Shreya Jampana, Audrey Gruian
4/25/25
ENGR102

Inverted Pendulum Project

I. Introduction

The objective of this project is to simulate the movement and stability of an inverted pendulum atop a cart moving from an initial rest position to a final rest position. The system is modeled using time-dependent cart position s(t) and time-dependent pendulum angular displacement θ(t). The pendulum is subjected to an angular acceleration disturbance of α(t) = 0.5 rad/sec^2, the cart has no more than 10 seconds to travel from the initial to the final position, the maximum acceleration of the cart is |a(t)| < 0.5 m/sec^2, and the system should not overshoot the final position. The pendulum length is L = 0.5 m. The cart-pendulum system can be described using the equations:

$$L\frac{d^2\theta(t)}{dt^2} - g \cdot sin\theta(t) = -a(t) \cdot cos\theta(t) + L \cdot \alpha(t) \text{ [1]}$$

$$\frac{d^2s(t)}{dt^2} = a(t) \text{ [2]},$$

from which we derived a state space design for a linear controller and a linear plant. While the cart-pendulum system is inherently unstable, our objective was to design a controller and an observer for the system to achieve stability and keep the inverted pendulum relatively upright as the cart moves from initial to final position.

II. State Space Design of a Linear Controller for a Linearized Plant

As described by equations 1 and 2 above, the cart pendulum plant is inherently nonlinear. This system was linearized using small angle approximation, in which $sin\theta(t) \approx \theta(t)$ and $cos\theta(t) \approx 1$. This resulted in the following linear differential equations to describe the plant:

$$L\frac{d^2\theta(t)}{dt^2} - g\theta(t) = -a(t) + L\alpha(t) \text{ [3]}$$

$$\frac{d^2s(t)}{dt^2} = a(t) \text{ [2]}.$$

After linearizing the system, a linear state space model was used to describe the system:

$$\frac{dx(t)}{dt} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ g/L & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \theta(t) \\ \frac{d\theta(t)}{dt} \\ s(t) \\ \dot{s}(t) \end{bmatrix} + \begin{bmatrix} 0 \\ -1/L \\ 0 \\ 0 \end{bmatrix} a(t) + \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} \alpha(t) \quad [4]$$

$$\begin{bmatrix} \theta(t) \\ s(t) \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \theta(t) \\ \frac{d\theta(t)}{dt} \\ s(t) \\ \dot{s}(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix} a(t) + \begin{bmatrix} 0 \\ 0 \end{bmatrix} \alpha(t) \quad [5]$$

Using Matlab's isstable(system) command, we found the system to be unstable; the controllability matrix Mc of the system to have full rank n = 4, suggesting the system is controllable; the observability matrix Mo of the system to have full rank n = 4, suggesting the system is observable. Given the system is controllable and observable, we set out to design a feedback control system with an observer.

Due to the system being subject to uncontrolled angular acceleration disturbance, we introduced integral action to make the system robust and eliminate steady state error due to disturbance. We augmented the state vector with an extra state variable x_I: the integral of the error signal, constructing augmented state space equations:

$$\begin{bmatrix} \frac{dx_I}{dt} \\ \frac{dx}{dt} \end{bmatrix} = \begin{bmatrix} 0 & -C \\ 0 & A \end{bmatrix} \begin{bmatrix} x_I \\ x \end{bmatrix} + \begin{bmatrix} -D \\ B \end{bmatrix} u + \begin{bmatrix} 1 \\ 0 \end{bmatrix} r + \begin{bmatrix} -D \\ B \end{bmatrix} w \quad [6]$$

$$y = [0 \; C] \begin{bmatrix} x_I \\ x \end{bmatrix} [7].$$

For our system, the augmented state space matrix in equation 6 is:

$$A_a = \begin{bmatrix} 0 & -C \\ 0 & A \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & g/L & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

and the augmented input matrix Ba in equation 6 is:

$$B_a = \begin{bmatrix} -D \\ B \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -1/L \\ 0 \\ 1 \end{bmatrix}.$$

It is important to note that only the second row of the C matrix was used to construct Aa and Ba, as this row corresponds to the position of the cart, which we were trying to control.

To find the desired poles of the system controller, we first selected values for damping coefficient ζ and natural frequency w_n. Because our system must not exhibit overshoot, but must still be underdamped in order to use the settling time equation, we selected a ζ value infinitesimally close to critically damped, or 1. With the selected $\zeta = 0.9999$, the given the settling time constraint of 10 seconds, and the formula for 1% settling time, w_n was computed to be the following: $w_n = \frac{4.6}{10 \cdot \zeta} = 0.4600$. Two system poles, p1 and p2, were also computed:

$$p1, \; p2 = -\zeta w_n \pm j w_n \sqrt{1 - \zeta^2} = -0.46 \pm 0.0065j \quad [8]$$

Based on these system poles, we selected three additional, faster, real poles: -4.6, -4.7, and -4.8. We chose these poles to be arbitrarily ten times faster than the dominant ones and purely real in order to avoid unwanted oscillations. Using the Matlab place command (place(Aa, Ba, p_desired)), a vector containing integral gain Ki and controller gain vector K was found, where Ki = -1.1206 and K = [-52.3151, -11.8171, -5.5866, -8.6143]. The place command computes these gains such that the feedback control law $u = [K_I \; -K] \begin{bmatrix} x_I \\ x \end{bmatrix}$ places the poles at the desired locations.

We then designed an observer based on the observer equation

$$\frac{dxhat}{dt} = A \cdot xhat + B \cdot u + L \cdot (y - yhat) \; [9].$$

Since we can't measure all the state variables of a system, the observer uses the measured output y and the calculated input u to estimate the state variables x_hat. By designing this observer, we ensure that we have estimates of *all* the state variables in order to apply the control law. The poles of the observer must

be faster than the fastest desired poles of the system in order for the estimated state to converge quickly to the actual state. Due to this fact, in designing observer gain L, we first chose our observer desired poles to be faster than the fastest controller desired poles: p1 = -6, p2 = -7, p3 = -8, p4 = -9. We found the observer gain to be the following using the Matlab place command with the original system matrices A and C (as we are observing the current state):

$$L = \begin{bmatrix} 14.66, 72.64, 0.86, 6.42 \\ 1.02, 7.76, 15.34, 57.95 \end{bmatrix}.$$

III. Simulation of Control of Nonlinear Plant with the Linear Controller

In Simulink, we first created a model of the nonlinear cart-pendulum plant as a subsystem within the larger control system. The cart-pendulum system takes acceleration and disturbance as inputs, and outputs cart-pendulum position s(t) and pendulum displacement $\theta(t)$ according to the system equations. We also created a subsystem of the observer that takes in the output y and input u to output an estimate of the state variables. These models can be found in the Appendix below.

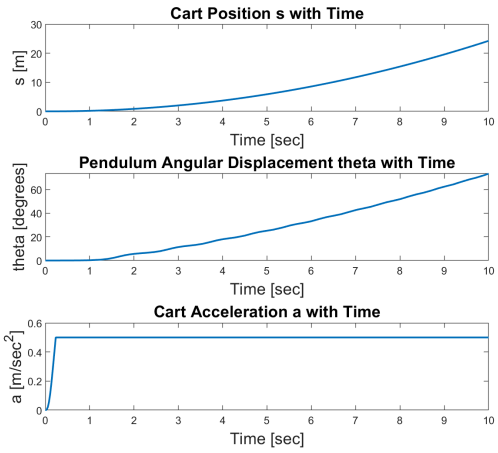

Figure 1 (left): Graphs of cart position, pendulum angular displacement, and cart acceleration with respect to time for initial observer with poles p = [-6, -7, -8, -9]. Graphs of cart position vs. time and the graph of angular displacement both exhibit infinite increases, while the graph of acceleration shows a rapid increase towards and constant maximum acceleration of 0.5 m/s^2. These observations led us to believe that the system is still unstable, as we expect a stabilization of angular displacement around 0 when the pendulum is close to straight upright. During the simulation phase, we adjusted L to meet the system specifications.

Seeing as our current system did not meet the specified requirements for overshoot and stability as shown in figure 1, we adjusted our design by changing the observer poles. Making the observer poles faster ensures that the estimated and actual state variables converge quickly, but making them too fast can cause noise from the output to propagate. As we incrementally made the observer poles faster, we observed some interesting behavior. When all four poles were around -9, we saw the cart's acceleration quickly alternate between +/- 0.5 m/sec^2, causing the cart to move right briefly before moving left and pendulum angular displacement to increase. We also noticed that the first pole had more impact on stability; setting p1 = -8 caused instability, while setting p4 = -8 (while keeping all other poles constant) moved the system to stability. However, only increasing p1 speed, while other poles remained slower, did not seem sufficient for stability. Setting p1 = -15, p2 = -11, and the other two poles around -7 made the acceleration oscillate between +/- 0.5 m/sec^2 and the cart position oscillate between +/- 1 m, while the pendulum slowly fell. Further experimentation led us to find that setting L = [-20, -25, -30, -35] ensured stability and that the conditions were met, as shown in figure 2.

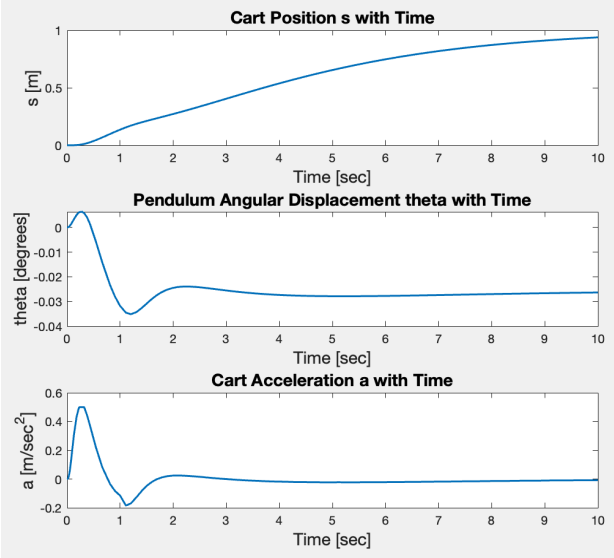

Figure 2 (left): Graphs of cart position, pendulum angular displacement, and cart acceleration with respect to time for adjusted observer with poles L = [-20 -25 -30 -35]. The position graph asymptotically approaches 1 m without overshoot and stabilizes at the value. Angular displacement oscillates before stabilizing at a near-zero value, suggesting a nearly upright pendulum position. Cart acceleration oscillates between minimum and maximum values before stabilizing at 0 as the cart-pendulum system reaches steady state and stabilizes. This behavior meets all of our system requirements.

To analyze the extent to which our controlled system could maintain closed loop stability, we incremented angular acceleration disturbance w by 0.01 rad/sec^2. We found that the largest allowable disturbance that maintains closed loop stability is α(t) = 0.631 rad/sec^2. For disturbances higher than this, the cart acceleration rapidly increased and remained at 0.5 m/sec^2; the cart's position overshot 1 m and reached 30 m in 10 seconds; the pendulum angular displacement increased to 60 degrees, indicating that the pendulum did not remain upright. These behaviors point to instability and hence, 0.631 rad/sec^2 is the maximum disturbance our designed system could withstand.

IV. Discussion and Conclusions

By employing integral action and an observer, we were able to design a state feedback control system that stabilized an inherently unstable system—an inverted pendulum on a moving cart. This control system allowed the system to meet the desired specifications of the cart: not overshooting the final position of 1 meter while maintaining the pendulum to remain upright in a 10 second interval. While we were able to meet these requirements, it is important to note that the system is highly sensitive. Our analysis of the maximum disturbance, which is only 0.131 rad/sec^2 greater than the current disturbance, shows that the system remains stable only for very specific conditions and can therefore easily become unstable.

Several key assumptions impacted our approach. We assumed that any external disturbance would cause the pendulum to fall freely, and that the system was free of hindrances such as air and friction (between the cart and ground, as well as between the pivot point and the pendulum arm). To simplify our calculations, we also employed small angle approximations where θ was near 0, limited cart motion to only the x direction, and assumed the pendulum mass was negligible.

While the inverted pendulum problem may seem inapplicable to real-world, practical systems, it does describe a number of systems that rely on feedback control to achieve stability or some desired movement. For example, segways or segway-like robots can be modeled as a pendulum on a moving platform, achieving a desired upright stability while moving forward and backward. Sensors such as gyroscopes and accelerometers are frequently used to measure angular displacement in these vehicles. More broadly, many ground and aerospace vehicles like cars, drones, and rockets use feedback to incrementally adjust trajectory.

Appendix A
Simulink Model

I. Top-level system

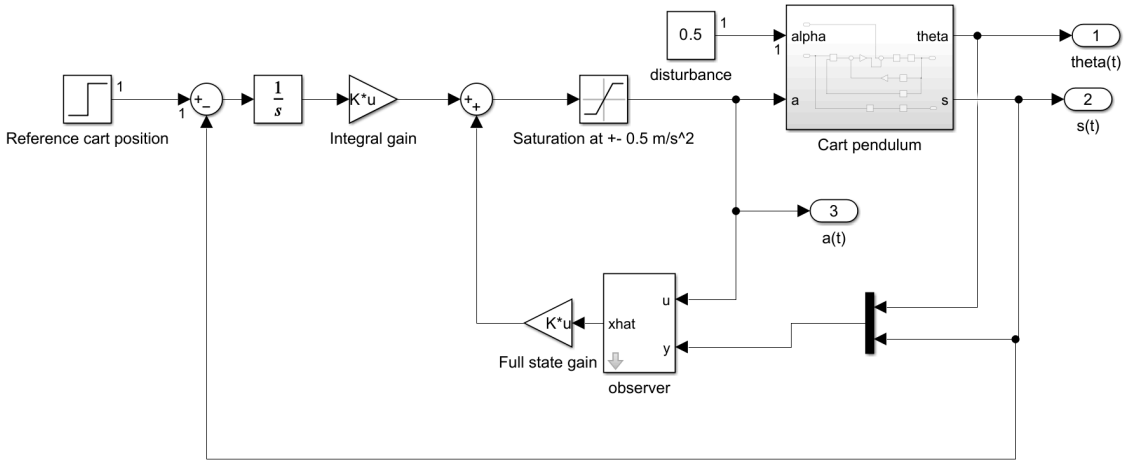

Figure 1A: Top-level system model in Simulink with observer and cart-pendulum plant

II. Cart-pendulum system

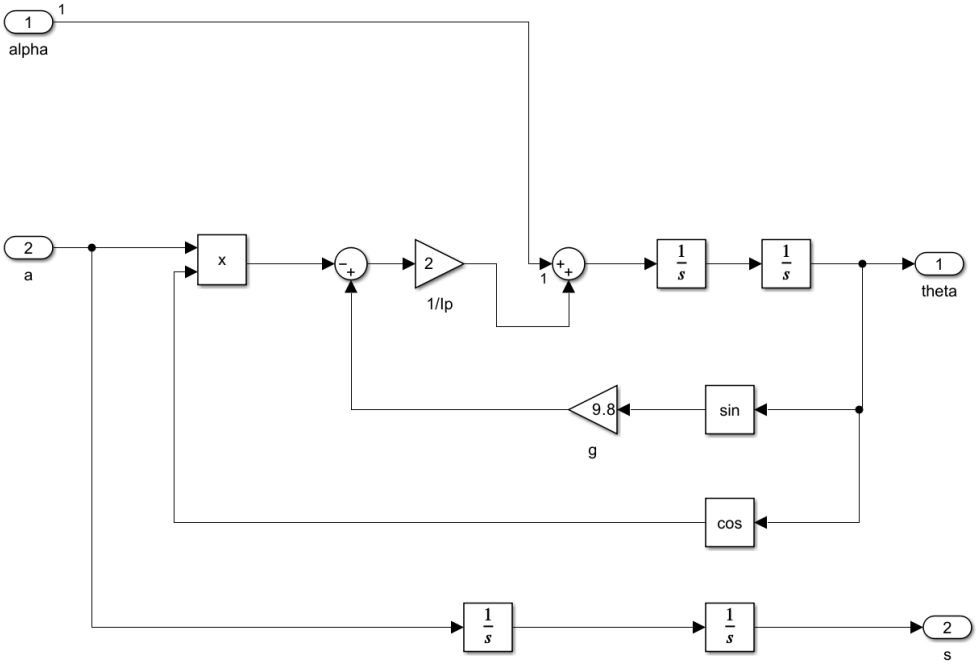

Figure 2A: Cart-pendulum plant system model in Simulink

III. Observer

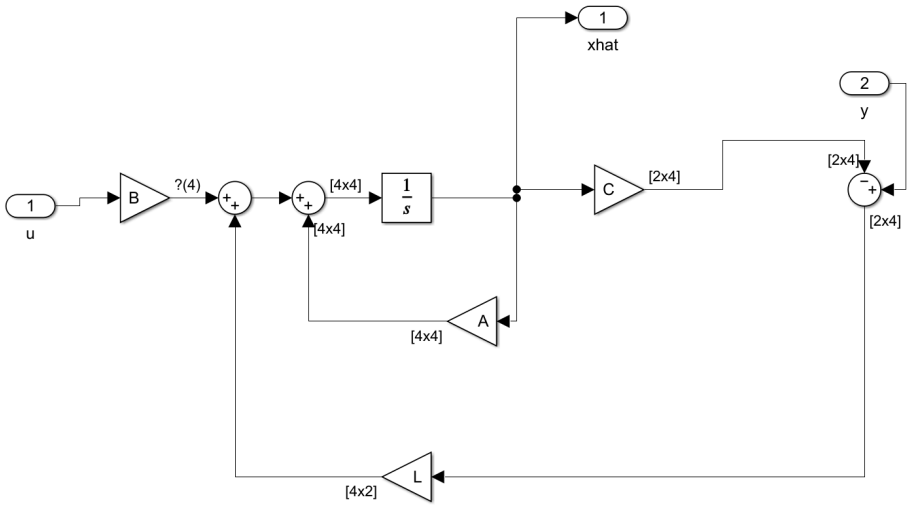

Figure 3A: Observer subsystem in Simulink

Appendix B

Matlab Code

```
%% #2 in Part I %%

g = 9.8;
L = 0.5;

% Defining matrices in state space
A = [0 1 0 0;
    g/L 0 0 0;
     0 0 0 1;
      0 0 0 0;];
B = [0; -1/L; 0; 1;];
C = [1 0 0 0;
     0 0 1 0;];
D = [0; 0;];

% Finding stability, controllability, and observability
sys = ss(A, B, C, D);

stability = isstable(sys);

Mc = ctrb(A, B);
Mc_rank = rank(Mc);

Mo = obsv(sys);
Mo_rank = rank(Mo);


```

```
%% #3 in Part I %%

zeta = 0.9999;
wn = 4.6/(10*zeta);
p1 = -zeta*wn + 1i*wn*sqrt(1-zeta^2);
p2 = -zeta*wn - 1i*wn*sqrt(1-zeta^2);
p3 = -4.6;
p4 = -4.7;
p5 = -4.8;
p_desired = [p1, p2, p3, p4, p5];

% Finding the augmented system matrix
Aa = [0 0 0 -1 0;
      0 0 1 0 0;
      0 g/L 0 0 0;
      0 0 0 0 1;
      0 0 0 0 0;];

Ba = [0; 0; -1/L; 0; 1;];

% Finding K
K_with_Ki = place(Aa, Ba, p_desired);
Ki = -1*K_with_Ki(1);
K = K_with_Ki(2:5);

% Designing observer
p_obs_des = [-6, -7, -8, -9];

L_trans = place(A', C', p_obs_des);
L = L_trans';


% Extracting variables from simulation
simout = sim('e102finalProject_simulink','ReturnWorkspaceOutputs', 'On');
theta = simout.yout{1}.Values.Data;
position = simout.yout{2}.Values.Data;
accel = simout.yout{3}.Values.Data;
time = simout.tout;

subplot(3,1,1)
plot(time, position,'LineWidth',1.5)
xlim([0 10])
title('Cart Position s with Time', 'FontSize', 14)
xlabel('Time [sec]','FontSize',14)
ylabel('s [m]', 'FontSize',14)

subplot(3,1,2)
plot(time, theta, 'LineWidth',1.5)
xlim([0 10])
title('Pendulum Angular Displacement theta with Time', 'FontSize', 14)
xlabel('Time [sec]', 'FontSize',14)
ylabel('theta [degrees]', 'FontSize',14)

subplot(3,1,3)
plot(time, accel, 'LineWidth',1.5)
xlim([0 10])
title('Cart Acceleration a with Time','FontSize',14)
xlabel('Time [sec]', 'FontSize',14)
ylabel('a [m/sec^2]', 'FontSize',14)
```