

**School of Engineering  
University of California, Merced**

---

**ME 141 - Control Engineering**

**Experiment No. Three**

# **Balance Control of an Inverted Pendulum**

# Objectives

1. Obtain a linear state-space representation of an open-loop system.
2. Design a state-feedback controller using LQR.

## Background

### (a) Linear State-Space Model

The linear Single Inverted Pendulum (SIP) model is shown below:

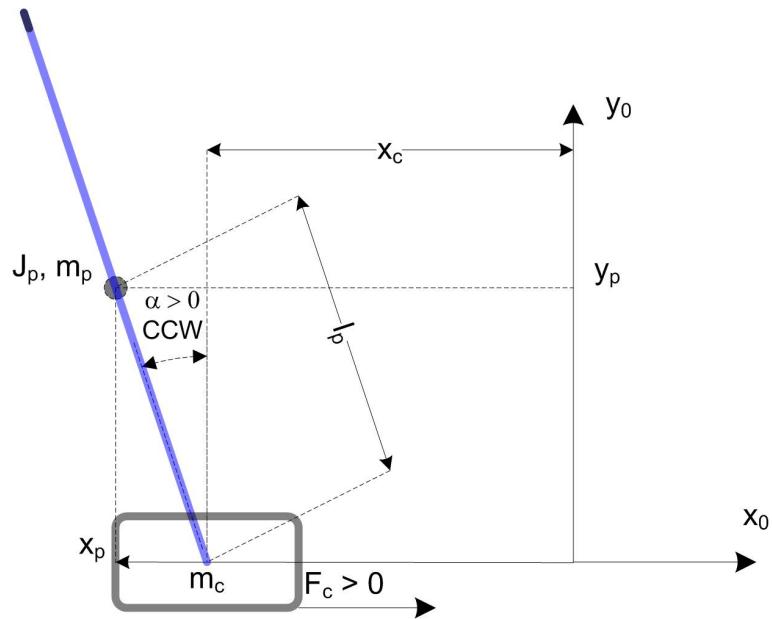


Figure 1: Linear Inverted Pendulum Schematic

The pendulum pivot is on the IP02 cart, and is measured using the *Pendulum* encoder. The center of mass of the pendulum is at length,  $l_p$ , and the moment of inertia about the center of mass is  $J_p$ . The pendulum angle,  $\alpha$ , is zero when it is perfectly balanced in an inverted position and increases positively when rotated counter-clockwise (CCW). The positive direction of linear displacement of the cart,  $x_c$ , is to the right when facing the cart. The position of the pendulum center of gravity is denoted as the  $(x_p, y_p)$  coordinate. The equations of motion (EOM) of the pendulum gantry to account for the orientation of

the inverted pendulum yield the following equations for the acceleration of the cart and pendulum:

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

and

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

where  $J_T = J_{eq}J_p + M_p J_p + J_{eq}M_p l_p^2$  and  $J_{eq} = M_c + \frac{\nu_g K_g^2 J_m}{r_{mp}^2}$ . For the linear inverted pendulum system, the state is defined as:

$$x^T = [x_c \quad \alpha \quad \dot{x}_c \quad \dot{\alpha}] \quad (3)$$

Where we can define  $\dot{x}_1 = x_3$  and  $\dot{x}_2 = x_4$ . Substituting state  $x$  into the equations of motion where  $x_c = x_1$ ,  $\alpha = x_2$ ,  $\dot{x}_c = x_3$ ,  $\dot{\alpha} = x_4$  gives

$$\dot{x}_3 = \frac{1}{J_T}(-(J_p + M_p l_p^2)B_{eq}x_3 - M_p l_p B_p x_4 + M_p^2 l_p^2 g x_2 + (J_p + M_p l_p^2)u) \quad (4)$$

and

$$\ddot{x}_4 = \frac{1}{J_T}(-(M_p l_p B_{eq})x_3 - (J_{eq} + M_p)B_p x_4 + (J_{eq} + M_p)M_p l_p g x_2 + M_p l_p u) \quad (5)$$

The  $A$  and  $B$  matrices in the  $\dot{x}_c = Ax + Bu$  equation are therefore:

$$A = \frac{1}{J_T} \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & M_p^2 l_p^2 g & -(J_p + M_p l_p^2)B_{eq} & -M_p l_p B_p \\ 0 & (J_{eq} + M_p)M_p l_p g & -M_p l_p B_{eq} & -(J_{eq} + M_p)B_p \end{bmatrix} \quad (6)$$

and

$$B = \frac{1}{J_T} \begin{bmatrix} 0 \\ 0 \\ J_p + M_p l_p^2 \\ M_p l_p \end{bmatrix} \quad (7)$$

In the output equation, only the position of the cart and pendulum angle is being measured. Based on this, the  $C$  and  $D$  matrices in the output equation are

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \quad (8)$$

and

$$D = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad (9)$$

By substituting the system parameters below:

- **Cart Position Rise Time:**  $t_r \leq 1.5s$
- **Maximum Pendulum Angle Deflection:**  $|\alpha| \leq 1.0 \text{ deg}$
- **Maximum Control Effort (V):**  $|V_m| < 10 \text{ V}$

we can calculate the eigenvalues of matrix  $A$ , which yields open-loop poles located at

$$OL = -16.26, -4.56, 4.84, \text{ and } 0$$

## (b) Linear Quadratic Regular (LQR)

If  $(A, B)$  are controllable, then the LQR optimization method can be used to find a feedback control gain. Given the plant model we can find a control input  $u$  that minimizes the cost function

$$J = \int_0^\infty x(t)^T Q x(t) + u(t)^T R u(t) dt \quad (10)$$

where  $Q$  and  $R$  are the weighting matrices. The weighting matrices affect how LQR minimizes the function and are treated as tuning variables. Given the control law  $u = -Kx$ , the state-space becomes

$$\dot{x} = Ax + B(-Kx) = (A - BK)x \quad (11)$$

## (c) Feedback Control

The reference state of the feedback control loop that balances the linear pendulum is defined as

$$x_d = [c_{cd} \ 0 \ 0 \ 0]$$

where  $x_{cd}$  is the desired cart position. The controller is

$$u = K(x_d - x) \quad (12)$$

Note that when  $x_d = 0$ , then  $u = -Kx$ , which is the control used in the LQR algorithm.

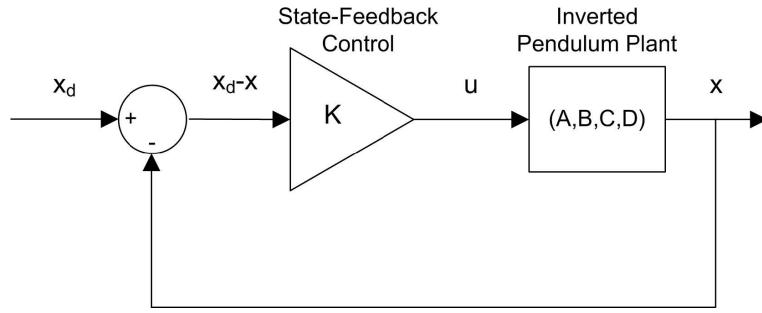


Figure 2: State-Feedback Control Loop

When running this on the physical system, the pendulum begins hanging in the downward position. We only want the balance control to be enabled when the pendulum is brought up around its upright vertical position. The controller is therefore

$$u = \begin{cases} K = (x_d - x) & |x_2| < \epsilon \\ 0 & \text{otherwise} \end{cases} \quad (13)$$

where  $\epsilon$  is the angle about which the controller should engage. For example if  $\epsilon = 10^\circ$ , then the control will begin when the pendulum is within  $\pm 10^\circ$  of its upright position, i.e., when  $|x_2| < 10^\circ$ .

## Procedure

For the performed experiments in the instructions below, we will need to use Matlab 2019b (other versions will not connect to the QUARC interface!).

### (a) Simulation

Change the workspace directory to the folder on the Desktop labeled "ME141 Lab #3" > "Linear Inverted Pendulum". We will be exploring the state-space model of the linear inverted pendulum.

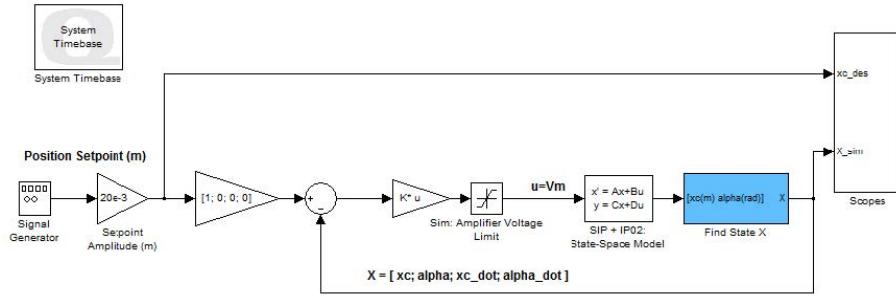


Figure 3: *s\_sip\_ipqr* Simulink diagram used to simulate the state-feedback control

1. Open the *setup\_ip02\_sip.m* script and locate the sections shown below and ensure they show exactly as follows before continuing:

```
% if IP02: Type of Cart Load: set to 'NO_LOAD', 'WEIGHT'  
IP02_LOAD_TYPE = 'NO_LOAD';  
%IP02_LOAD_TYPE = 'WEIGHT';
```

*The above assigns zero weight to the system's pendulum tip*

```
% ##### USER-DEFINED CONTROLLER DESIGN #####  
% Type of Controller: set it to 'LQR_AUTO', 'MANUAL'  
%CONTROLLER_TYPE = 'LQR_AUTO'; % LQR controller design: automatic mode  
CONTROLLER_TYPE = 'MANUAL'; % controller design: manual mode
```

*The above allows you to change the matrices in the parameters*

```
% ##### LQR CONTROL #####  
if strcmp ( CONTROLLER_TYPE, 'MANUAL' )  
Q = diag([1111]);  
R = 0.1;  
[K, S, EIG_CL] = lqr(A, B, Q, R);
```

The *Q* and *R* are initially set to the default values of:

$$Q = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (14)$$

and

$$R = 0.1$$

2. Run the script to generate the default gain  $K$ . Note: this will not give the desired response, but we will use it as a baseline gain  $K$  to compare with as we change matrices  $Q$  and  $R$ .
3. Run  $s_ip02_sip.m$  to simulate the closed-loop response with this gain.
  - If the scopes do not display any curves, begin the simulation from the green play button in any of the scope windows.
4. Alter the  $R$  parameter to observe its effect on the response of the system. Observe also how the values of gain  $K$  change.
5. For  $Q = \text{diag}[q_1, q_2, q_3, q_4]$ , vary each  $q_i$  independently and examine its effect on the gain and the closed-loop response. Vary each  $q_i$  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. Record your results so you can document and explain them in your lab report.
6. Find a  $Q$  and  $R$  that will satisfy the system specifications from *Background (a)*.
7. Save the data from the workspace labeled:
  - $\text{data\_}xc$ 
    - \* time =  $\text{data\_}xc(:,1)$
    - \* set-point position =  $\text{data\_}xc(:,2)$
    - \* measured position =  $\text{data\_}xc(:,3)$
  - $\text{data\_}\alpha$ 
    - \* time =  $\text{data\_}\alpha(:,1)$
    - \* measured angle =  $\text{data\_}\alpha(:,2)$
  - $\text{data\_}vm$ 
    - \* time =  $\text{data\_}vm(:,1)$
    - \* input voltage =  $\text{data\_}vm(:,2)$

## (b) Control Implementation

Open Simulink file named *q\_sip\_lqr* to view the block diagram as seen below:

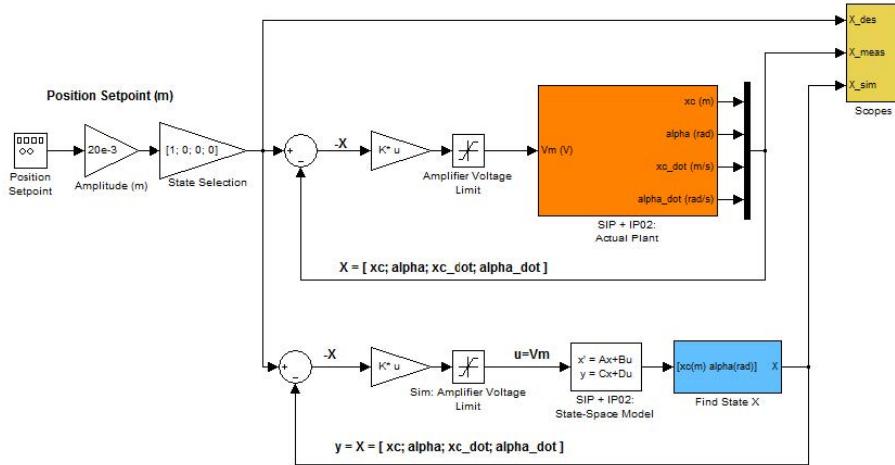


Figure 4: *q\_sip\_lqr* Simulink diagram used to implement the state-feedback control

The *Amplitude (m)* gain block is used to change the desired cart position. The state-feedback gain  $K$  is read from the Matlab workspace. The *SIP + IP02: Actual Plant* block interfaces with the IP02 cart motor and sensors of the system. The *SIP + IP02: State-Space Model* and *Find State X* blocks are used to simulate the response of the inverted pendulum using the linear state-space model.

1. Run the *setup\_ip02\_sip* script with the gain  $K$  you calculated in the previous steps
2. Turn on the Amplifier and set it to 1x and ensure the pendulum is centered
3. Open the *q\_sip\_lqr* Simulink file
4. At the top, open the "Monitor & Tune" drop-down menu and click on "Build for Monitoring"
5. Connect to the system and click *Run*
  - Once the system is running, the cart will not move until the pendulum is stood vertically
  - Manually rotate the pendulum until you feel the voltage take control of the pendulum

- Do NOT manually apply force to the cart or the pendulum while the control is operational
6. Once the system has balanced the pendulum, introduce the  $\pm 100$  mm cart position command by setting the *Amplitude (mm)* gain in the Simulink diagram to 100.
  7. Allow for a full cycle to display on the scope before stopping the controller
    - **The pendulum will fall after being released by the controller; keep clear of the system**
  8. Save the data from the workspace labeled:
    - *data\_xc*
      - \* time = *data\_xc(:,1)*
      - \* set-point position = *data\_xc(:,2)*
      - \* measured position = *data\_xc(:,3)*
    - *data\_alpha*
      - \* time = *data\_alpha(:,1)*
      - \* measured angle = *data\_alpha(:,2)*
    - *data\_vm*
      - \* time = *data\_vm(:,1)*
      - \* input voltage = *data\_vm(:,2)*
  9. Verify the pendulum response characteristics and control action satisfy the control parameters

## Action Items

1. Discuss whether the inverted pendulum system (without the control) is stable, marginally stable or unstable. Reference the open-loop poles in your discussion.
2. Designing a controller with the LQR technique is an iterative process. In order to gain some insight into how changing the different elements in  $Q$  and  $R$  will affect

the response, we only change the diagonal elements in  $Q$ , thus let:

$$Q = \begin{bmatrix} q_1 & 0 & 0 & 0 \\ 0 & q_2 & 0 & 0 \\ 0 & 0 & q_3 & 0 \\ 0 & 0 & 0 & q_4 \end{bmatrix} \quad (15)$$

Since we are dealing with a single-input system,  $R$  is a scalar value. Using the  $Q$  and  $R$  defined, expand the cost function (Equation 10).

3. For the feedback control  $u = -Kx$ , the LQR algorithm finds a gain  $K$  that minimizes the cost function  $J$ . Matrix  $Q$  sets the weight on the states and determines how  $u$  will minimize  $J$  (and hence how it generates gain  $K$ ). From *Action Item 2*, explain how increasing each of the diagonal elements,  $q_i$  affects the generated gain  $K = [k_1 \ k_2 \ k_3 \ k_4]$ .
4. State and explain how increasing  $R$  affects the generated gain  $K$ .
5. Provide a graph showing the simulated and control implementation responses.
6. Calculate the rise time and maximum pendulum angle of the simulated and control implementation responses.
7. Briefly discuss any sources of error, and how they affect your final results.

<i>FileName</i>	<i>Description</i>
setup_ip02_sip.m	Run this file only to set up the initial part of the experiment's inverted pendulum control parameter
config_ip02.m	Returns the configuration-based IP02 model specifications $Rm, Jm, Kt, Eff_m, Km, Kg, Eff_g, M, r_mp$ , and $B_{eq}$
config_sp.m	Returns the configuration-based pendulum specifications
d_ip02_sip_lqr.m	Determines the control gain $K$ using LQR.
SPG_ABCD_sqns_student.m	Creates the state space model of the linear inverted pendulum system.
s_sip_lqr.mdl	Simulink file that simulates the closed-loop pendulum gantry state-feedback control step response
q_sip_lqr.mdl	Simulink to implement the closed-loop IP02 lead speed controller using QU

Table 1: Matlab files needed for the Inverted Pendulum Control Experiment