School of Engineering
University of California, Merced

ME 141 - Control Engineering

Experiment No. Two

Pendulum Gantry

Objectives

1. Develop a mathematical model of a gantry pendulum using Lagrangian mechanics.
2. Obtain a linear state-space representation of an open-loop system .
3. Learn to apply fundamentals of state-space modeling and state-feedback.
4. Learn about pole-placement.

Background

(a) Modeling the System

The linear Single Pendulum Gantry (spg) model is shown below:

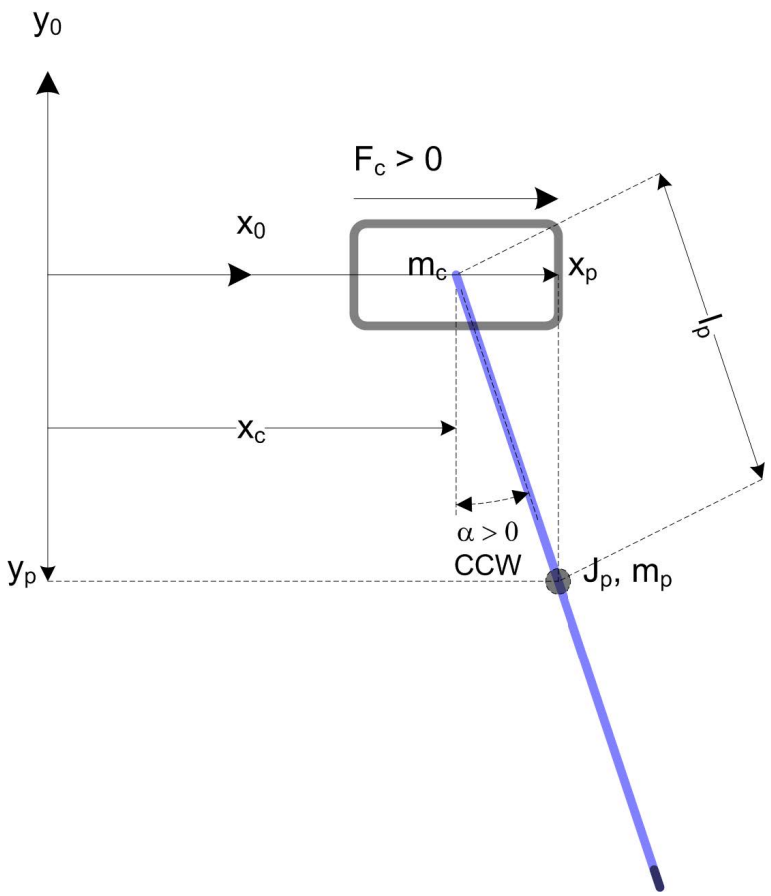

Figure 1: Desired closed-loop pole locations

The pendulum pivot is on the IP02 cart, and is measured using the $Pendulum$ encoder. The pendulum has mass, M_p, the centre of mass of the pendulum is at length, l_p, and the moment of inertia about the centre of mass is J_p. The pendulum angle, α, is zero when it is suspended perfectly vertically 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 centre of gravity is denoted as the $(x_p; y_p)$ coordinate. The nonlinear equations of motion for the cart and for the pendulum, respectively, are:

$$(J_{eq} + M_p)\ddot{x}_c + M_p l_p cos(\alpha)\ddot{\alpha} - M_p l_p sin(\alpha)\dot{\alpha}^2 = F_c - B_{eq}\dot{x}_c \tag{1}$$

and

$$M_p l_p cos(\alpha)\ddot{x}_c + (J_p + M_p l_p^2)\ddot{\alpha} + M_p l_p g sin(\alpha) = -B_p\dot{\alpha} \tag{2}$$

where F_c is the linear force applied to the cart, M_p is the mass of pendulum, l_p is the length of center of mass of pendulum from pivot, J_{eq} is the effective mass of the cart, J_p is the moment of inertia of the pendulum about the center of mass, g is the acceleration due to gravity, and B_{eq} is the equivalent viscous damping coefficient. This is our control input to the system. These equations match the standard form for a single body's equations of motion:

$$J\ddot{x} + b\dot{x} + g(x) = F \tag{3}$$

These equations of motion can be linearized to give the following linearized equations of motion:

$$(J_{eq} + M_p)\ddot{x}_c + M_p l_p\ddot{\alpha} = F_c - B_{eq}\dot{x}_c \tag{4}$$

$$M_p l_p\ddot{x}_c + (J_p + M_p l_p^2)\ddot{\alpha} + M_p l_p g\alpha = -B_p\dot{\alpha} \tag{5}$$

Thus, we can linearize as a state-space model:

$$\dot{x} = Ax + Bu \tag{6}$$

and

$$y = Cx + Du \tag{7}$$

where x is the state, u is the control input, and A, B, C, and D are the state-space matrices. The state x is a 4x1 vector given by the cart position x_c, pendulum angle α, cart speed $\dot{x}_c$, and pendulum angular velocity $\dot{\alpha}$ and can be written as the vector:

$$x = \begin{bmatrix} x_c \\ \alpha \\ \dot{x}_c \\ \dot{\alpha} \end{bmatrix} \tag{8}$$

The output y of the system is partial, and consists of the measurements of cart position and the pendulum angle:

$$y = \begin{bmatrix} x_c \\ \alpha \end{bmatrix} \tag{9}$$

Therefore, the A matrix is 4x4, B is 4x1, C is a 4x2, and D is 2x1. Based on this partial output, the matrices C and D are:

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \tag{10}$$

and

$$D = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{11}$$

In this lab, we will design and implement a controller of the form:

$$u = K(x_d - x) \tag{12}$$

where K is a vector of gains for each variable in the state vector, $x_d = [x_{cd}000]^T$ is the desired state vector, and x_{cd} is the desired cart position. We can pick the gains K to give desired poles for the closed-loop system.

(b) Gantry Control

For controlling the gantry, we will set the requirement for the response of the tip of the suspended pendulum to satisfy the following conditions:

- Percent Overshoot: $PO \leq 5\%$
- Percent Undershoot: $PU \leq 10\%$
- Settling Time (2%): $t_s \leq 2.2s$
- Steady-state Error: $e_{ss} = 0$
- Maximum Control Effort (V): $|V_m| < 10V$.

The linear pendulum gantry system has four poles where p_1 and p_2 are the complex conjugate *dominant* poles and are chosen to satisfy the setting time and overshoot specifications. Let the conjugate poles be:

$$p_1 = -\sigma + j\omega_d \tag{13}$$

and

$$p_2 = -\sigma - j\omega_d \tag{14}$$

where $\sigma = \zeta\omega_n$, and $\omega_d = \omega_n\sqrt{1-\zeta^2}$ is the *damped* natural frequency. The natural frequency, ω_n, and damping ratio, ζ, required to meet the specifications can be found using:

$$PO = \frac{100(y_{max} - R_0)}{R_0} \tag{15}$$

The remaining closed-loop poles, p_3 and p_4, are placed along the real-axis to the left of the dominant poles as seen in figure 2. Since the real part of these poles are more negative, their associated transients will decay faster, leading the system response to be dominated by the poles p_1 and p_2.

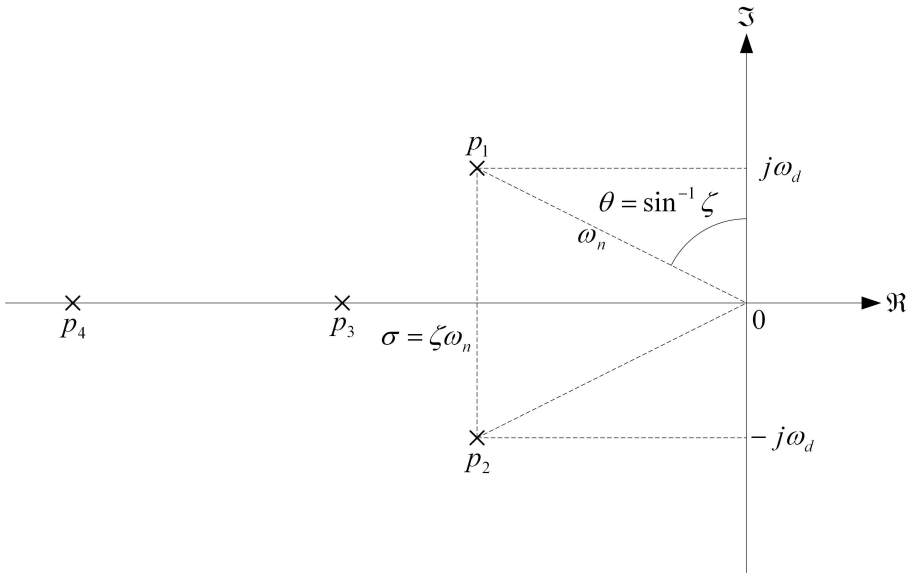

Figure 2: Desired closed-loop pole locations

The feedback control loop that controls the pendulum gantry is shown below. The reference state is defined as

$$x_d = [x_{cd}\ 0\ 0\ 0] \tag{16}$$

where x_cd is the desired cart position. The controller is $u = K(x_d - x)$

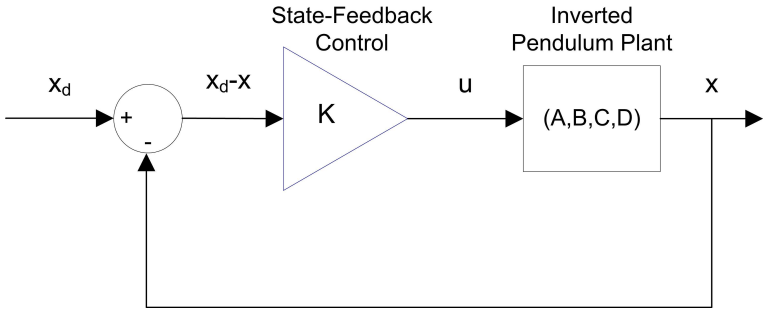

Figure 3: Desired closed-loop pole locations

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) Model Analysis

Change the workspace directory to the folder on the Desktop labeled "ME141 Lab #2" $>$ "Gantry Pendulum". We will be exploring the state-space model of the linear gantry pendulum.

1. Open and run the $setup\_ip02\_spg$ script
2. Open the $SPG\_ABCD\_eqns\_student.m$ script and enter the state-space matrices that you calculate under "Action Items"
3. Run the $setup\_ip02\_spg$ script with the CONTROLLER_TYPE set to 'MANUAL'. This will call the function $SPG\_ABCD\_eqns\_student$ and output the state-space

matrices that you entered. You can confirm that you entered these correctly by running the $setup\_ip02\_spg$ script with the CONTROLLER_TYPE set to 'PP_AUTO' and comparing the elements of the state-space matrices.

4. Identify the open-loop poles of the system

(b) Simulation

Open Simulink file named s_spg_pp to view the block diagram as seen below:

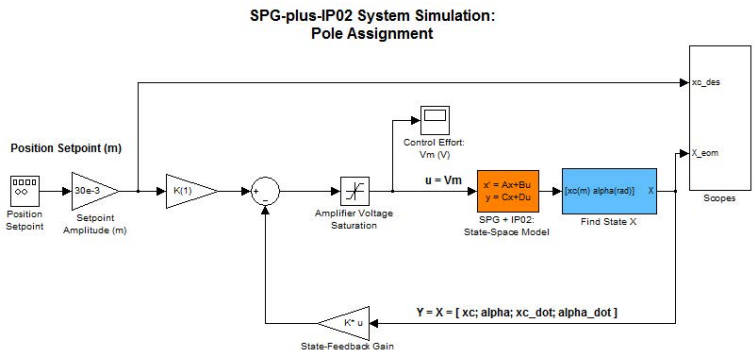

Figure 4: Simulink diagram for Pendulum Gantry Controller Simulation

1. Run the $setup\_ip02\_spg$ script and enter the calculated gain K
2. Run the $s\_spg\_pp.mdl$ Simulink file; save the data from the workspace labeled:
   - $data\_pend\_pos$
     - time = $data\_pend\_pos(:, 1)$
     - target pendulum tip position = $data\_pend\_pos(:, 2)$
     - measured pendulum tip position = $data\_pend\_pos(:, 3)$
   - $data\_vm$
     - time = $data\_vm(:, 1)$
     - control input = $data\_vm(:, 2)$
3. Verify the pendulum response characteristics and control action satisfy the control parameters

(c) Full-State Feedback

In full-state feedback, the ”full state” is available to the control. This means that the all the state variables, i.e. the cart position and speed and the pendulum angle and angular acceleration, are measured and known. So the controller can calculate a control signal based on all information in the state.
Open the Simulink file named q_spg_pp to view the block diagram as seen below:

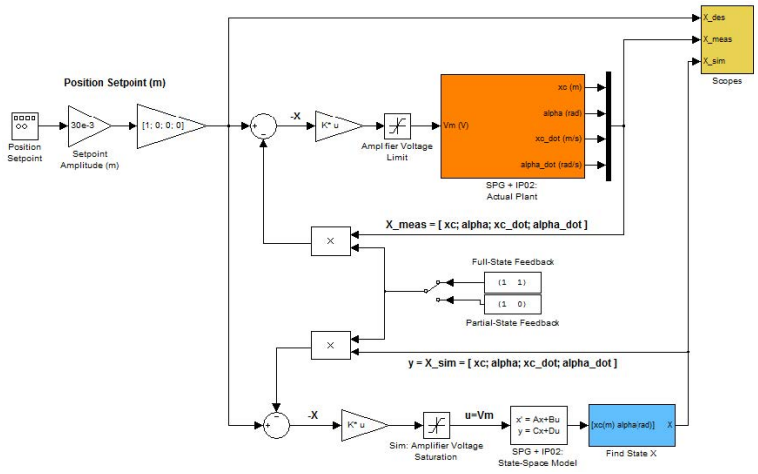

Figure 5: Simulink diagram for Pendulum Gantry Controller

1. Run the $setup\_ip02\_spg$ script then enter the calculated gain K
2. Turn on the physical amplifier unit and ensure the switch is set to 1x
3. Assign the following parameters to the $Signal\ Generator$:
   - Signal Type = $square$
   - Amplitude = 1
   - Frequency = $0.1Hz$
4. Open the $Amplitude\ (m)$ gain block and set the value to 0.03 to generate a step size of 30 millimeters
5. At the top, open the ”$Monitor\ \&\ Tune$” drop-down menu and click on ”$Build\ for\ Monitoring$”

6. Ensure the pendulum is hanging down and motionless
7. Run the file and allow for the scopes to display a full cycle before stopping
   - $Pend\,Tip\,Pos\,(mm)$ scope traces
     * Yellow = Set point
     * Purple = Measured Position
     * Cyan = Simulated Position
8. Export the data from $data\_pos$ and $data\_vm$

(d) Partial-State Feedback

In partial-state feedback, only part of the state is known to the controller. In this case, the cart position and speed are known, but the pendulum angle and angular velocity are unknown. This changes the effectiveness of the controller.
You will be using the same q_spg_pp Simulink file for part (c)

1. Change the *Signal Generator* step reference block to have the following values:
   - Signal type = $square$
   - Amplitude = 1
   - Frequency = $0.1Hz$
2. Set the $Amplitude\,(m)$ gain block to 0.03 to generate a step of $30\,mm$
3. Set the *Manual Switch* to the *Partial-State Feedback* (downward) position
4. At the top, open the "$Monitor\,\&\,Tune$" drop-down menu and click on "$Build\,for\,Monitoring$"
   - Verify that the amplifier is turned on with the gain set to 1x
   - Ensure the pendulum is hanging down and motionless
5. Once the file has compiled, click on $Connect$ then $Start$; "$Stop$" when you see a full cycle on the scopes

Action Items

1. Fit the two linear equations of motion (equations 4 and 5) into the matrix form in terms of the state variables x_c, $\dot{x}_c$, α and $\dot{\alpha}$ and the system constants. Your result should be in the form:

$$\begin{bmatrix} d_{11} & d_{12} \\ d_{21} & d_{22} \end{bmatrix} \begin{bmatrix} \ddot{x}_c \\ \ddot{\alpha} \end{bmatrix} + \begin{bmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{bmatrix} \begin{bmatrix} \dot{x}_c \\ \dot{\alpha} \end{bmatrix} + \begin{bmatrix} g_{11} & g_{12} \\ g_{21} & g_{22} \end{bmatrix} \begin{bmatrix} x_c \\ \alpha \end{bmatrix} = \begin{bmatrix} F_1 \\ F_2 \end{bmatrix} \tag{17}$$

2. Solve your result to get independent equations for the cart acceleration $\ddot{x}_c$ and the pendulum angular acceleration $\ddot{\alpha}$. Hint: You can achieve this by pre-multiplying by the inverse of matrix d, whose inverse is given by:

$$d^{-1} = \begin{bmatrix} d_{11} & d_{12} \\ d_{21} & d_{22} \end{bmatrix}^{-1} = \frac{1}{\det(d)} \begin{bmatrix} d_{22} & -d_{12} \\ -d_{21} & d_{11} \end{bmatrix} \tag{18}$$

   You will end up with two equations of the form: $\ddot{x}_c = f_1(x_c, \alpha, \dot{x}_c, \dot{\alpha})$ and $\ddot{\alpha} = f_2(x_c, \alpha, \dot{x}_c, \dot{\alpha})$. The equations will also contain the system constants from the equations of motion such as M_p, l_p, etc. Find the matrices A and B that allow you to write the system in linear state-space form as given in equation 6.

3. Find the open-loop poles of the system using Matlab. The open-loop poles are the eigenvalues of the state matrix, A. The MATLAB function eig can help.

4. Derive the characteristic equation of A using the open-loop poles.

5. Identify the location of the two dominant poles, p_1 and p_2 to satisfy the time-domain specifications for percent overshoot and settling time. We will place the other poles at $p_3 = -20$ and $p_4 = -40$. Note that here, the 2% settling time is used, which is related to the damping coefficient and natural frequency by the equation:

$$t_s = -\frac{\ln 0.02}{\zeta \omega_n} \approx \frac{4}{\zeta \omega_n}$$

   Use the MATLAB $place$ command (or $acker$), to calculate the gains K that will place the closed-loop poles at the desired locations. These should be:

$$K = \begin{bmatrix} 160.5347 & -210.625 & 88.0092 & 23.4776 \end{bmatrix}$$

   You can confirm that you entered these correctly by running the $setup\_ip02\_spg$ script with the CONTROLLER_TYPE set to 'PP_AUTO' and comparing the ele-

ments of the state-space matrices. Write down the matrices A and B that you obtain from running $setup\_ip02\_spg$ with the CONTROLLER_TYPE set to 'MANUAL' which uses the state-space matrices you derived.

6. Provide a graph showing the measured pendulum tip response using state-feedback.

7. Calculate the steady-state error, the percent overshoot, and the peak time for the pendulum tip.

8. Discuss the difference between the *Full-State Feedback* and the *Partial-State Feedback* responses.

9. Briefly discuss any sources of error, and how they affect your final results.

$\boldsymbol{FileName}$	$\boldsymbol{Description}$
setup_ip02_spg.m	Run this file only to set up the experiment's gantry control para
config_ip02.m	Returns the configuration-based IP02 model specificatio Rm, Jm, Kt, Eff_m, Km, Kg, Eff_g, M, r_mp, and E
config_sp.m	Returns the configuration-based pendulum specification
d_ip02_spg_pp.m	Determines the control gain K.
SPG_ABCD_sqns_student.m	Creates the state space model of the linear pendulum gantry s MATLAB variables are defined as: Mp = mass of pendulum, M_p lp = length of center of mass of pendulum from pivot, $l_$ Jeq = effective mass of the cart, J_{eq} Jp = moment of inertia of the pendulum about the center of m g = acceleration due to gravity, g Beq = equivalent viscous damping coefficient, B_{eq}
s_spg_pp.mdl	Simulink file that simulates the closed-loop pendulum gantry state-feedback
q_spg_pp.mdl	Simulink to implement the closed-loop IP02 lead speed controller u
q_spg_mdl.mdl	Simulink file that simulates and compares the state-space model to the line
q_spg_pp_l.mdl	Simulink file to implement the integrator in the closed-loop pendulum tip

Table 1: Matlab files needed for the Gantry Control Experiment