Teagan Kilian
Lab 3.1
MEE 416
Due 11/19

Linear Inverted Pendulum

1. Introduction

There are many situations in which mechanical engineers may need to solve a control problem such as keeping an autonomous vehicle on the road, ensuring that a building's air temperature will remain at a certain temperature, or when dealing with energy systems to ensure that too much or too little energy is never supplied to the system. The purpose of this lab was to investigate a control problem with the objective of keeping an inverted pendulum balanced atop a frictionless cart system. During this lab, we were able to simulate the situation using a linear state-space representation of the open-loop system, design a state-feedback controller, and implement these controllers while evaluating their performance.

2. Experimental Facility and Instrumentation

The IP02 linear pendulum system was utilized to carry out this experiment. This system was used in the single inverted pendulum configuration (SIP). The pendulum is attached to the IP02 solid aluminum cart using a pivot on the front of the cart so there can be 360 degrees of rotational freedom. The angle from the vertical is measured using the quadrature incremental encoder. The cart utilizes a rack and pinion and a 6 Volt DC motor to move along a stainless steel shaft using linear bearings. There is a second pinion attached to the rack which holds a sensor that measures the position of the cart. One must ensure that there is at least 1.27 m of height and 2.19 m of width surrounding the pendulum system to ensure safe operation. The experimental facility is also required to have Matlab software in order to control the motion of the cart as well as to receive the data obtained from the sensors on the cart.

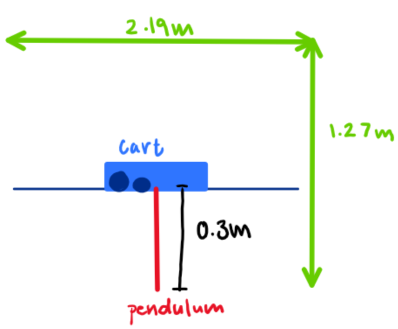

Figure 1: *Schematic showing the experimental setup.*

3. Method

This entire experiment is based on a closed-loop (state-feedback) control system where a linear control system is represented by u = -Kx. In this case, the equation of state is given in equation 1. The linear quadratic regulator method (LQR) is utilized in order to determine the optimum K value. The cost function (equation 2) is to be minimized which is where the Q matrix and the R value comes in. These two variables are called tuning variables because they affect how the cost function is able to be minimized. By running simulations, one can determine how these variables affect the cost function and the overall output.

$$\dot{x} = Ax + B(-Kx)$$
$$= (A - BK)x$$

Equation 1: *State-space equation*

$$J = \int_0^{\infty} x'(t)Qx(t) + u(t)'Ru(t)\,dt$$

Equation 2: *Cost function*

The values of interest for this experiment were obtained using different methods for the simulation and the physical experiment.The main goal of these experiments were to ensure that the following parameters were met: the rise time is less than or equal to 1.5s, the absolute value of the maximum pendulum angle was less than or equal to 1.0°, and the absolute value of the control effort never went above 10 volts. For both experiments, a 4x4 diagonal matrix representing the variable Q was input into the code. A value was also input for the R variable. Figure 2 depicts the control process used within the Matlab code to simulate the motion of the pendulum. The code then returns graphs with the pendulum angle, applied voltage, and cart position all in the time domain. After viewing the results to see if they fulfill the requirements, multiple iterations were performed by changing the Q and R variables and running the simulation until favorable results were obtained. The state-feedback gain (K) is read from the Matlab workspace and recorded each time a new simulation is run. The amplitude (m) is controlled in order to dictate the cart's position along the rod. One important block located in figure 2 is the Find State X block which determines the velocity of the rotary arm and pendulum utilizing high pass filters. Additionally, the State-Space blocks read A, B, C, and D which are state space matrices which are defined as the outputs.

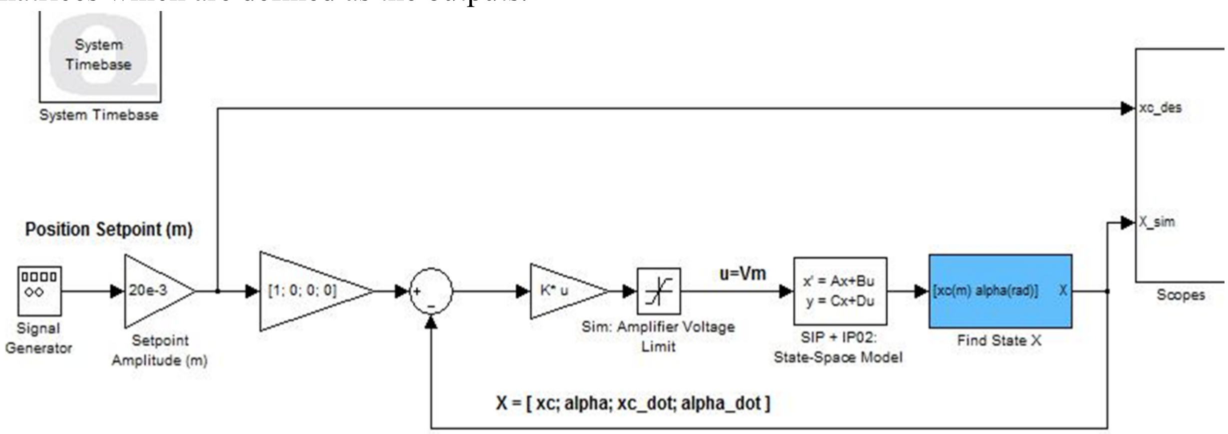

Figure 2: *s_sip_lqr Simulink diagram used to simulate the linear pendulum system in Matlab.*

In the physical experiment, a Simulink program named q_sip_lqr was used to run the actual experiment and a simulated experiment in parallel with the same given input values. In the case of physical implementation, Simulink interfaces with the sensors on the cart which communicate with the motors in order to monitor and adjust the pendulum angle and the cart position.

For all experimental runs, feedback control is used to adjust the cart in order to keep the pendulum upright and to minimize the pendulum angle. For this situation, the state is defined in equation 3. The desirable outcome is shown in equation 4 where all parameters except for the cart position are zero (these being cart velocity, pendulum angle, and pendulum angular velocity).

$$x^T = \begin{bmatrix} x_c & \alpha & \dot{x}_c & \dot{\alpha} \end{bmatrix}$$

Equation 3: *State equation for the linear inverted pendulum problem.*

$$x_d = \begin{bmatrix} x_{cd} & 0 & 0 & 0 \end{bmatrix}$$

Equation 4: *Optimal outcome for the state equation where* x_{cd} *is the optimal cart position*

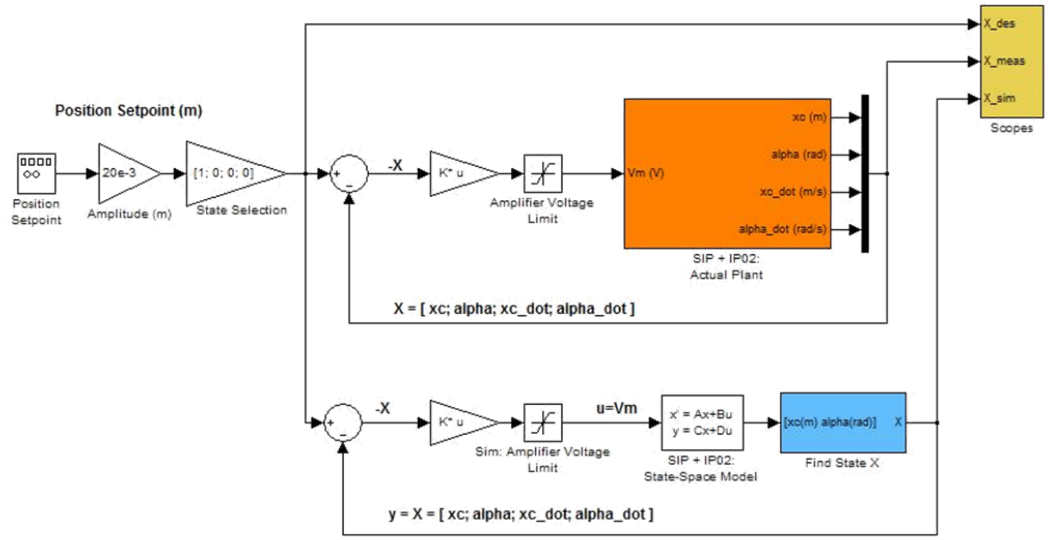

Figure 3: *q_sip_lqr Simulink diagram representing the process of implementation in this experiment.*

4. Experimental Design

Throughout the course of this experiment, my lab section carried out 11 trials in order to best accomplish the requirements outlined in the previous section. Figure 4 displays each Q matrix and R value that were inputted into Matlab during our simulation portion. The table also displays all of the associated K values that were given as the output of our simulation.

Trial #	Diagonal Values of Q Matrix	R	K
1	[1, 1, 1, 1]	0.1	[-3.16, 47.67, -16.25, 10.22]
2	[10, 1, 1, 1]	0.1	[-10, 55.93, -19.70, 17.88]
3	[20, 1, 1, 1]	0.1	[-14.14, 60.81, -21.76, 12.86]
4	[20, 100, 1, 1]	0.1	[-14.14, 72.07, -22.83, 13.64]
5	[20, 400, 1, 1]	0.1	[-14.14, 96.39, -24.90, 15.20]
6	[20, 400, 100, 0]	0.1	[-14.14, 172.53, -46.10, 33.41]
7	[40, 0.1, 0.1, 10]	0.1	[-20, 84.50, -26.28, 20.11]
8	[40, 400, 0.1, 10]	0.1	[-20, 113.37, -29.12, 21.80]
9	[40, 0.1, 0.1, 10]	10	[-2, 42.07, -14.96, 8.53]

10	[40, 0.1, 0.1, 10]	0.001	[-200, 998.26, -203.56, 189.32]
11	[40, 0.1, 0.1, 10]	0.5	[-8.94, 64.29, -19.48, 12.63]

Figure 4: *Trials run during the simulation portion of the lab.*

My lab group ended up running the implementation trial with the Q and R values used for the 11th trial of the simulation since the response of voltage, angle, and cart position were quite favorable.

The experimental conditions that were required for the implementation are the cart beginning in the centered position and the motor not engaging for the balance control until the pendulum was manually brought up to within 10° of the perfectly vertical position. Given that ε is set at 10° for this experiment, equation 5 shows the equation for this controller.

$$u = \begin{cases} K(x_d - x) & |x_2| < \varepsilon \\ 0 & otherwise \end{cases}$$

Equation 5: *Controller that determines when they balance control shall begin being implemented.*

5. Experimental Procedure

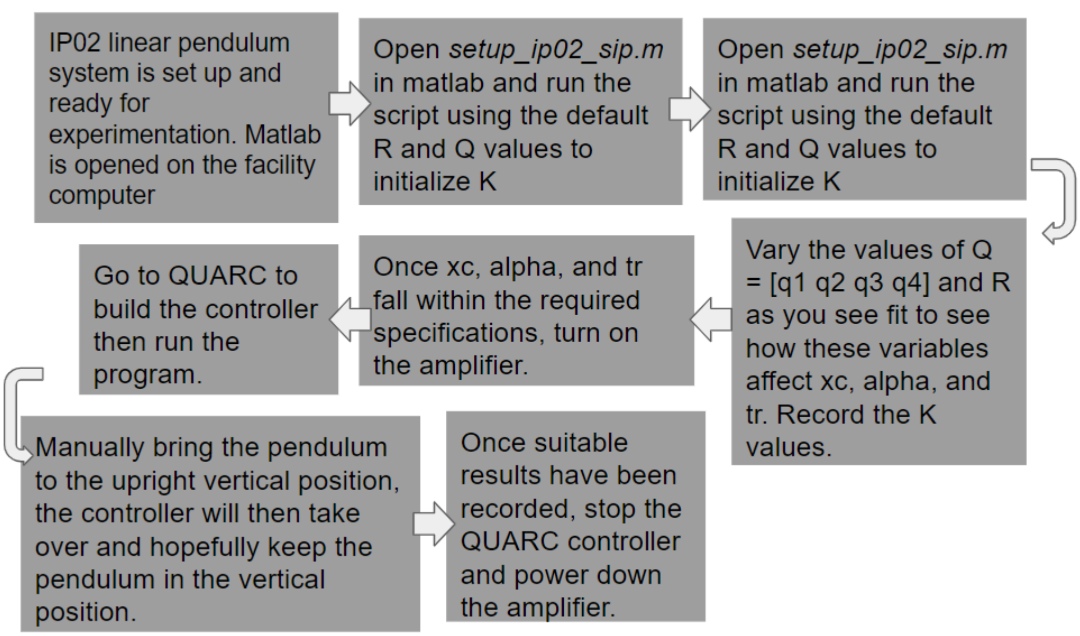

Figure 5: *Brief description of experimental procedure*

6. Results and Discussion

As stated in the lab manual, using the system parameters, the eigenvalues of matrix A come out to be -16.26, -4.56, 4.84, and 0. These values represent the open loop poles of our open loop system. Since there is at least one positive eigenvalue (4.84), this tells us that the open loop system is unstable. Due to the fact that this system is unstable, we can conclude that we will need a closed loop feedback control to bring all poles to the left side of the plane making the system stable.

$$J = \int_0^{\infty} [x_1 \quad x_2 \quad x_3 \quad x_4] \begin{bmatrix} q_1 & 0 & 0 & 0 \\ 0 & q_2 & 0 & 0 \\ 0 & 0 & q_3 & 0 \\ 0 & 0 & 0 & q_4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + u(t)'Ru(t)dt$$

Equation 5: *The cost function with Q and x as a matrix*

The previous sections state that Q and R are considered tuning variables in the cost function so changing them determines how the optimization will occur and thus, the K values that will result. It is evident from our iterations that R has a negative correlation with the absolute value of all the K values; however, this action has a positive correlation on the actual value of q_1 and q_3 since they are negative values. As stated by LQR optimization theory, increasing R will increase J which we now know will decrease the absolute value of all K entries. From running the simulation and viewing the graphs that followed, we were able to deduce that q_1 is the variable that affects the time response and adjustments made to q_2 will adjust the pendulum angle response. It was not evident what specifically q_3 and q_4 controlled as they were adjusted.

It is known that a smaller u value is needed when R is larger (giving more weight to the control input) in order to minimize J. It is also known that increasing Q will cause the minimization algorithm to need to work harder and therefore will require a larger u value. Since the linear control system equation shows us that the absolute values of u and K have a positive correlation, this allows us to assume that increasing R decreases the absolute values of K as well

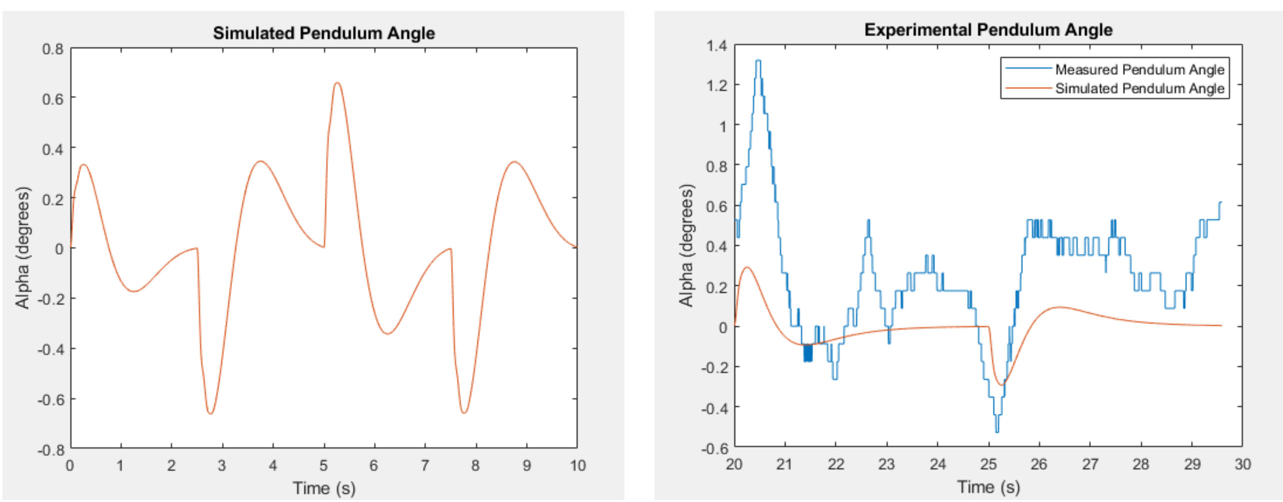

Figure 6: *Left - graph depicting the simulated pendulum angle. Right - graph depicting the experimental pendulum angle and the parallel simulated results.*

as increasing values of Q shall increase the absolute values of K. This result seems to fall in line with what was recorded during our simulations. One can see that
In figure 5 - which utilized the inputs written under trial 11 of figure 4 - we are able to compare the simulated results, the physical implementation results, and the results of the simulation that ran in parallel with the implementation. Both simulated results are fairly similar however, the time response for the parallel simulation is far slower than in the first simulation. Regardless, the overall shape of each graph is very similar. The physical implementation graph has much more noise than the simulated graphs which is to be expected due to certain conditions in the room that

one is not able to control for such as air blowing on the system, or the table being unstable. The first peak of the experimental pendulum angle is slightly higher than any angle on either simulation but this could be the effect of one of the participants who brought the pendulum to the upright position accidentally exerting a force on the pendulum as the motor took over.

During the physical implementation, the cart position veered way off to one side. This was not foreseen in any of the simulations but this result makes sense when looked at next to the pendulum angle results. As mentioned previously, there is a very large angular deflection in the positive direction at the beginning of the time domain. In order to correct this, the controls in the cart would have driven it farther in the positive x direction as well. The control system recognized that there was a disturbance in the pendulum and it made the correction in order to fulfill the requirements that we put in place.

In the simulation, the graphs show that the pendulum angle does not exceed an absolute value of 0.6619°. In the implementation, the angle reaches a value of 1.3189° therefore this physical experiment did not stay within the specifications given in the lab manual. The maximum absolute value of voltage in the simulation is only 1.2011 V and the maximum voltage for the implementation is 5.813 V. The voltage values for both the simulation and the implementation stayed within the bounds provided. The rise time in the simulation and implementation were calculated to be 0.966s and 2.874s respectively. Again, the simulation results fall within the desired specifications but the implementation results do not.

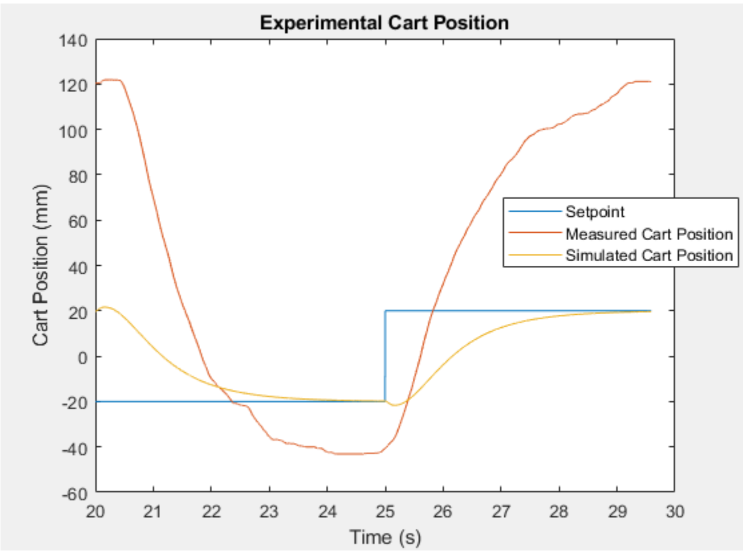

Figure 7 : *Graph depicting the experimental cart position*

7. Conclusion and Recommendations

This lab taught me a lot about control systems. My main takeaway from this experiment was that even if you think a simulation is very accurate and it gives you a favorable result, there may be some variables that you are not aware of that will heavily impact your physical implementation results. I have learned that in the future, physical implementations should be used alongside the simulations during the iterative development to ensure that the results will come out as expected.

In order to improve this experiment, I suggest having smaller lab groups so people can individually see what each change to the Q matrix does to the K matrix. When there was a large group it was hard to know what the person on the computer was manipulating.

8. References

1) MEE 416 Mechanical Engineering Laboratory - Student Manual Lab T3.1 Linear Inverted Pendulum
2) Quanser Inc. QUARC User Manual.
3) Quanser Inc. IP02 QUARC Integration, 2008.
4) Quanser Inc. IP02 User Manual, 2009.
5) Quanser Inc. QUARC Installation Guide, 2009.
6) Quanser Inc. SIP and SPG User Manual, 2009.
7) Quanser Inc. Linear Pendulum Gantry Workbook, 2012.
8) K. J. strm and K. Furuta. Swinging up a pendulum by energy control. 13th IFAC World Congress, 1996.

My group's presentation is attached to my lab submission. I contributed to the experimental procedure slide.

---

9. Appendix: Additional Information Linear Inverted Pendulum Lab

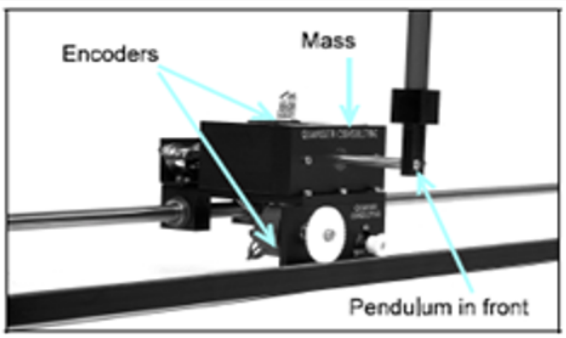

Figure A1: *Physical configuration of IP02 cart and pendulum system.*

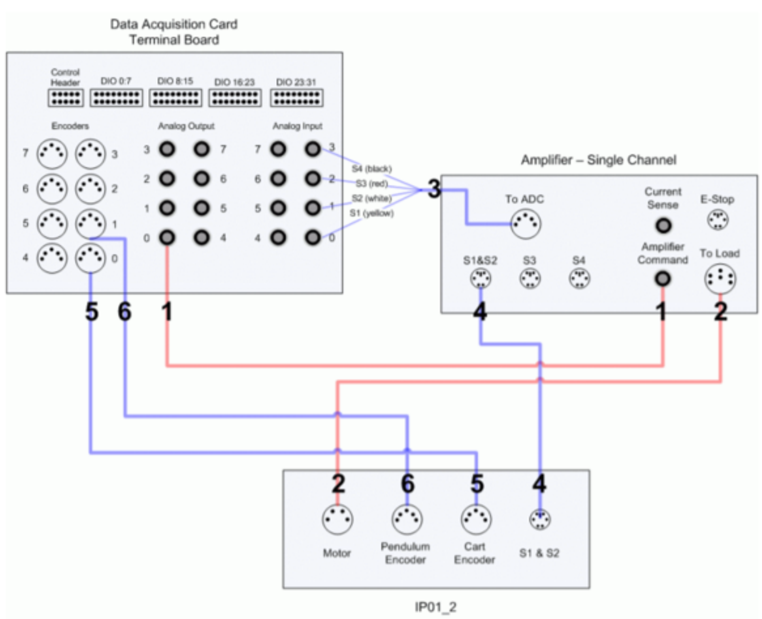

Figure A2: *How the IP02 system communicates with the data acquisition system and the code in order to output the correct motion.*

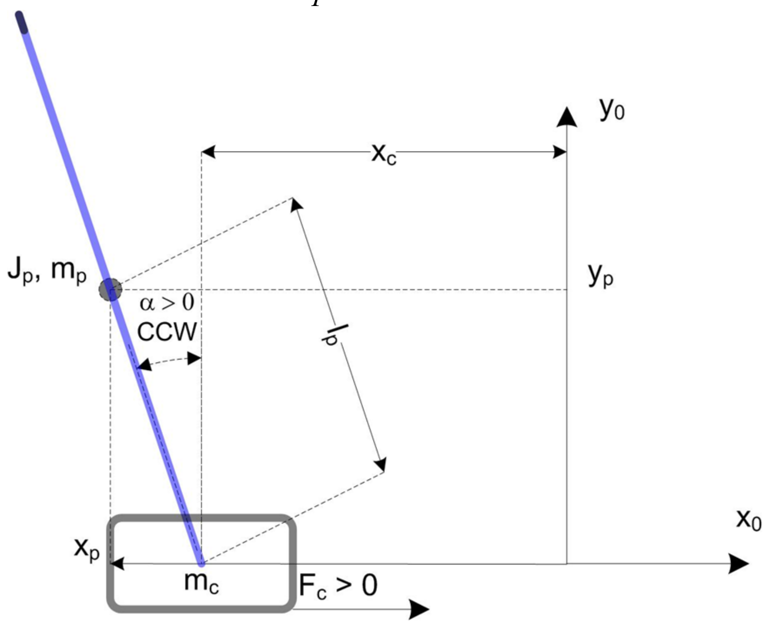

Figure A3: *Linear inverted pendulum schematic*

$$Q = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Equation A1: *Initial Q matrix*

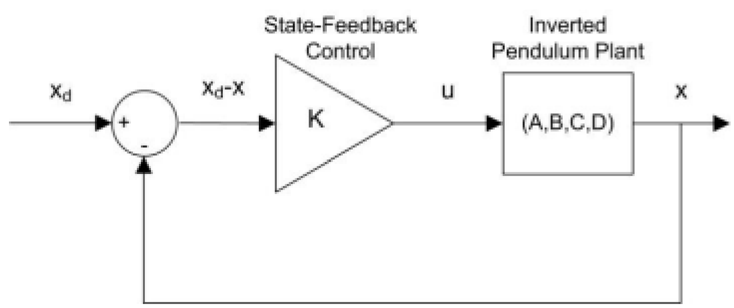

Figure A4: *State-feedback control loop such as the one used in this experiment.*

$$\dot{x}_3 = \frac{1}{J_T}\left(-(J_p + M_p l_p^2) B_{eq} x_3 - M_p l_p B_p x_4 + M_p^2 l_p^2 g x_2 + (J_p + M_p l_p^2) u\right)$$

Equation A2: *Linear velocity of the cart*

$$\dot{x}_4 = \frac{1}{J_T}\left(-(M_p l_p B_{eq})x_3 - (J_{eq} + M_p)B_p x_4 + (J_{eq} + M_p)M_p l_p g x_2 + M_p l_p u\right)$$

Equation A3: *Angular velocity of the pendulum*

$$\ddot{x}_c = \frac{1}{J_T}\left(-(J_p + M_p l_p^2\right) B_{eq}\dot{x}_c - M_p l_p B_p \dot{\alpha} + M_p^2 l_p^2 g\alpha + (J_p + M_p l_p^2)F_c)$$

Equation A4: *Linear acceleration of the cart*

$$\ddot{\alpha}_c = \frac{1}{J_T}(-(M_p l_p B_{eq})\dot{x}_c - (J_{eq} + M_p)B_p\dot{\alpha} + (J_{eq} + M_p)M_p l_p g\alpha + M_p l_p F_c)$$

Equation A5: *Angular acceleration of the pendulum*

$$A = \frac{1}{J_T}\begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & M_p^2 l_p^2 g & -(J_p + M_p l_p^2)B_{eq} & -M_p l_p B_p \\ 0 & (J_{eq} + M_p)M_p l_p g & -M_p l_p B_{eq} & -(J_{eq} + M_p)B_p \end{bmatrix}$$

Equation A6: *A matrix equation as in the state-space representation*

$$B = \frac{1}{J_T}\begin{bmatrix} 0 \\ 0 \\ J_p + M_p l_p^2 \\ M_p l_p \end{bmatrix}$$

Equation A7: *B matrix equation as in the state-space representation*

$$J_{eq} = M_c + \frac{\eta_g K_g^2 J_m}{r_{mp}^2}$$

Equation A8: *Equation for variable utilized in the equations of motion*

$$J_T = J_{eq} J_p + M_p J_p + J_{eq} M_p l_p^2$$

Equation A9: *Equation for variable utilized in the equations of motion*

Symbol	Description	Value
M_{pl}	Long Pendulum Mass (with T-fitting)	0.230 kg
M_{pm}	Medium Pendulum Mass (with T-fitting)	0.127 kg
L_{pl}	Long Pendulum Full Length (from Pivot to Tip)	0.6413 m
L_{pm}	Medium Pendulum Full Length (from Pivot to Tip)	0.3365 m
l_{pl}	Long Pendulum Length from Pivot to Center Of Gravity	0.3302 m
l_{pm}	Medium Pendulum Length from Pivot to Center Of Gravity	0.1778 m
I_{pl}	Long Pendulum Moment of Inertia, about its Center Of Gravity	7.88 10 ^{3} kg.m^2
I_{pm}	Medium Pendulum Moment of Inertia about its Center Of Gravity	1.20 10 ^{3} kg.m^2
B_p	Viscous Damping Coefficient, as seen at the Pendulum Axis	0.0024 N.m.s/rad
g	Gravitational Constant on Earth	9.81 m/s^2

Figure A5: *Pendulum specifications associated with SPG and SIP configurations.*

Symbol	Description	Value	Variation
V_{nom}	Motor nominal input voltage	6.0 V	
R_m	Motor armature resistance	2.6 Ω	$\pm$ 12%
L_m	Motor armature inductance	0.18 mH	
k_t	Motor current-torque constant	7.68×10^{-3} N m/A	$\pm$ 12%
k_m	Motor back-emf constant	7.68×10^{-3} V/(rad/s)	$\pm$ 12%
η_m	Motor efficiency	0.69	$\pm$ 5%
$J_{m,rotor}$	Rotor moment of inertia	3.90×10^{-7} kg $\cdot$ m^2	$\pm$ 10%
K_g	Planetary gearbox gear ratio	3.71	
η_g	Planetary geabox efficiency	0.90	$\pm$ 10%
M_c	Mass of cart	0.38 kg	
M_w	Mass of cart weight	0.37 kg	
$B_{eq,c}$	Equivalent viscous damping coefficient (Cart)	4.3 N m s/rad	
$B_{eq,c}$	Equivalent viscous damping coefficient (Cart and Weight)	5.4 N m s/rad	
L_t	Track length	0.990 m	
T_c	Cart travel	0.814 m	
P_r	Rack pitch	1.664×10^{-3} m/tooth	
r_{mp}	Motor pinion radius	6.35×10^{-3} m	
N_{mp}	Motor pinion number of teeth	24	
r_{pp}	Position pinion radius	0.01483 m	
N_{pp}	Position pinion number of teeth	56	
K_{ec}	Cart encoder resolution	2.275×10^{-5}	
K_{ep}	Pendulum encoder resolution	0.0015 rad/count	
f_{max}	Maximum input voltage frequency	50 Hz	
I_{max}	Maximum input current	1 A	
ω_{max}	Maximum motor speed	628.3 rad/s	

Figure A6: *Experimental set up specifications*

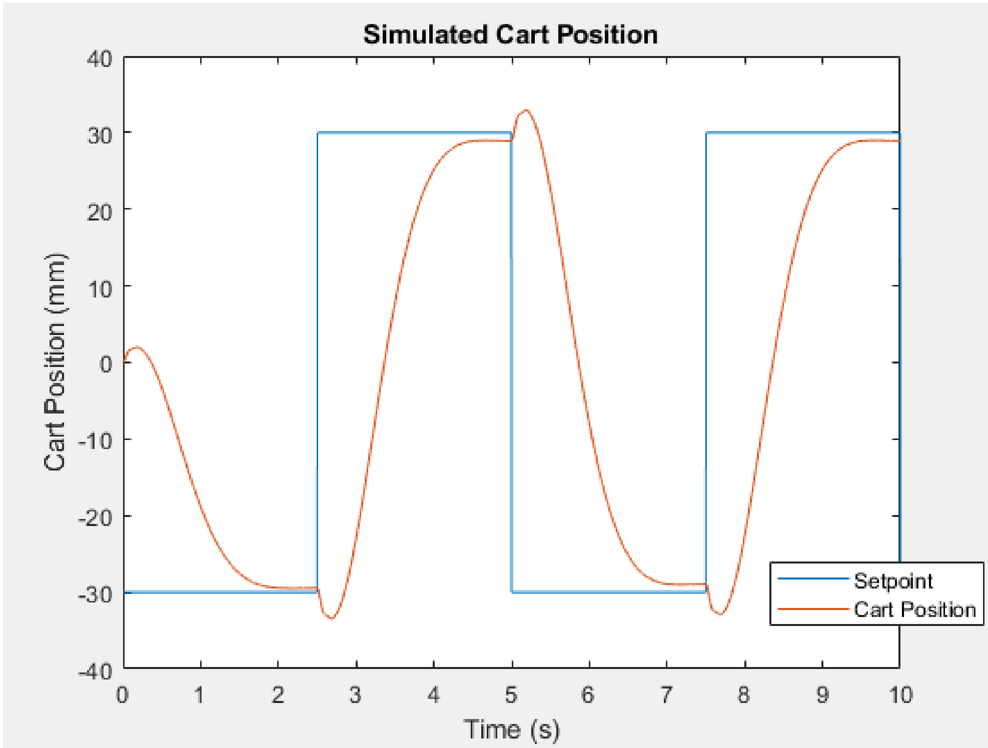

Figure A7: *Graph depicting simulated cart position*

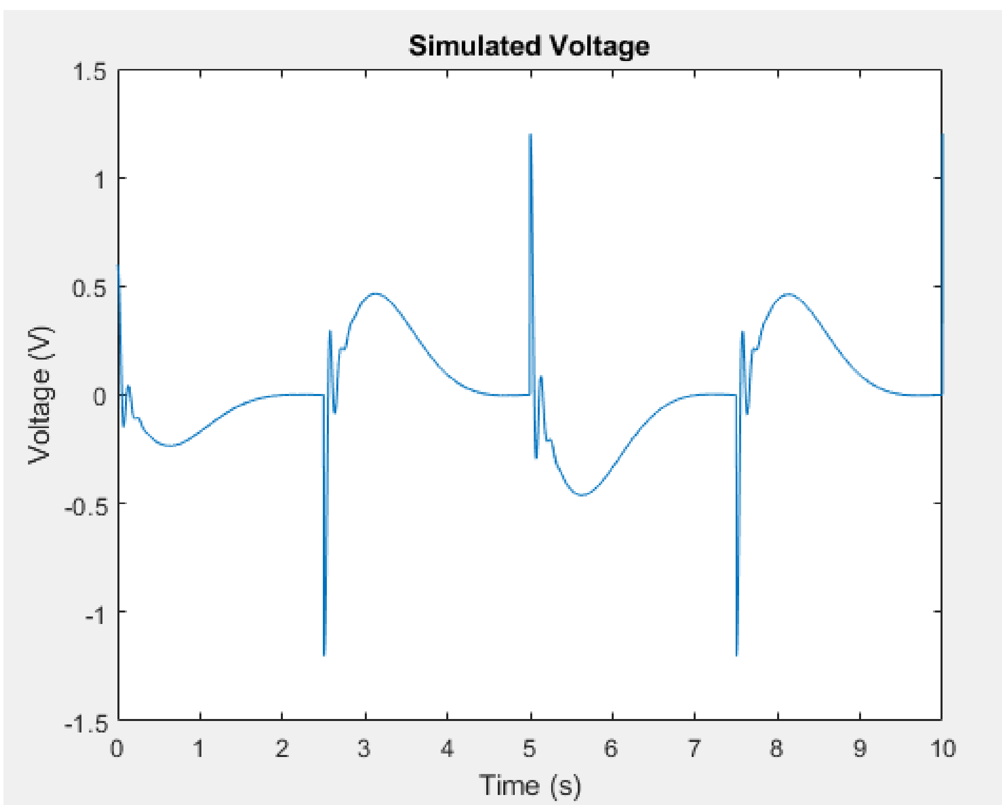

Figure A8: *Graph depicting simulated voltage values*

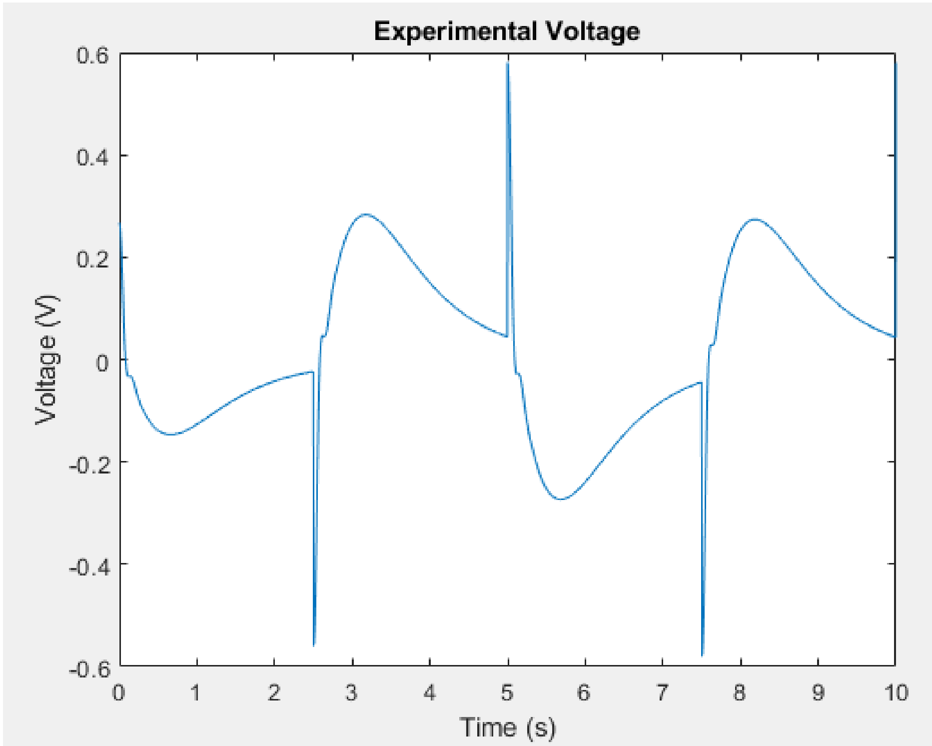

Figure A9: *Graph depicting experimental voltage values*

Detailed Procedure:

<u>Control Simulation</u>

1. Open setup_ip02_sip.m and go down to the LQR Control section
2. Run the script with the default values of Q and R to generate the default gain K.
3. Run s_sip_lqr to simulate the closed-loop response with this gain.
4. Vary the value of the R parameter and observe its effect on the response of the system.
5. If Q = diag[q1, q2, q3, q4], vary each qi independently and examine its effect on the gain and the closed-loop response. For example, when increasingq3, what happens to xc and α? Vary each qi by the same order of magnitude and compare how the new gain K changes compared to the original gain. Keep R = 0.1 throughout your testing. Summarize your results.

   Note: Recall your analysis in pre-lab Question 3 where the effect of adjusting Q on the generated K was assessed generally by inspecting the cost function equation.
6. Find a Q and R that will satisfy the specifications given in section d) Specifications. When doing this, don't forget to keep the DC motor voltage as small as possible within 10 V. This control will later be implemented on actual hardware. Enter the weighting matrices, Q and R, used and the resulting gain, K.

Note: Recall that the most important design objective for the controller is to maintain the pendulum in the balanced inverted position.

7. Plot the responses from the xc (mm), alpha (deg), and Vm (V) scopes in a Matlab figure. When the QUARC controller is stopped, these scopes automatically save the last 5 seconds of their response data to the variables data_xc, data_alpha, and data_vm. For data_xc, the time is in data_xc(:,1), the setpoint (desired cart position) is in data_xc(:,2), and the simulated cart position is in data_xc(:,3). In the data_alpha and data_vm variables, the (:,1) holds the time vector and the (:,2) holds the actual measured data.

8. Measure the rise time of the simulated cart position response and the maximum pendulum angle. Does the response satisfy the specifications given in the above section?

<u>Control Implementation</u>

1. Run the setup_ip02_sip script.
2. Make sure the gain K you found in the section above is loaded.
3. Open the s_sip_lqr Simulink diagram.
4. Turn ON the power amplifier.
5. Go to QUARC | Build to build the controller.
6. Ensure the pendulum is in the hanging down position and is motionless. Go to QUARC | Run to start the controller. Once it is running, manually bring up the pendulum to its upright vertical position. You should feel the voltage kick-in when it is within the range where the balance control engages. Once it is balanced, introduce the 30 mm cart position command by setting the Amplitude (mm) gain in the Simulink diagram to 30.

   Note: Once the controller has engaged, do not attempt to manually lower the pendulum. If the pendulum or cart moves outside of a safe workspace, the system watchdog should halt the controller automatically.
7. The response should look similar to your simulation. Once you have obtained a suitable response, go to QUARC | Stop to stop the controller. Be careful, as the pendulum will fall down when the controller is stopped. Similar to the simulation, the response data will be saved to the workspace. Use this to plot the cart, pendulum, and control input responses in a Matlab figure.
8. Measure the rise time of the cart position, pendulum deflection and voltage used. Are the specifications given in the above section satisfied for the implementation?
9. Shut off the power amplifier.

Matlab Code:

```
% Teagan Kilian
% MEE 416
% Lab 3.1

% Simulation
figure (1)
t_a1 = data_alpha1(:,1);
alpha1 = data_alpha1(:,2);
plot(t_a1,alpha1)

title ("Simulated Pendulum Angle")
xlabel ("Time (s)")
ylabel ("Alpha (degrees)")

figure (2)
t_vm1 = data_vm1(:,1);
vm1 = data_vm1(:,2);
plot(t_vm1,vm1)

title ("Simulated Voltage")
xlabel ("Time (s)")
ylabel ("Voltage (V)")

figure (3)
t_xc1 = data_xc1(:,1);
xc1 = data_xc1(:,2);
plot(t_xc1,xc1)
hold on

xc12 = data_xc1(:,3);
plot(t_xc1,xc12)

title ("Simulated Cart Position")
xlabel ("Time (s)")
ylabel ("Cart Position (mm)")
legend ("Setpoint", "Cart Position")
```

```
% Experiment
figure (4)
t_a4 = data_alpha4(:,1);
alpha4 = data_alpha4(:,2);
plot(t_a4,alpha4)

hold on

alpha42 = data_alpha4(:,3);
plot(t_a4,alpha42)

legend ("Measured Pendulum Angle", "Simulated Pendulum Angle")

title ("Experimental Pendulum Angle")
xlabel ("Time (s)")
ylabel ("Alpha (degrees)")

figure (5)
t_vm4 = data_vm4(:,1);
vm4 = data_vm4(:,2);
plot(t_vm4,vm4)
```

```
title ("Experimental Voltage")
xlabel ("Time (s)")
ylabel ("Voltage (V)")

figure (6)
t_xc4 = data_xc4(:,1);
xc4 = data_xc4(:,2);
plot(t_xc4,xc4)
hold on

xc42 = data_xc4(:,3);
plot(t_xc4,xc42)

xc43 = data_xc4(:,4);
plot(t_xc4,xc43)

title ("Experimental Cart Position")
xlabel ("Time (s)")
ylabel ("Cart Position (mm)")
legend ("Setpoint", "Measured Cart Position", "Simulated Cart Position")
```