QUANSER
INNOVATE. EDUCATE.

STUDENT WORKBOOK

Linear Pendulum Gantry Experiment for MATLAB®/Simulink® Users

Standardized for ABET* Evaluation Criteria

Developed by:
Jacob Apkarian, Ph.D., Quanser
Hervé Lacheray, M.A.SC., Quanser
Peter Martin, M.A.SC., Quanser

Quanser educational solutions are powered by:
MATLAB
SIMULINK

Course material complies with:
ABET

CAPTIVATE. MOTIVATE. GRADUATE.

*ABET Inc., is the recognized accreditor for college and university programs in applied science, computing, engineering, and technology; and has provided leadership and quality assurance in higher education for over 75 years.

Quanser Inc.
119 Spy Court
Markham, Ontario
L3R 5H6
Canada
info@quanser.com
Phone: 1-905-940-3575
Fax: 1-905-940-3576

Printed in Markham, Ontario.

For more information on the solutions Quanser Inc. offers, please visit the web site at:
http://www.quanser.com

CONTENTS

1 INTRODUCTION

The objective of this laboratory is to develop a feedback system to track the linear cart to a commanded position while minimizing the swing of the suspended pendulum. A full-state-feedback controller using pole placement is designed to control the position of the pendulum tip within a specified set of requirements.

Topics Covered

- Developing of a mathematical model of the pendulum in the gantry configuration using Lagrangian mechanics.
- Linearizing nonlinear equations of motion
- Obtaining the linear state-space representation of the open-loop system
- Designing a state-feedback control system using pole placement
- Simulating the closed-loop system to ensure that the specifications are met
- Implementing the controller on the IP02 plant and evaluating its performance

Prerequisites

In order to successfully carry out this laboratory, the user should be familiar with the following:

- The required software and hardware outlined in Section 4.
- State-space modeling fundamentals.
- Some knowledge of state-feedback.
- Basics of QUARC®.
- Laboratory described in the QUARC Integration [2] in order to be familiar using QUARC®with the IP02.

2 MODELING

2.1 Background

2.1.1 Model Convention

The linear Single Pendulum Gantry (SPG) model is shown in Figure 2.1. The pendulum pivot is on the IP02 cart, and is measured using the *Pendulum* encoder. 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.

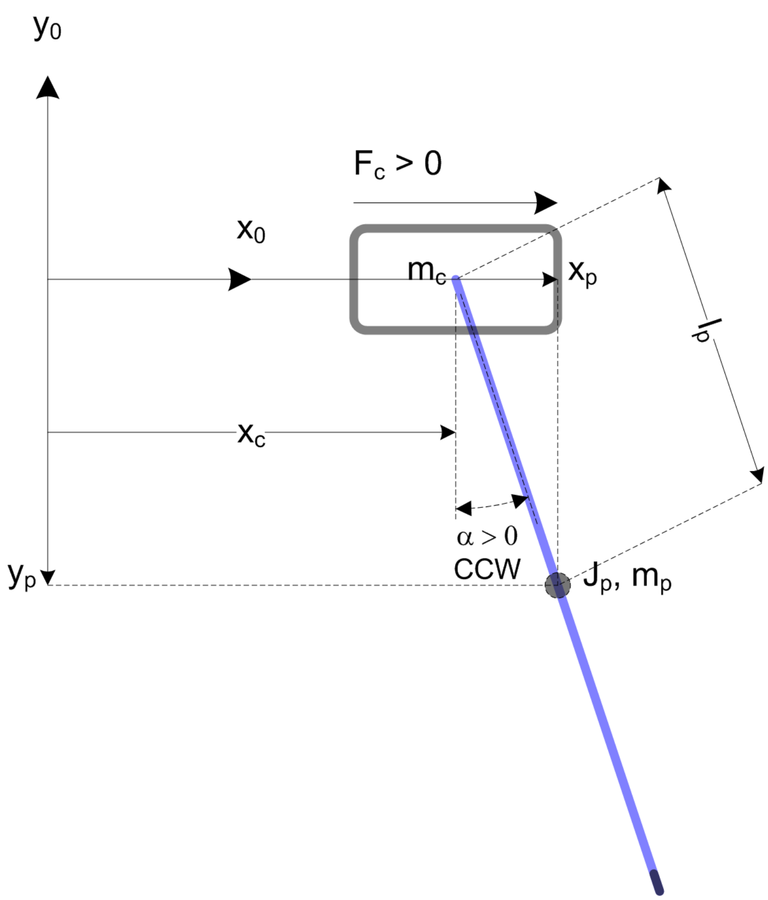


Figure 2.1: Linear Gantry Pendulum schematic

2.1.2 Nonlinear Equations of Motion

Instead of using classical mechanics, the Lagrange method is used to find the equations of motion of the system. This systematic method is often used for more complicated systems such as robotic manipulators with multiple joints.

More specifically, the dynamics equations that describe the motion of the linear cart and pendulum with respect to the motor voltage will be obtained using the Euler-Lagrange equation:

$$\frac{\partial^2 L}{\partial t \partial \dot{q}_i} - \frac{\partial L}{\partial q_i} = Q_i$$

The variables q_i are called *generalized coordinates*. For this system let

$$q(t)^\top = [x_c(t)\ \alpha(t)] \tag{2.1}$$

where $x_c(t)$ is the cart position and $\alpha(t)$ is the pendulum angle. The corresponding velocities are

$$\dot{q}(t)^\top = \left[\frac{\partial x_c(t)}{\partial t}\ \frac{\partial \alpha(t)}{\partial t}\right]$$

Note: The dot convention for the time derivative will be used throughout this document, i.e., $\dot{x_c} = \frac{dx_c}{dt}$. The time variable t will also be dropped for x_c and α, i.e., $x_c = x_c(t)$ and $\alpha = \alpha(t)$.

With the generalized coordinates defined, the Euler-Lagrange equations for the linear pendulum gantry are:

$$\begin{aligned} \frac{\partial^2 L}{\partial t \partial \dot{x_c}} - \frac{\partial L}{\partial x_c} &= Q_{x_c} \\ \frac{\partial^2 L}{\partial t \partial \dot{\alpha}} - \frac{\partial L}{\partial \alpha} &= Q_\alpha \end{aligned}$$

The Lagrangian of the system, L, is described:

$$L = T - V$$

where T is the total kinetic energy of the system and V is the total potential energy of the system. Therefore, the Lagrangian represents the difference between the kinetic and potential energy of the system.

According to the reference frame definition shown in Figure 2.1, the Cartesian coordinates of the centre of gravity of the pendulum are defined by:

$$x_p = x_c + l_p \sin(\alpha)$$

and

$$y_p = -l_p \cos(\alpha)$$

Since the linear motion of the cart is fixed vertically, the total potential energy of the system can be expressed as the gravitational potential energy of the pendulum:

$$V_T = -M_p g l_p \cos(\alpha) \tag{2.2}$$

The total kinetic energy of the system is the sum of the translational and rotational kinetic energies arising from the cart and pendulum. The translational kinetic energy of the cart, T_{ct}, can be expressed as:

$$T_{ct} = \frac{1}{2} M \dot{x_c}^2$$

and the rotational energy due to the DC motor, $T_c r$, is:

$$T_{cr} = \frac{1}{2} \frac{\eta_g J_m K_g^2 \dot{x_c}^2}{r_{mp}^2}$$

where the corresponding IP02 parameters are defined in the IP02 User Manual [3]. If we combine the kinetic energy equations together, the resultant total kinetic energy of the cart is:

$$T_c = \frac{1}{2} J_{eq} \dot{x_c}^2 \tag{2.3}$$

where

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

The pendulum translational kinetic energy can be expressed as a function of the linear velocity of the centre of gravity:

$$T_{pt} = \frac{1}{2} M_p \sqrt{\dot{x_p}^2 + \dot{y_p}^2} \tag{2.4}$$

where the components of the linear velocity of the pendulum centre of gravity are defined by:

$$\dot{x_p} = \dot{x_c} + l_p \cos(\alpha)\dot{\alpha}$$

and

$$\dot{y_p} = l_p \sin(\alpha)\dot{\alpha}.$$

The rotational kinetic energy of the pendulum is defined as:

$$T_{pr} = \frac{1}{2} J_p \dot{\alpha}^2 \tag{2.5}$$

where the parameters of the pendulum are listed in the SIP and SPG User Manual [5].

The total kinetic energy of the system can be found by combining the kinetic energy of the cart in Equation 2.3, with the pendulum kinetic energy in Equation 2.4 and Equation 2.5:

$$T_T = \frac{1}{2}(J_{eq} + M_p)\dot{x_c}^2 + M_p l_p \cos(\alpha)\dot{\alpha}\dot{x_c} + \frac{1}{2}(J_p + M_p l_p^2)\dot{\alpha}^2 \tag{2.6}$$

The generalized forces, Q_{x_c} and Q_α, are used to describe the non-conservative forces (e.g., friction) applied to a system with respect to the generalized coordinates. The generalized force acting on the linear cart is:

$$Q_{x_c} = F_c - B_{eq}\dot{x_c} \tag{2.7}$$

and acting on the pendulum is

$$Q_\alpha = -B_p \dot{\alpha}. \tag{2.8}$$

Note: The nonlinear Coulomb friction applied to the linear cart, and the force on the linear cart due to the pendulum's action have been neglected in the dynamic model.

By substituting the total kinetic and potential energy of the system shown in Equation 2.2 and Equation 2.6, and the generalized forces into the Euler-Lagrange formulation, the nonlinear Equations of Motion (EOM) are:

$$(J_{eq} + M_p)\ddot{x_c} + M_p l_p \cos(\alpha)\ddot{\alpha} - M_p l_p \sin(\alpha)\dot{\alpha}^2 = F_c - B_{eq}\dot{x_c} \tag{2.9}$$

and

$$M_p l_p \cos(\alpha)\ddot{x_c} + (J_p + M_p l_p^2)\ddot{\alpha} + M_p l_p g \sin(\alpha) = -B_p \dot{\alpha}. \tag{2.10}$$

The linear force applied to the cart, F_c, is generated by the servo motor and described by the equation:

$$F_c = \left(\frac{\eta_g K_g K_t}{R_m r_{mp}}\right)\left(-\frac{K_g K_m \dot{x_c}}{r_{mp}} + \eta_m V_m\right) \tag{2.11}$$

where the servo motor parameters are defined in the IP02 User Manual [3].

Both the equations match the typical form of an EOM for a single body:

$$J\ddot{x} + b\dot{x} + g(x) = F_1$$

where x is a linear position, J is the moment of inertia, b is the damping, $g(x)$ is the gravitational function, and F_1 is the applied force (scalar value).

For a generalized coordinate vector q, this can be generalized into the matrix form

$$D(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) = F \tag{2.12}$$

where D is the inertial matrix, C is the damping matrix, $g(q)$ is the gravitational vector, and F is the applied force vector.

The nonlinear equations of motion given in 2.9 and 2.10 can be placed into this matrix format.

2.1.3 Linearization

Here is an example of how to linearize a two-variable nonlinear function, $f(z)$. Variable z is defined

$$z^\top = [z_1 \; z_2]$$

and $f(z)$ is to be linearized about the operating point

$$z_0{}^\top = [a \; b]$$

The linearized function is

$$f_{lin} = f(z_0) + \left(\frac{\partial f(z)}{\partial z_1}\right)\Bigg|_{z=z_0}(z_1 - a) + \left(\frac{\partial f(z)}{\partial z_2}\right)\Bigg|_{z=z_0}(z_2 - b)$$

2.1.4 Linear State-Space Model

The linear state-space equations are

$$\dot{x} = Ax + Bu \tag{2.13}$$

and

$$y = Cx + Du \tag{2.14}$$

where x is the state, u is the control input, A, B, C, and D are state-space matrices. For the linear pendulum gantry system, the state and output are defined

$$x^\top = [x_c \; \alpha \; \dot{x_c} \; \dot{\alpha}] \tag{2.15}$$

and

$$y^\top = [x_1 \; x_2]. \tag{2.16}$$

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

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \tag{2.17}$$

and

$$D = \begin{bmatrix} 0 \\ 0 \end{bmatrix}. \tag{2.18}$$

The velocities of the servo and pendulum can be computed in the digital controller by taking the derivative and filtering the result though a high-pass filter.

2.2 Pre-Lab Questions

1. Linearize the first nonlinear linear pendulum gantry equation, Equation 2.9. The initial conditions for all the variables are zero, i.e., $x_{c_0} = 0$, $\alpha_0 = 0$, $\dot{x}_{c0} = 0$, $\dot{\alpha}_0 = 0$.

2. When the linearization procedure outlined in Section 2.1.3 is applied to the second nonlinear equation of motion Equation 2.10, the process yields

$$(M_p l_p)\,\ddot{x}_c + \left(J_p + M_p l_p^2\right)\ddot{\alpha} + M_p l_p g \alpha = -B_p \dot{\alpha}. \tag{2.19}$$

   Fit the two linear equations of motion into the matrix form shown in Equation 2.12. Make sure the equation is in terms of x_c and α (and their derivatives).

3. Solve for the acceleration terms in the equations of motion. You can either solve this using the two linear equations or using the matrix form. If you're doing it in the matrix form, recall that the inverse of a 2x2 matrix is

$$A^{-1} = \begin{bmatrix} a & b \\ c & d \end{bmatrix}^{-1} = \frac{1}{\text{det}(A)} \begin{bmatrix} d & -b \\ -c & a \end{bmatrix}, \tag{2.20}$$

   where det$(A) = ad - bc$.

   In any case, you'll have two equations of the form: $\ddot{x}_c = g_1(x_c, \alpha, \dot{x}_c, \dot{\alpha})$ and $\ddot{\alpha} = g_2(x_c, \alpha, \dot{x}_c, \dot{\alpha})$. Make sure you collect the terms with respect to the x_c, α, $\dot{x}_c$, and $\dot{\alpha}$ variables.

4. Find the linear state-space of the linear gantry pendulum system.

2.3 In-Lab Exercises

The goal of this laboratory is to explore the state-space model of the linear gantry pendulum. You will conduct an experiment to validate the model by comparing the response of the model to the response of the actual system.

2.3.1 Model Analysis

1. Run the *setup_ip02_spg* script. The IP02 linear cart and pendulum model parameters are automatically loaded using the *config_ip02.m* and *config_sp.m* functions. It then calls the *SPG_ABCD_eqns_student.m* script to load the model in the Matlab workspace.
2. Open the *SPG_ABCD_eqns_student.m* script. The script should contain the following code:

   ```
   A = eye(4,4);
   B = [0;0;0;1];
   C = eye(2,4);
   D = zeros(2,1);

   %Actuator Dynamics
   A(3,3) = A(3,3) - B(3)*eta_g*Kg^2*eta_m*Kt*Km/r_mp^2/Rm;
   A(4,3) = A(4,3) - B(4)*eta_g*Kg^2*eta_m*Kt*Km/r_mp^2/Rm;
   B = eta_g*Kg*eta_m*Kt/r_mp/Rm*B;
   ```

   The representative C and D matrices have already been included. You need to enter the state-space matrices A and B that you found in Section 2.2. The actuator dynamics have been added to convert your state-space matrices to be in terms of voltage. Recall that the input of the state-space model you found in Section 2.2 is the force acting on the IP02 cart. However, *we do not control force directly - we control the servo input voltage*. The above code uses the voltage-torque relationship given in Equation 2.11 in Section 2.1.2 to transform torque to voltage.
3. Run the *SPG_ABCD_eqns_student.m* script to load the state-space matrices in the Matlab workspace. Show the numerical matrices that are displayed in the Matlab prompt.
4. Using MATLAB, find the open-loop poles of the system.

Before ending this lab... To do the pre-lab questions in Section 3.3, you need the A and B matrices (numerical representation) and the open-loop poles. Make sure you record these.

2.3.2 Model Validation

Experimental Setup

The *q_spg_mdl.mdl* Simulink diagram shown in Figure 2.2 is used to confirm that the actual system hardware matches the state-space model derived in Section 2.1.4. The QUARC blocks are used to interface with encoders and DC motor of the system. For more information about QUARC, see Reference [2]. This model outputs the pendulum link angles and can apply a voltage to the DC motor.

Note: Before you can conduct these experiments, you need to make sure that the lab files are configured according to your IP02 and pendulum setup. If they have not been configured already, then go to Section 4.2 to configure the lab files before you begin.

1. Make sure that you have completed the procedure in Section 2.3.1 and the state-space parameters are present in the MATLAB workspace.
2. Double-click on the *Signal Generator* block and ensure the following parameters are set:

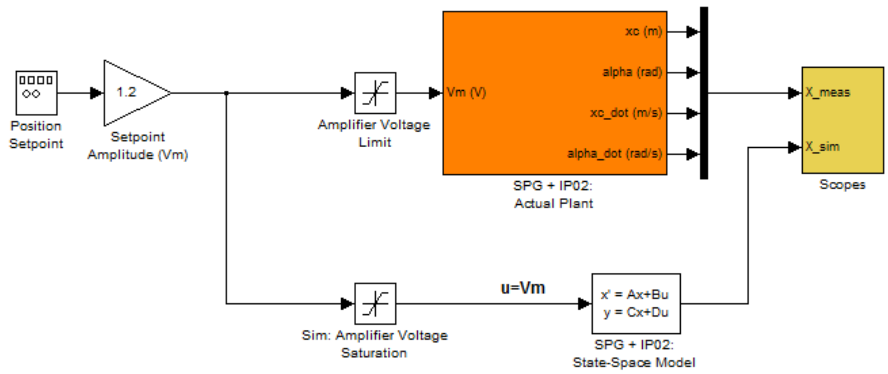


Figure 2.2: q_spg_mdl Simulink diagram used to confirm system model

- Wave form: square
- Amplitude: 1.0
- Frequency: 0.8
- Units: Hertz

3. Set the *Amplitude (V)* slider to 1.2 V.
4. Open the pendulum tip and cart position scopes, *Pend Tip Pos (mm)* and *xc (mm)*.
5. Click on QUARC | Build to compile the Simulink diagram.
6. Ensure the pendulum is in the hanging down position and is motionless.
7. Select QUARC | Start to run the controller. The IP02 cart should begin moving back and forth along the track, and the scopes should be as shown in figures Figure 2.4 and Figure 2.3. Recall that the yellow trace is the measured position and the purple trace is the simulated position.

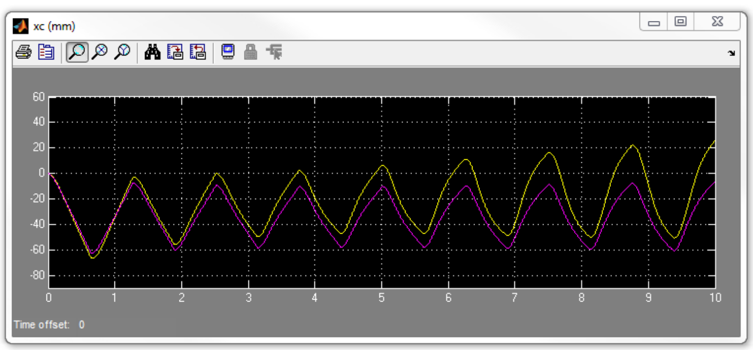


Figure 2.3: Simulated and measured cart position

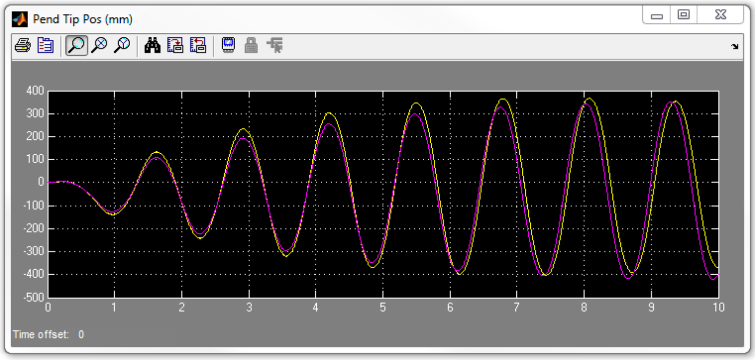


Figure 2.4: Simulated and measured pendulum position

8. Create a MATLAB figure that compares the simulated and experimental response of the pendulum and cart position, and provide two reasons why the model does not represent the IP02 and pendulum gantry with better accuracy. When the model is stopped, the *Pend Tip Pos (mm)* scope saves the last five seconds of response data to the MATLAB workspace in the *data_pend_pos* parameter. The cart position data is saved in the *data_xc* parameter. The variables have the following structure: *data_x(:,1)* is the time vector, *data_x(:,2)* is the measured position, and *data_x(:,3)* is the simulated position.

3 GANTRY CONTROL

3.1 Specifications

The response of the tip of the suspended pendulum should satisfy the following requirements:

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

Note: The previous specifications are given in response to a ±30 mm square wave cart position setpoint. PO and PU are defined to limit the relative endpoint position of the gantry.

3.2 Background

In the Pendulum Gantry Modeling experiment presented in Section 2, we found a linear state-space model that represents the linear pendulum gantry system. In Section 2.3.1 we found that the system has poles in the left-hand plane making it stable, but two complex-conjugate poles with a very low corresponding damping ratio. The linear pendulum gantry system is thus largely underdamped, and as a consequence requires a state-feedback controller in order to meet the settling time and overshoot design specifications presented in Section 3.1.

In Section 3.2.1, the notion of controllability is introduced. The procedure to transform matrices to their companion form is described in Section 3.2.2. Once in their companion form, it is easier to design a gain according to the pole-placement principles, which is discussed in Section 3.2.3. Lastly, Section 3.2.5 describes the state-feedback control used to control the gantry pendulum.

3.2.1 Controllability

If the control input u of a system can take each state variable, x_i where $i = 1 \ldots n$, from an initial state to a final state then the system is controllable, otherwise it is uncontrollable.

Rank Test The system is controllable if the rank of its controllability matrix

$$T = \left[B\ AB\ A^2B \ldots A^nB \right] \tag{3.1}$$

equals the number of states in the system,

$$\text{rank}(T) = n.$$

3.2.2 Companion Matrix

If (A, B) are controllable and B is $n \times 1$, then A is similar to a companion matrix. Let the characteristic equation of A be

$$s^n + a_n s^{n-1} + \ldots + a_1.$$

Then the companion matrices of A and B are

$$\tilde{A} = \begin{bmatrix} 0 & 1 & \cdots & 0 & 0 \\ 0 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & 1 \\ -a_1 & -a_2 & \cdots & -a_{n-1} & -a_n \end{bmatrix} \tag{3.2}$$

and

$$\tilde{B} = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} \tag{3.3}$$

Define

$$W = T\tilde{T}^{-1}$$

where T is the controllability matrix defined in Equation 3.1 and

$$\tilde{T} = [\tilde{B}\ \tilde{B}\tilde{A} \ldots \tilde{B}\tilde{A}^n].$$

Then

$$W^{-1}AW = \tilde{A}$$

and

$$W^{-1}B = \tilde{B}.$$

3.2.3 Pole Placement

If (A,B) are controllable, then pole placement can be used to design the controller. Given the control law $u = -Kx$, the state-space in Equation 2.13 becomes

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

We can generalize the procedure to design a gain K for a controllable (A,B) system as follows:

Step 1 Find the companion matrices $\tilde{A}$ and $\tilde{B}$. Compute $W = T\tilde{T}^{-1}$.

Step 2 Compute $\tilde{K}$ to assign the poles of $\tilde{A} - \tilde{B}\tilde{K}$ to the desired locations. Applying the control law $u = -Kx$ to the general system given in Equation 3.2,

$$\tilde{A} = \begin{bmatrix} 0 & 1 & \cdots & 0 & 0 \\ 0 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & 1 \\ -a_1 - k_1 & -a_2 - k_2 & \cdots & -a_{n-1} - k_{n-1} & -a_n - k_n \end{bmatrix} \tag{3.4}$$

Step 3 Find $K = \tilde{K}W^{-1}$ to get the feedback gain for the original system (A,B).

Remark: It is important to do the $\tilde{K} \rightarrow K$ conversion. Remember that (A,B) represents the actual system while the companion matrices $\tilde{A}$ and $\tilde{B}$ do not.

3.2.4 Desired Poles

The linear pendulum gantry system has four poles. As depicted in Figure 3.1, poles p_1 and p_2 are the complex conjugate *dominant* poles and are chosen to satisfy the settling time and overshoot specifications given in Section 3.1. Let the conjugate poles be

$$p_1 = -\sigma + j\omega_d \tag{3.5}$$

and

$$p_2 = -\sigma - j\omega_d \tag{3.6}$$

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 outlined in Section 3.1 can be found using:

$$PO = \frac{100(y_{max} - R_0)}{R_0}. \tag{3.7}$$

and

$$t_s = \frac{4}{\zeta\omega_n} \tag{3.8}$$

The remaining closed-loop poles, p_3 and p_4, are placed along the real-axis to the left of the dominant poles, as shown in Figure 3.1.

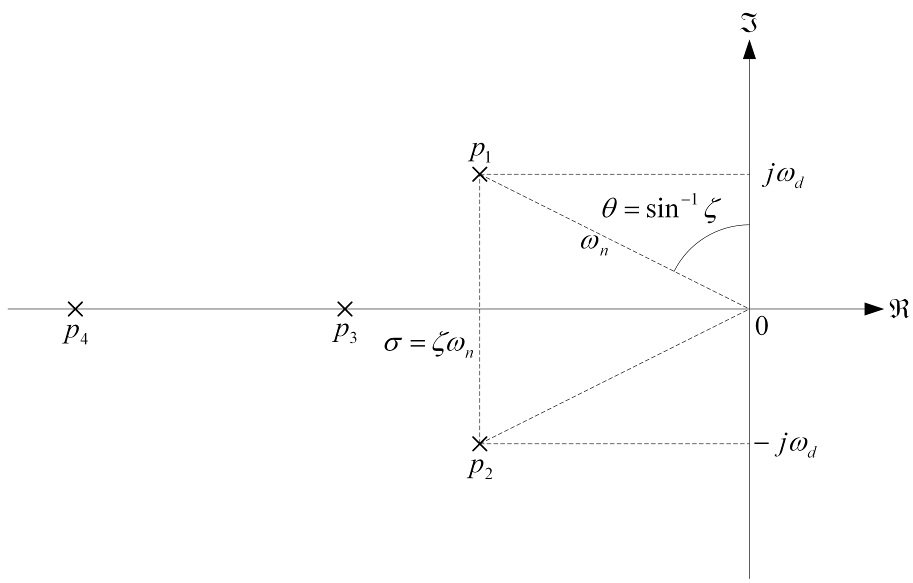


Figure 3.1: Desired closed-loop pole locations

3.2.5 Feedback Control

The feedback control loop that controls the pendulum gantry is illustrated in Figure 3.2. The reference state is defined

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

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

$$u = K(x_d - x). \tag{3.9}$$

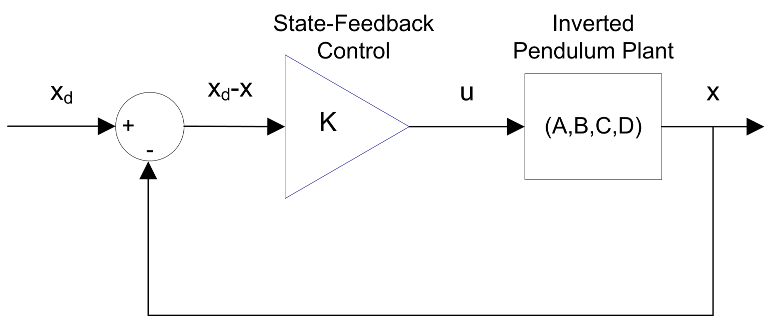


Figure 3.2: State-feedback control loop

3.3 Pre-Lab Questions

1. Using the open-loop poles, find the characteristic equation of A.

   Note: The poles are the roots of the system's characteristic equation.

2. Find the location of the two dominant poles, p_1 and p_2, based on the specifications given in Section 3.1. Place the other poles at $p_3 = -20$ and $p_4 = -40$.

3.4 In-Lab Exercises

3.4.1 Control Design

1. Using Matlab commands, determine if the system is controllable. Explain why.
2. The pole-placement procedure outlined in Section 3.2.3 can be performed automatically using a pre-defined *Compensator Design* Matlab command. Find gain K using a Matlab pole-placement command that will place the closed-loop poles at the desired locations found in Section 3.3.

3.4.2 Simulating the Gantry Control

Experiment Setup

The *s_spg_pp* Simulink diagram shown in Figure 3.3 is used to simulate the closed-loop response of the Linear Pendulum Gantry using the state-feedback control described in Section 3.2.5 with the control gain K found in Section 3.4.1.

The *Amplitude (m)* gain block is used to change the pendulum position. The state-feedback gain K is read from the Matlab workspace. The Simulink *State-Space* block reads the A, B, C, and D state-space matrices that are loaded in the Matlab workspace. The *Find State X* block contains high-pass filters to find the velocity of the rotary arm and pendulum.

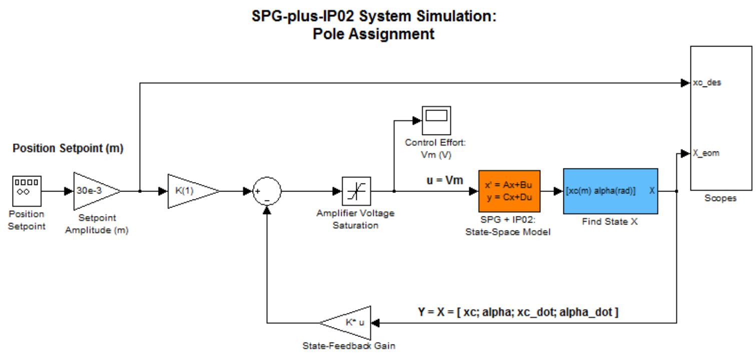


Figure 3.3: s_spg_pp Simulink diagram used to simulate the state-feedback control

IMPORTANT: Before you can conduct this experiment, you need to make sure that the lab files are configured according to your system setup. If they have not been configured already, go to Section 4.2 to configure the lab files first. **Make sure the model you found in Section 2.3 is entered in SPG_ABCD_eqns_student.m.**

1. Run *setup_ip02_spg.m*. Ensure the gain K you found in Section 3.4.1 is loaded.
2. Run the *s_spg_pp.mdl*. The response in the scopes shown in Figure 3.4 and Figure 3.5 were generated using an arbitrary feedback control gain. Plot the simulated response of the pendulum tip, and motor input voltage obtained using your obtained gain K in a Matlab figure and attach it to your report.

   Note: When the simulation stops, the last 10 seconds of data is automatically saved in the Matlab workspace to the variables *data_pend_pos* and *data_vm*. The time is stored in the *data_pend_pos(:,1)* vector, the desired and measured pendulum tip positions are saved in the *data_pend_pos(;,2)* and *data_pend_pos(;,3)* arrays, and the control input is in the *data_vm(:,2)* structure.

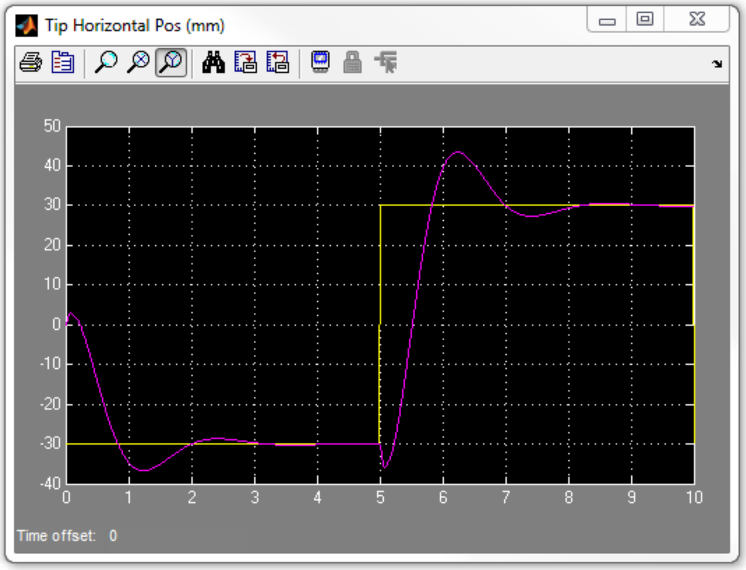


Figure 3.4: Simulated pendulum tip position

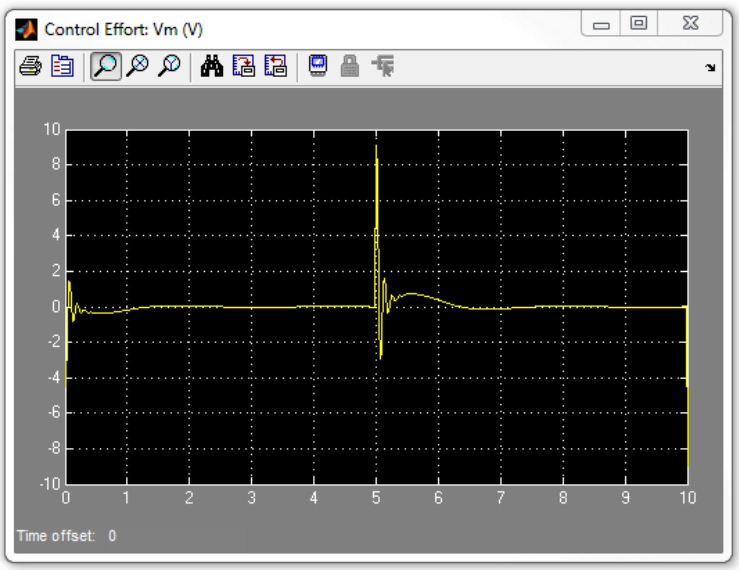


Figure 3.5: Simulated control action

3. Measure the pendulum response characteristics and control action. Are the specifications given in Section 3.1 satisfied?
4. Close the Simulink diagram when you are done.

3.4.3 Implementing the Gantry Controller

In this section, ths state-feedback control that was designed and simulated in the previous sections is run on the actual IP02 Linear Pendulum Gantry device.

Experiment Setup

The *q_spg_pp* Simulink diagram shown in Figure 3.6 is used to run the state-feedback control on the Quanser Linear

Pendulum Gantry system. The *SPG + IP02: Actual Plant* subsystem contains QUARC blocks that interface with the DC motor and sensors of the system. The feedback developed in Section 3.4.1 is implemented using a Simulink *Gain* block.

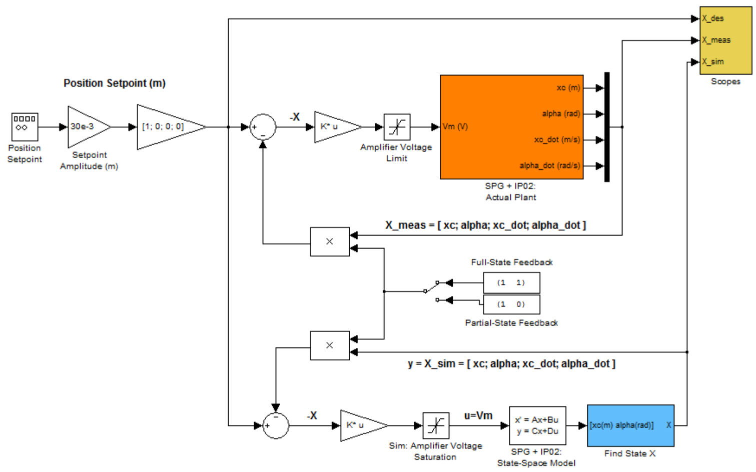


Figure 3.6: q_spg_pp Simulink diagram is used to run the pendulum gantry controller

IMPORTANT: Before you can conduct this experiment, you need to make sure that the lab files are configured according to your system setup. If they have not been configured already, then go to Section 4.2 to configure the lab files first.

1. Run *setup_ip02_spg.m*. Ensure the gain K you found in Section 3.4.1 is loaded.
2. Open the q_spg_pp Simulink diagram.
3. Turn ON the power amplifier.
4. To generate a step reference, ensure the *Signal Generator* is set to the following:
   - Signal type = *square*
   - Amplitude = 1
   - Frequency = 0.1 Hz
5. In the Simulink diagram, set the *Amplitude (m)* gain block to 0.03 to generate a step with an amplitude of 30 millimeters (i.e., square wave goes between ± 0.03 m which results in a step amplitude of 0.06 m).
6. Go to QUARC | Build to build the controller.
7. Ensure the pendulum is in the hanging down position and is motionless.
8. Select QUARC | Start to begin running the controller. The scopes should display responses similar to figures 3.7 and 3.8. Note that in the *Pend Tip Pos (mm)* scope, the yellow trace is the setpoint, the purple trace is the measured position, and the cyan trace is the simulated position.

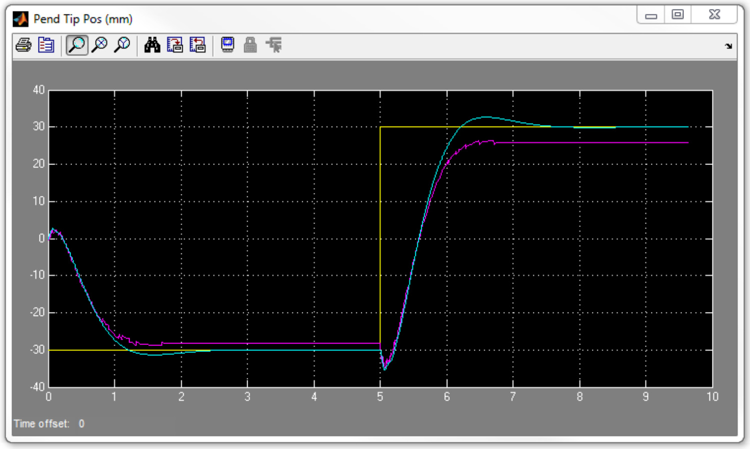


Figure 3.7: Simulated and measured cart position

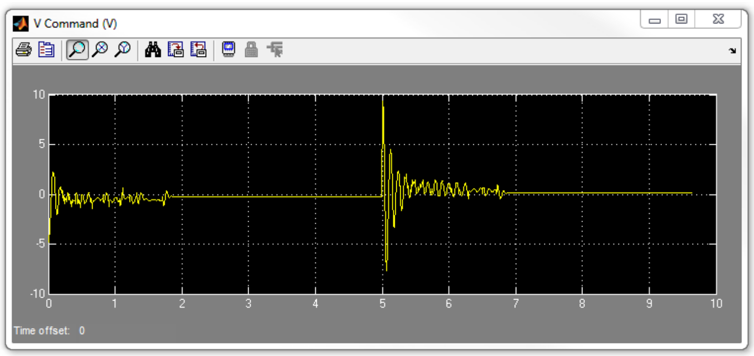


Figure 3.8: Simulated and measured pendulum position

9. When a suitable response is obtained, click on the Stop button in the Simulink diagram toolbar (or select QUARC | Stop from the menu) to stop running the code.
10. Generate a MATLAB figure showing the measured pendulum tip position response and its input voltage. As in the *s_spg_pp* Simulink diagram, when the controller is stopped each scope automatically saves their response to a variable in the MATLAB workspace. Thus, the *Pend Tip Pos (mm)* scope saves its response to the *data_pos* variable and the *V Command (V)* scope saves its data to the *data_vm* variable.
11. Measure the steady-state error, the percent overshoot, and the settling time of the pendulum tip. Does the response satisfy the specifications given in Section 3.1?
12. How could you alter the controller to compensate for the steady-state error?

3.4.4 Partial-State Feedback

1. Run *setup_ip02_spg.m*. Ensure the gain K you found in Section 3.4.1 is loaded.
2. Open the q_spg_pp Simulink diagram.
3. Turn ON the power amplifier.
4. To generate a step reference, ensure the *Signal Generator* is set to the following:
   - Signal type = *square*
   - Amplitude = 1

- Frequency = 0.1 Hz

5. In the Simulink diagram, set the *Amplitude (m)* gain block to 0.03 to generate a step with an amplitude of 30 millimeters (i.e., square wave goes between ± 0.03 m which results in a step amplitude of 0.06 m).
6. Go to QUARC | Build to build the controller.
7. Ensure the pendulum is in the hanging down position and is motionless.
8. Set the *Manual Switch* to the *Partial-State Feedback* (downward) position. Select QUARC | Start to begin running the controller.
9. Stop the controller once you have obtained a representative response.
10. Create a MATLAB figure representing the pendulum tip position response, as well as the input voltage.
11. Examine the difference between the partial-state feedback (PSF) response and the full-state feedback (FSF) response. Explain why PSF control behaves this way by looking at the *q_spg_pp* Simulink diagram.
12. Click the Stop button on the Simulink diagram toolbar (or select QUARC | Stop from the menu) to stop the experiment.
13. Turn off the power to the amplifier if no more experiments will be performed in this session.

4 SYSTEM REQUIREMENTS

Before you begin this laboratory make sure:

- QUARC®is installed on your PC, as described in Reference [4].
- You have a QUARC compatible data-aquisition (DAQ) card installed in your PC. For a listing of compliant DAQ cards, see Reference [1].
- IP02 and amplifier are connected to your DAQ board as described Reference [5].

4.1 Overview of Files

File Name	**Description**
Linear Pendulum Gantry Workbook (Student).pdf	This laboratory guide contains a modeling and gantry control experiment demonstrating state-space modeling and feedback control of the Quanser linear pendulum gantry. The in-lab exercises are explained using the QUARC software.
setup_ip02_spg.m	The main Matlab®script that sets the IP02 pendulum gantry control parameters. **Run this file only to setup the laboratory.**
config_ip02.m	Returns the configuration-based IP02 model specifications *Rm, Jm, Kt, Eff_m, Km, Kg, Eff_g, M, r_mp,* and *Beq*.
config_sp.m	Returns the configuration-based pendulum specifications.
d_ip02_spg_pp.m	Determines the control gain *K*.
SPG_ABCD_eqns_student.m	Creates the student state space model of the linear pendulum gantry system.
s_spg_pp.mdl	Simulink file that simulates the closed-loop pendulum gantry state-feedback control step response.
q_spg_pp.mdl	Simulink file that implements the closed-loop IP02 lead speed controller using QUARC.
q_spg_mdl.mdl	Simulink file that simulates and compares the state-space model to the linear pendulum gantry.

Table 4.1: Files supplied with the Linear Pendulum Gantry Laboratory.

4.2 Configuring the IP02 and the Lab Files

Before beginning the lab exercises the IP02 device, the *q_spg_pp* Simulink diagram and the *setup_ip02_spg.m* script must be configured.

Follow these steps to get the system ready for this lab:

1. Set up the IP02 with the long pendulum and additional weight as described in [5].
2. Load the Matlab®software.
3. Browse through the Current Directory window in Matlab®and find the folder that contains the linear pendulum gantry files, e.g. q_spg_pp.mdl.

4. Double-click on the q_spg_pp.mdl file to open the Simulink diagram shown in Figure Figure 3.6.

5. **Configure DAQ:** Double-click on the HIL Initialize block in the Simulink diagram and ensure it is configured for the DAQ device that is installed in your system. For instance, the block shown in Figure 3.6 is setup for the Quanser Q2-USB hardware-in-the-loop board. See the QUARC Installation Guide [4] for more information on configuring the HIL Initialize block.

6. Repeat the HIL Initialize configuration for *q_spg_mdl.mdl*.

7. Go to the *Current Directory* window and double-click on the setup_ip02_spg.m file to open the setup script.

8. **Configure setup script:** The beginning of the setup script is shown below. Ensure the script is setup to match the configuration of your actual IP02 device. For example, the script given below is setup for an IP02 plant with the additional weight and long pendulum, and it is actuated using the Quanser VoltPAQ device with a gain of 1. See the IP02 User Manual [3] for more information on IP02 plant options and corresponding accessories. Finally, make sure MODELING_TYPE is set to 'MANUAL'.

```
% if IP02: Type of Cart Load: set to 'NO_WEIGHT', 'WEIGHT'
% IP02_WEIGHT_TYPE = 'NO_WEIGHT';
IP02_WEIGHT_TYPE = 'WEIGHT';
% Type of single pendulum: set to 'LONG_24IN', 'MEDIUM_12IN'
PEND_TYPE = 'LONG_24IN';
% PEND_TYPE = 'MEDIUM_12IN';
% Turn on or off the safety watchdog on the cart position: set it to 1 , or 0
X_LIM_ENABLE = 1;       % safety watchdog turned ON
%X_LIM_ENABLE = 0;      % safety watchdog turned OFF
% Safety Limits on the cart displacement (m)
X_MAX = 0.2;            % cart displacement maximum safety position (m)
X_MIN = - X_MAX;        % cart displacement minimum safety position (m)
% Turn on or off the safety watchdog on the pendulum angle: set it to 1 , or 0
%ALPHA_LIM_ENABLE = 1;       % safety watchdog turned ON
ALPHA_LIM_ENABLE = 0;      % safety watchdog turned OFF
% Safety Limits on the pendulum angle (deg)
global ALPHA_MAX ALPHA_MIN
ALPHA_MAX = 25;            % pendulum angle maximum safety position (deg)
ALPHA_MIN = - ALPHA_MAX;   % pendulum angle minimum safety position (deg)
% Amplifier Gain used: set VoltPAQ to 1
K_AMP = 1;
% Amplifier Type, e.g. 'VoltPAQ', or 'Q3'
AMP_TYPE = 'VoltPAQ';
% AMP_TYPE = 'Q3';
% Digital-to-Analog Maximum Voltage (V); for MultiQ cards set to 10
VMAX_DAC = 10;

% ##### USER-DEFINED CONTROLLER DESIGN #####
% Type of Controller: set it to 'PP_AUTO', 'MANUAL'
%CONTROLLER_TYPE = 'PP_AUTO';    % pole placement design: automatic mode
CONTROLLER_TYPE = 'MANUAL';    % controller design: manual mode
% Pole Placement Design Specifications
PO = 5;     % spec #1: maximum percent overshoot
ts = 2.2;   % spec #2: maximum 2% settling time
```

9. Run the script by selecting the Debug | Run item from the menu bar or clicking on the *Run* button in the tool bar.

5 LAB REPORT

When you prepare your lab report, you can follow the outline given in Section 5.1 to build the *content* of your report. Also, in Section 5.2 you can find some basic tips for the *format* of your report.

5.1 Template for Content

I. PROCEDURE

I.1. Modeling Experiment

1. Briefly describe the main goal of this experiment and the procedure.
    - Briefly describe the experimental procedure (Section 2.3.1), *Model Analysis*
    - Briefly describe the experimental procedure in Step 2 in Section 2.3.1.
    - Briefly describe the experimental procedure in Step 3 in Section 2.3.1.
    - Briefly describe the experimental procedure (Section 2.3.2), *Model Validation*

I.2. Gantry Control Experiment

1. Briefly describe the main goal of this experiment and the procedure.
    - Briefly describe the experimental procedure (Section 3.4.1), *Control Design*
    - Briefly describe the experimental procedure in Step 1 in Section 3.4.1.
    - Briefly describe the experimental procedure in Step 2 in Section 3.4.1.
    - Briefly describe the experimental procedure (Section 3.4.3), *Control Implementation*
    - Briefly describe the experimental procedure in Step 10 in Section 3.4.3.
    - Briefly describe the experimental procedure in Step 12 in Section 3.4.3.
    - Briefly describe the experimental procedure (Section 3.4.3), *Partial-State Feedback*
    - Briefly describe the experimental procedure in Step 10 in Section 3.4.4.

II. RESULTS
Do not interpret or analyze the data in this section. Just provide the results.

II.1. Modeling Experiment

1. Numeric matrices from Step 3 in Section 2.3.1, *Model Analysis*
2. Simulation and experimental response comparison from Step 8 in Section 2.3.2, *Model Validation*

II.2. Gantry Control Experiment

1. Simulated pendulum tip, and motor voltage responses from Step 2 in Section 3.4.1, *Control Design*
2. Pendulum and motor voltage response from Step 10 in Section 3.4.3, *Control Implementation*
3. Pendulum and motor voltage response from Step 10 in Section 3.4.4, *Partial-State Feedback*

III. ANALYSIS
Provide details of your calculations (methods used) for analysis for each of the following:

III.1. Modeling Experiment

1. Step 4 in Section 2.3.1, *Model Analysis*.

III.2. Gantry Control Experiment

1. Step 1 in Section 3.4.1, *Control Design*.
2. Step 2 in Section 3.4.1, *Control Design*.
3. Step 3 in Section 3.4.1, *Control Design*.
4. Step 11 in Section 3.4.3, *Control Implementation*.
5. Step 11 in Section 3.4.4, *Partial-State Feedback*.

IV. CONCLUSIONS
Interpret your results to arrive at logical conclusions.

1. Steps 8 in Section 2.3.
2. Steps 3, and 11 in Section 3.4.

5.2 Tips for Report Format

PROFESSIONAL APPEARANCE

- Has cover page with all necessary details (title, course, student name(s), etc.)
- Each of the required sections is completed (Procedure, Results, Analysis and Conclusions).
- Typed.
- All grammar/spelling correct.
- Report layout is neat.
- Does not exceed specified maximum page limit, if any.
- Pages are numbered.
- Equations are consecutively numbered.
- Figures are numbered, axes have labels, each figure has a descriptive caption.
- Tables are numbered, they include labels, each table has a descriptive caption.
- Data are presented in a useful format (graphs, numerical, table, charts, diagrams).
- No hand drawn sketches/diagrams.
- References are cited using correct format.

REFERENCES

[1] Quanser Inc. *QUARC User Manual*.

[2] Quanser Inc. *IP02 QUARC Integration*, 2008.

[3] Quanser Inc. *IP02 User Manual*, 2009.

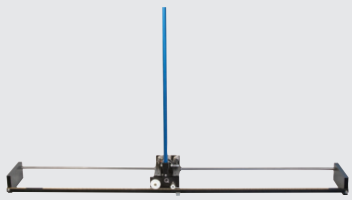
[4] Quanser Inc. *QUARC Installation Guide*, 2009.

[5] Quanser Inc. *SIP and SPG User Manual*, 2009.

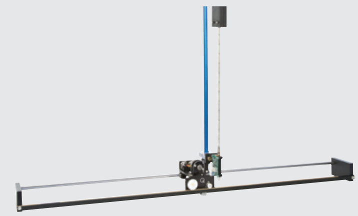
Nine linear motion plants for teaching fundamental and advanced controls concepts

IP02 Base Unit

Linear Pendulum



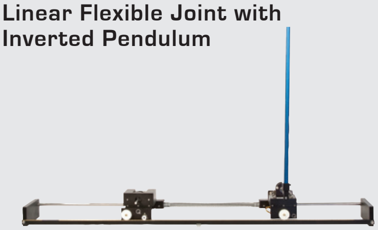
Linear Flexible Inverted Pendulum



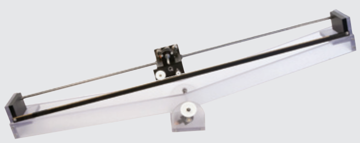
Double Inverted Pendulum

Linear Flexible Joint

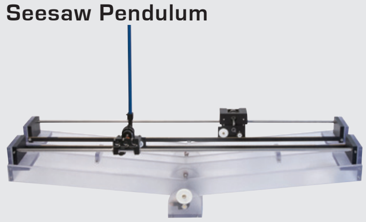
Linear Flexible Joint with Inverted Pendulum



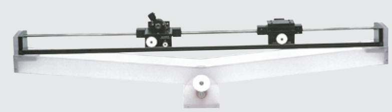
Seesaw



Seesaw Pendulum



Linear Flexible Joint on Seesaw



Quanser's linear collection allows you to create experiments of varying complexity – from basic to advanced. With nine plants to choose from, students can be exposed to a wide range of topics relating to mechanical and aerospace engineering. For more information please contact info@quanser.com

QUANSER
INNOVATE. EDUCATE.

INFO@QUANSER.COM   +1-905-940-3575   QUANSER.COM

Solutions for teaching and research. Made in Canada.