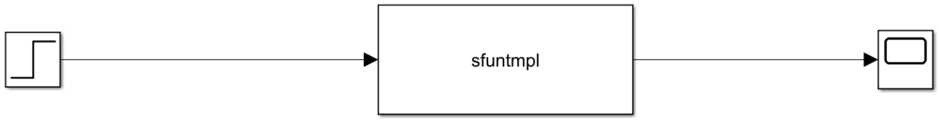


Figure 11 S-Function Block for Mass Spring Damper System

Results and Discussions

Results using Maximum Step Size = 0.001

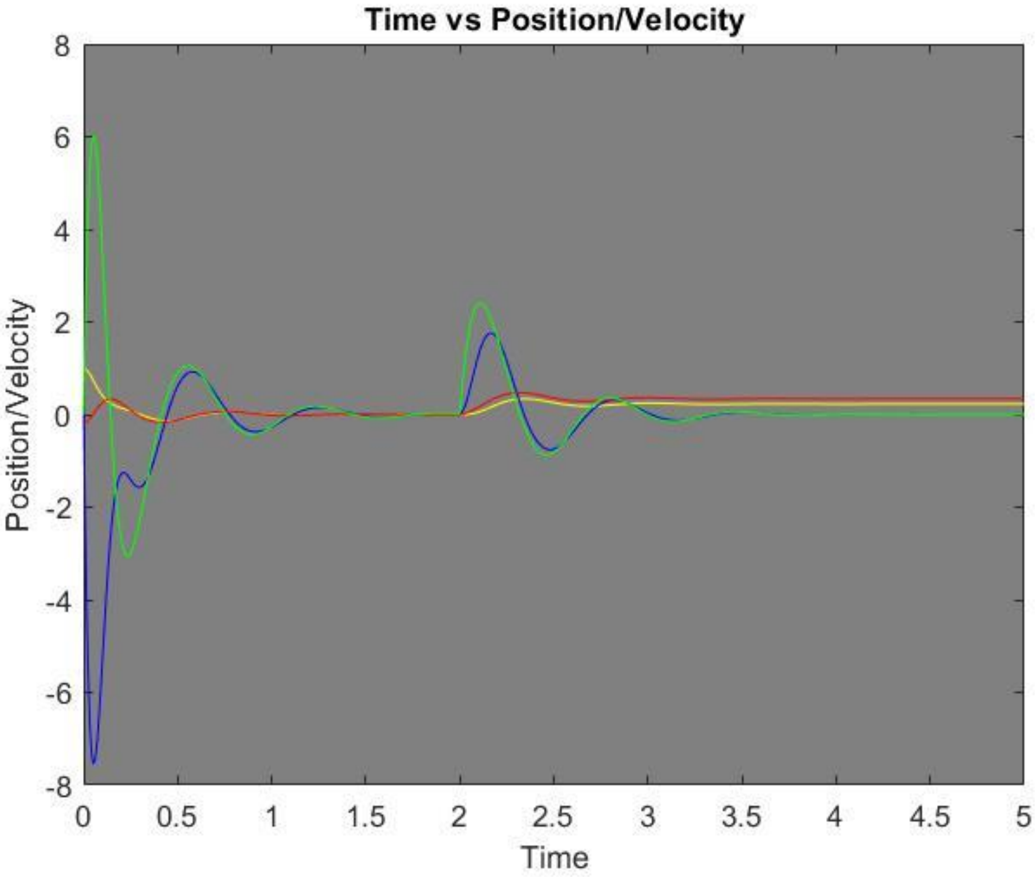


Figure 12 Mass Spring Damper System with the Step Size set to 0.001

The effect of the step size is that the simulation takes smaller steps to evaluate the stems state over time in the graph. This is evident in higher resolution in the graph, however, due to the step size being smaller, the computation time is also higher.

Results using Maximum Step Size = 0.01

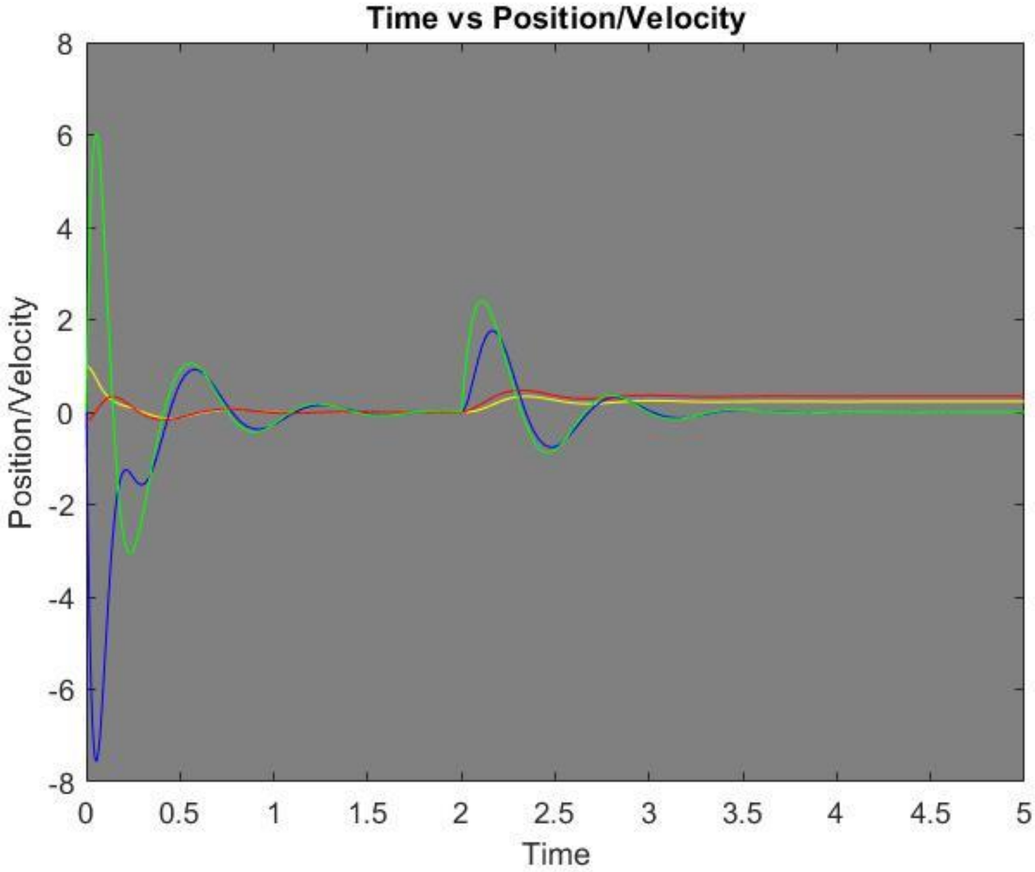


Figure 13 Mass Spring Damper System with the Step Size set to 0.01

In this graph, the step size is larger. When compared to the graph with a 0.001 step size, it is evident that the previous graph has a smoother curve and is a neater overall graph. However, the computation time for

Overall, a smaller step size should be used when a more accurate representation of the system with a cleaner and smoother graph is required. However, a larger step size should be used when a graph that takes less computational power is required.

Results when Damping Value is set to 0

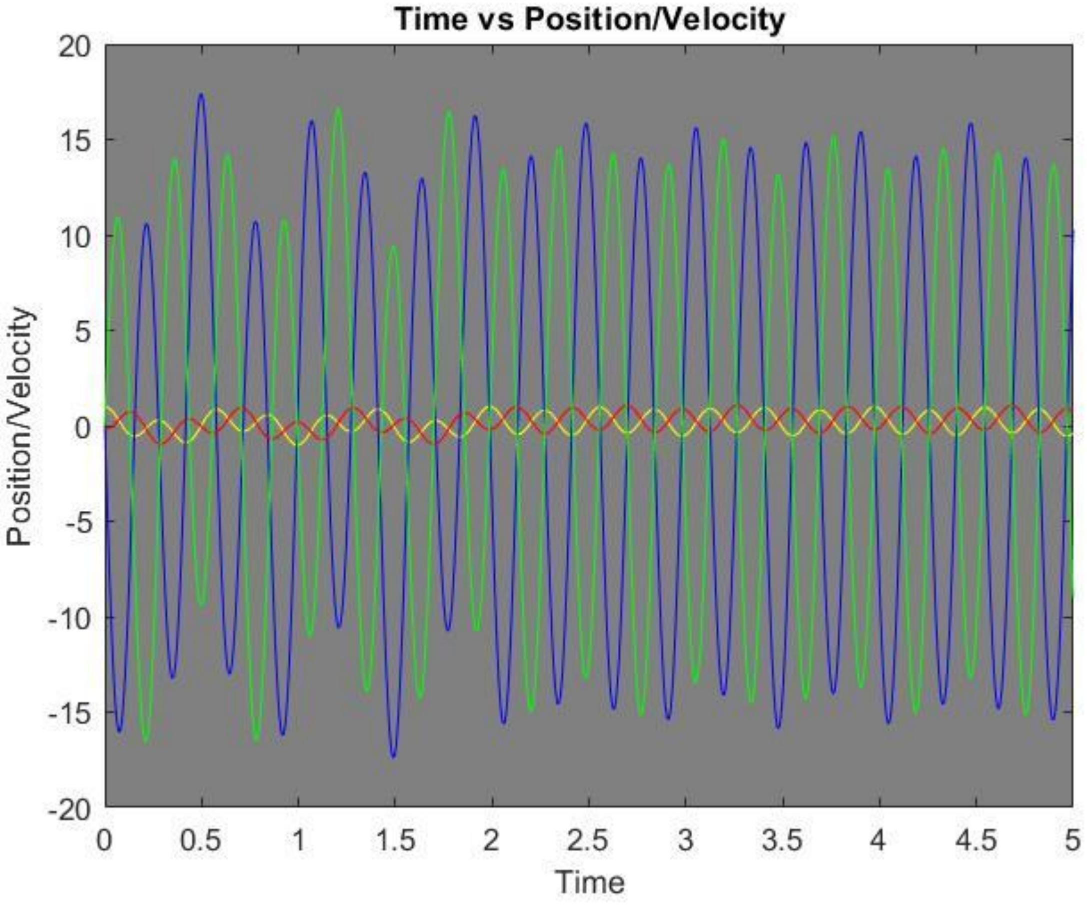


Figure 14 Mass Spring Damper System when Damping Co-efficient is 0

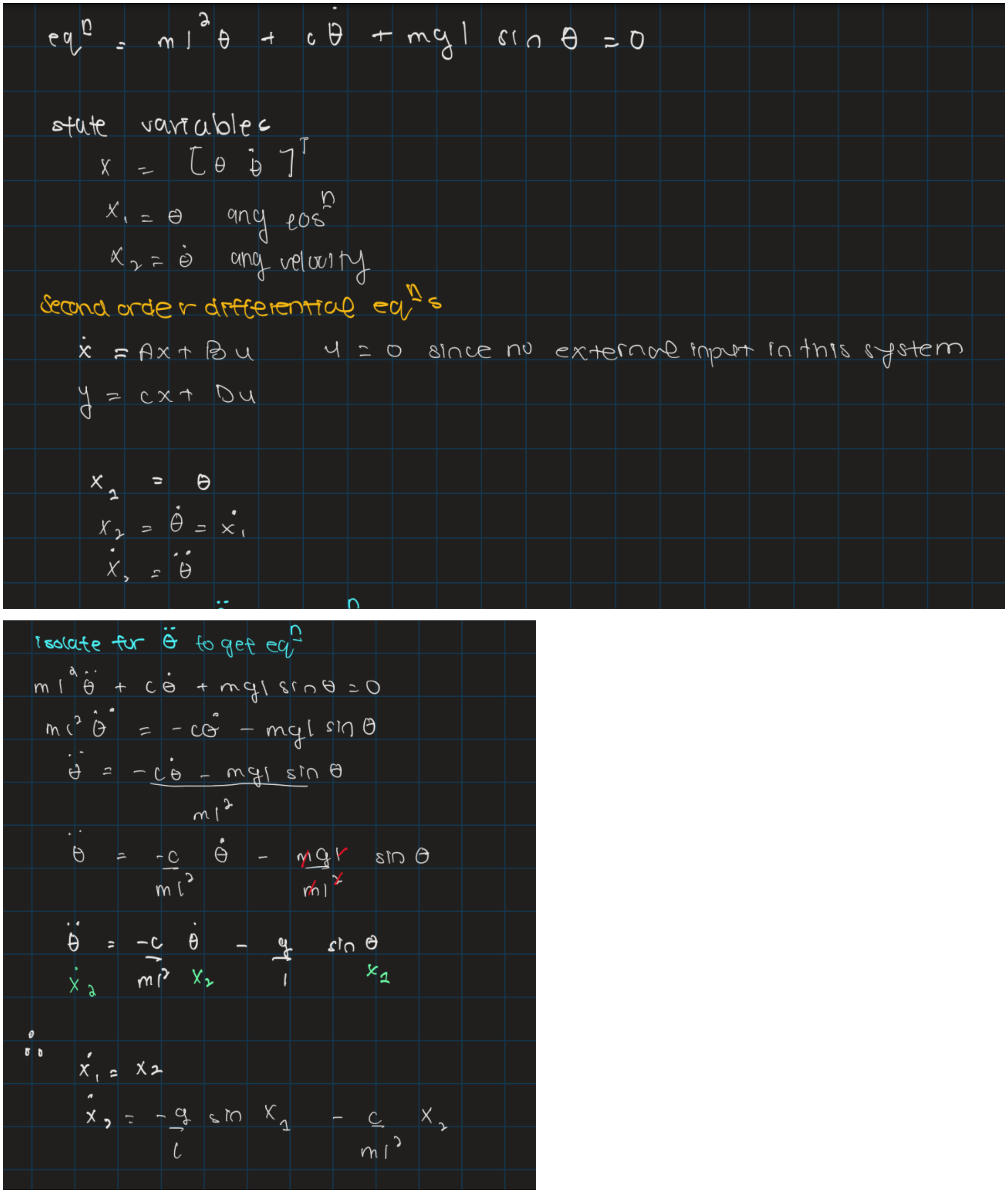
When the damping coefficient is set to 0, this system becomes underdamped. Since there is no dampening effects, there is no energy dissipation. This causes the system to remain oscillating forever due to there being no energy loss.

Conclusions

This lab provided a comprehensive understanding of different types of Simulink modelling tools to simulate dynamic systems. It used a single pendulum system and a spring mass damper system to illustrate which models are useful for simulating different types of systems, the effects of different variables in a dynamic system, and how to solve for and implement state space equations. Three types of modelling blocks were used to simulate these systems: the integrator blocks, state space blocks, and s function blocks. The integrator method is straightforward and good for a simple system like a singular pendulum, however, when a system becomes more complicated, it becomes convoluted and tedious to implement. The state space model is easier to implement and provides a neater representation of the system compared to the integrator blocks which made it better for a more complicated system like a 2 mass spring damper system. Finally, the s-block, while being slightly more complicated and the same level of neatness, provided more efficiency and flexibility and is better suited for a system as it grows in complexity. Using these methods to simulate systems provided a better platform to change see how the system reacts to different things, how each variable affects the system, and comparing different system states. During the single pendulum lab, the matrices were calculated in 2 methods, using sin(θ) as a term in our state space equation, and replacing sin(θ) with an approximation of just θ by assuming that θ is small. When both graphs were compared side by side, we saw that they provided similar results, however, is not a completely accurate simulation for when the pendulum is inverted at the top position. The single-mass pendulum and the two-mass spring damper system also provided an understanding of the dampening effect. It shows in both cases that when the dampening effect is not considered, the system oscillates forever due to no energy being lost in the system, whereas, with the dampening effect the system goes to equilibrium after a set amount of time. In addition, the effect of step size in creating graphs was considered. A larger step size generates a graph with a better computation time but compromises the accuracy of the graph. In conclusion, this lab provided tools and information to consider when modelling different types of systems in the future.

Appendix

Appendix A: Handwritten Solutions



$eq^n = ml^2\theta + c\dot{\theta} + mgl\sin\theta = 0$

state variables

$x = [\theta\ \dot{\theta}]^T$

$x_1 = \theta$ ang pos^n

$x_2 = \dot{\theta}$ ang velocity

Second order differential eq^{n}s

$\dot{x} = Ax + Bu$ $u = 0$ since no external input in this system

$y = cx + Du$

$x_1 = \theta$

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

$\dot{x}_2 = \ddot{\theta}$

isolate for $\ddot{\theta}$ to get eq^n

$ml^2\ddot{\theta} + c\dot{\theta} + mgl\sin\theta = 0$

$ml^2\ddot{\theta} = -c\dot{\theta} - mgl\sin\theta$

$$\ddot{\theta} = \frac{-c\dot{\theta} - mgl\sin\theta}{ml^2}$$

$$\ddot{\theta} = \frac{-c}{ml^2}\dot{\theta} - \frac{\cancel{m}g\cancel{l}}{\cancel{m}l^{\cancel{2}}}\sin\theta$$

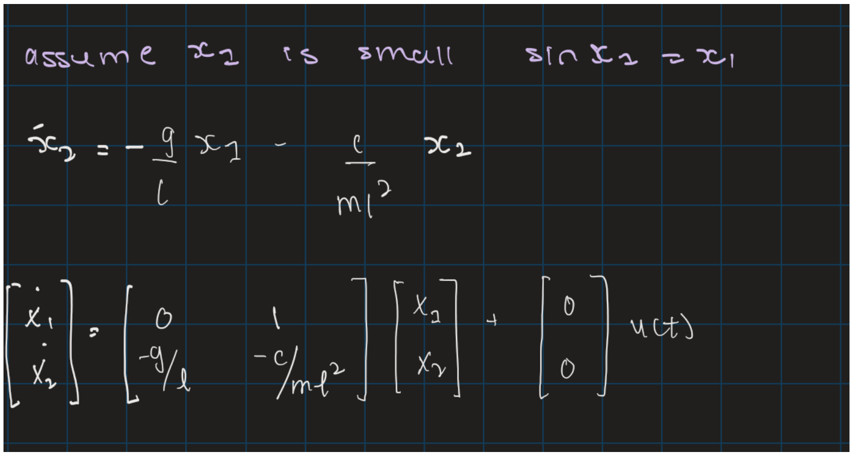
$$\underset{\dot{x}_2}{\ddot{\theta}} = \frac{-c}{ml^2}\underset{x_2}{\dot{\theta}} - \frac{g}{l}\sin\underset{x_1}{\theta}$$

$\therefore$

$\dot{x}_1 = x_2$

$$\dot{x}_2 = -\frac{g}{l}\sin x_1 - \frac{c}{ml^2}x_2$$

Figure 15 State Space Model of a Single Pendulum Handwritten Solution

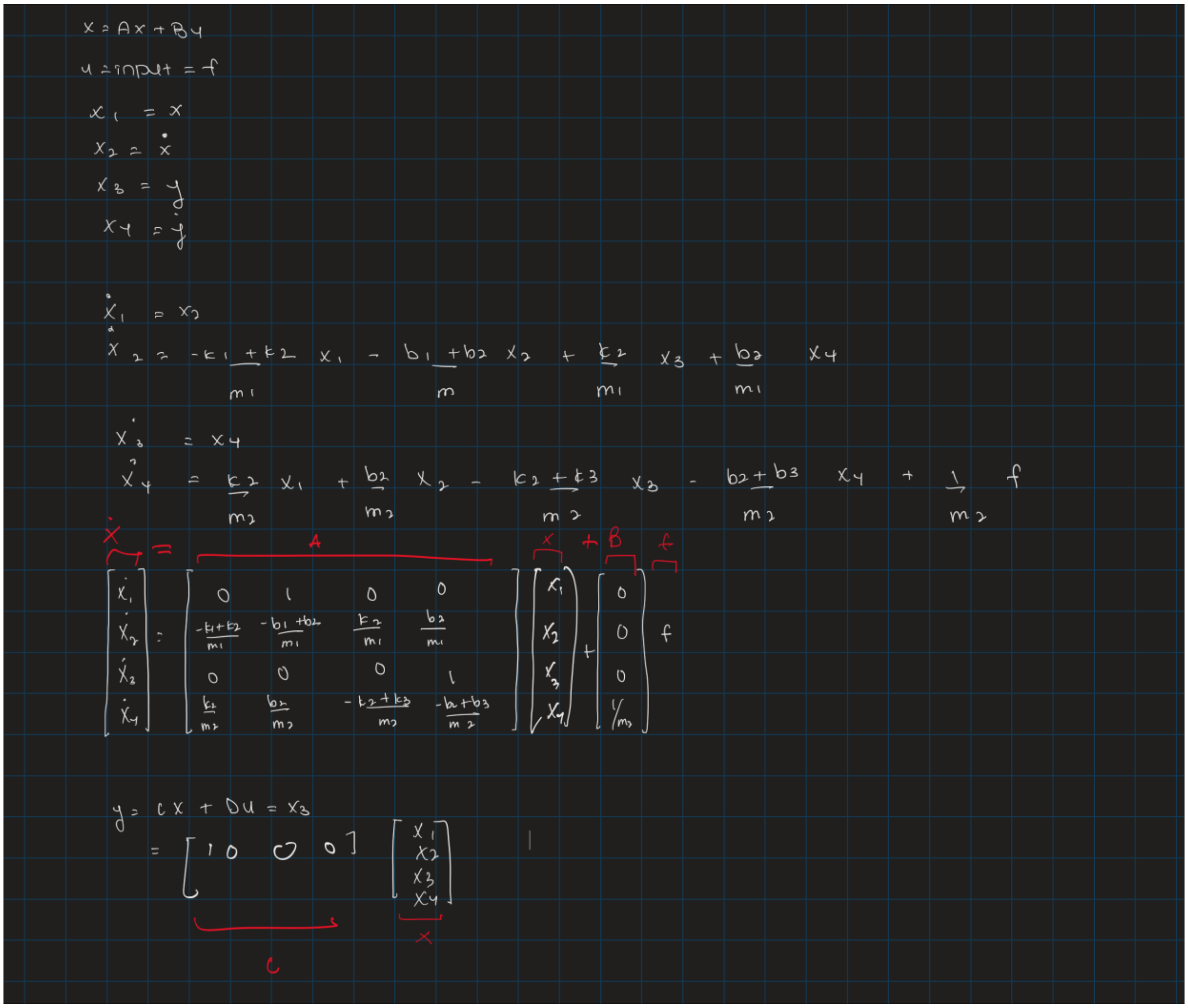


assume x_1 is small $\sin x_1 = x_1$

$$\dot{x}_2 = -\frac{g}{l}x_1 - \frac{c}{ml^2}x_2$$

$$\begin{bmatrix}\dot{x}_1\\ \dot{x}_2\end{bmatrix} = \begin{bmatrix}0 & 1\\ -g/l & -c/ml^2\end{bmatrix}\begin{bmatrix}x_1\\ x_2\end{bmatrix} + \begin{bmatrix}0\\ 0\end{bmatrix}u(t)$$

Figure 16 State Space Model of a Single Pendulum with Approximation Handwritten Solution



$x = Ax + Bu$

$u = \text{input} = f$

$x_1 = x$

$x_2 = \dot{x}$

$x_3 = y$

$x_4 = \dot{y}$

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

$$\dot{x}_2 = \frac{-k_1+k_2}{m_1}x_1 - \frac{b_1+b_2}{m}x_2 + \frac{k_2}{m_1}x_3 + \frac{b_2}{m_1}x_4$$

$$\dot{x}_3 = x_4$$

$$\dot{x}_4 = \frac{k_2}{m_2}x_1 + \frac{b_2}{m_2}x_2 - \frac{k_2+k_3}{m_2}x_3 - \frac{b_2+b_3}{m_2}x_4 + \frac{1}{m_2}f$$

$$\overbrace{\begin{bmatrix}\dot{x}_1\\ \dot{x}_2\\ \dot{x}_3\\ \dot{x}_4\end{bmatrix}}^{\dot{X}} = \overbrace{\begin{bmatrix}0 & 1 & 0 & 0\\ \frac{-k_1+k_2}{m_1} & \frac{-b_1+b_2}{m_1} & \frac{k_2}{m_1} & \frac{b_2}{m_1}\\ 0 & 0 & 0 & 1\\ \frac{k_2}{m_2} & \frac{b_2}{m_2} & -\frac{k_2+k_3}{m_2} & -\frac{b_2+b_3}{m_2}\end{bmatrix}}^{A}\overbrace{\begin{bmatrix}x_1\\ x_2\\ x_3\\ x_4\end{bmatrix}}^{X} + \overbrace{\begin{bmatrix}0\\ 0\\ 0\\ 1/m_2\end{bmatrix}}^{B}\overbrace{f}^{f}$$

$$y = Cx + Du = x_3$$

$$= \underbrace{\begin{bmatrix}1 & 0 & 0 & 0\end{bmatrix}}_{C}\underbrace{\begin{bmatrix}x_1\\ x_2\\ x_3\\ x_4\end{bmatrix}}_{X}$$

Figure 17 Matrices A,B,C, and D for Double Mass-Spring Damper System Handwritten Solution