QUANSER
INNOVATE. EDUCATE.

LABORATORY GUIDE

Linear Flexible Joint with Inverted Pendulum Experiment for MATLAB®/Simulink® Users

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®

CAPTIVATE. MOTIVATE. GRADUATE.

CONTENTS

1 INTRODUCTION

This laboratory manual describes how to design a state-feedback control system that can balance an inverted pendulum mounted on the linear flexible joint cart while minimizing the spring deflection.

The plant has two main components: the Quanser IP02 linear motion plant and the Quanser LFJC-PEN-E module.

Topics Covered

- Obtain a state-space representation of the open-loop system.

- Design and tune an LQR-based state-feedback controller satisfying the closed-loop system's desired design specifications.

- Simulate the system and ensure it is stabilized using the designed state-feedback control.

- Implement the state-feedback controller on the LFJC-PEN system and evaluate its actual performance.

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

1. See the system requirements in Section 4 for the required hardware and software.

2. Modeling and state-space representation.

3. State-feedback design using Linear-Quadratic Regular (LQR) optimization.

4. Basics of Simulink®.

5. QUARC Integration lab detailed in Appendix A in the IP02 Lab Workbook [5].

2 BACKGROUND

2.1 Modeling

2.1.1 Model Convention

The single inverted pendulum and linear flexible joint cart model is shown in Figure 2.1. The LFJC-PEN module is coupled to the IP02 Linear Servo Base Unit through a linear spring with equivalent stiffness K_s. The positive sense of linear displacement is to the right when facing the cart. The positive sense of rotation of the pendulum is defined to be counter-clockwise (CCW), when facing the cart pinions. Finally, the zero angle, $\alpha = 0$, of the pendulum corresponds to inverted pendulum perfectly vertical pointing upwards.

The IP02 cart location is at linear position x_c, while the LFJC-PEN cart is located at x_{jc}. The IP02 cart assembly has mass, m_c, and is actuated by an applied force, F_c. The total mass of the flexible joint cart and pendulum assembly is m_{jc}. The IP02 and LFJC-PEN carts have equivalent damping terms, B_{eq}, and $B_{eq_{jc}}$, respectively. The mass of the pendulum is defined as, m_p, with damping, B_p, located at the pendulum centre of mass, (x_p, y_p). The distance from the pendulum pivot to the centre of mass of the pendulum is l_p.
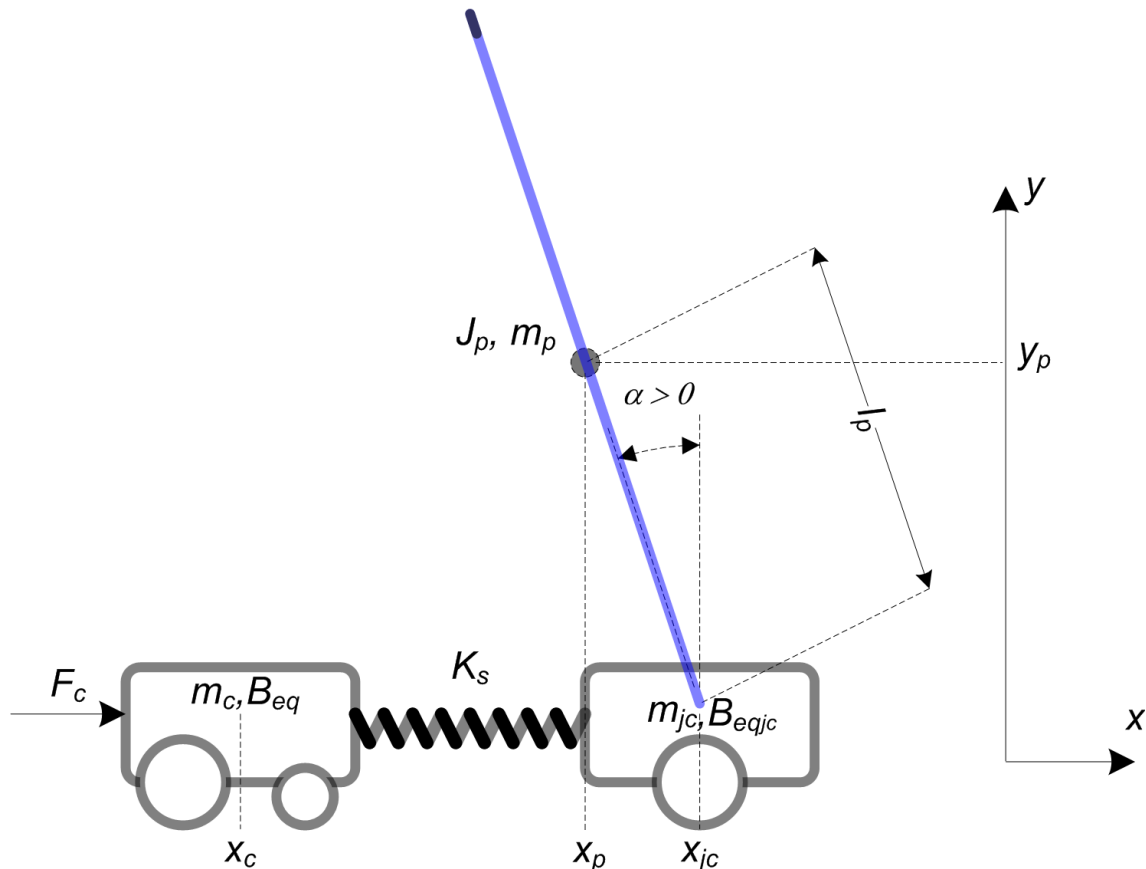

Figure 2.1: Linear flexible joint cart and single inverted pendulum conventions

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 robot manipulators with multiple joints.

The equations that describe the motions of the IP02 cart, the LFJC cart, and the pendulum with respect to the servo motor voltage, i.e. the dynamics, 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

$$q(t)^\top = \begin{bmatrix} x_c(t) & x_{jc} & \alpha \end{bmatrix}$$

where, as shown in Figure 2.1, $x_c(t)$ is the IP02 cart position, x_{jc} is the LFJC cart position, and $\alpha(t)$ is the pendulum angle. The corresponding velocities are

$$\dot{q}(t)^\top = \begin{bmatrix} \dfrac{\partial x_c(t)}{\partial t} & \dfrac{\partial x_{jc}(t)}{\partial t} & \dfrac{\partial \alpha(t)}{\partial t} \end{bmatrix}$$

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

With the generalized coordinates defined, the Euler-Lagrange equations for the rotary pendulum system are

$$\frac{\partial^2 L}{\partial t \partial \dot{x}_c} - \frac{\partial L}{\partial x_c} = Q_1$$

$$\frac{\partial^2 L}{\partial t \partial \dot{x}_{jc}} - \frac{\partial L}{\partial x_{jc}} = Q_2$$

$$\frac{\partial^2 L}{\partial t \partial \dot{\alpha}} - \frac{\partial L}{\partial \alpha} = Q_3$$

The Lagrangian of the system 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. Thus the Lagrangian is the difference between a system's kinetic and potential energies.

The generalized forces Q_i are used to describe the non-conservative forces (e.g., friction) applied to a system with respect to the generalized coordinates. In this case, the generalized force acting on the IP02 cart is

$$Q_1 = F_c - B_{eq}\dot{x}_c, \tag{2.1}$$

the force acting on the LFJC cart is

$$Q_2 = -B_{eq_{jc}}\dot{x}_{jc}, \tag{2.2}$$

and the force acting on the pendulum is

$$Q_3 = -B_p\dot{\alpha}. \tag{2.3}$$

The Euler-Lagrange equations is a systematic method of finding the equations of motion, i.e., EOMs, of a system. Once the kinetic and potential energy are obtained and the Lagrangian is found, then the task is to compute various derivatives to get the EOMs. After going through this process, the nonlinear equations of motion for the system can be obtained. **See the supplied Maple worksheet (or its equivalent HTML representation) for the complete derivation.**

Based on the system schematic shown in Figure 2.1 and the generalized forces Equation 2.1, Equation 2.2, and Equation 2.3, the first Lagrange equation can be expressed as:

$$m_c\ddot{x}_c - K_s(x_{jc} + x_c) = F_c - B_{eq}\dot{x}_c,$$

the second Lagrange equation is

$$m_p l_p \sin(\alpha)\dot{\alpha}^2 - m_p l_p \cos(\alpha)\ddot{\alpha} + (m_{jc} + m_p)\ddot{x}_{jc} + K_s(x_{jc} - x_c) = -B_{eq_{jc}}\dot{x}_{jc}$$

and the third Lagrange equation is

$$m_p l_p^2 \ddot{\alpha} - m_p l_p \ddot{x}_{jc} \cos(\alpha) - m_p l_p g \sin(\alpha) = -B_p \dot{\alpha}$$

Solving the three Lagrange equations for the second-order time derivative of the Lagrangian coordinates results in the following non-linear equations:

$$\ddot{x}_c = \frac{1}{m_c}\left(-B_{eq}\dot{x}_c - K_s x_c + K_s x_{jc} + F_c\right),$$

$$\ddot{x}_{jc} = \frac{1}{-m_p + m_p \cos(\alpha)^2 - m_{jc}}\left(m_p \sin(\alpha)l_p\dot{\alpha}^2 + \frac{\cos(\alpha)B_p\dot{\alpha}}{l_p} + B_{eq_{jc}}\dot{x}_{jc} - K_s x_c + K_s x_{jc} - \cos(\alpha)m_p g \sin(\alpha)\right),$$

and

$$\ddot{\alpha} = \frac{1}{-m_p + m_p \cos(\alpha)^2 - m_{jc}}\left(m_p \cos(\alpha)\sin(\alpha)\dot{\alpha}^2 + \frac{(B_p m_p + B_p m_{jc})\dot{\alpha}}{l_p^2 m_p} + \frac{\cos(\alpha)B_{eq_{jc}}\dot{x}_{jc}}{l_p} - \frac{\cos(\alpha)K_s x_c}{l_p}\right.$$
$$\left. + \frac{\cos(\alpha)K_s x_{jc}}{l_p} + \frac{-m_p^2 g \sin(\alpha)l_p - m_p g \sin(\alpha)l_p m_{jc}}{l_p^2 m_p}\right)$$

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

$$F_c = \frac{\eta_g K_g K_t}{R_m r_{mp}}\left(-\frac{K_g K_m \dot{x}_c}{r_{mp}} + \eta_m V_m\right) \tag{2.4}$$

See [2] for a description of the corresponding IP02 parameters (e.g. such as the back-emf constant, K_m).

2.1.3 Linearizing

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

$$\dot{x} = Ax + Bu \tag{2.5}$$

and

$$y = Cx + Du \tag{2.6}$$

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

$$x^\top = \begin{bmatrix} x_c & x_s & \alpha & \dot{x}_c & \dot{x}_s & \dot{\alpha} \end{bmatrix}$$

and

$$y^\top = \begin{bmatrix} x_1 & x_2 & x_3 \end{bmatrix}.$$

where x_s is the variation in the linear spring length defined $x_s = x_{jc} - x_c$.

After linearizing the nonlinear equations of motion about the zero angle, and substituting the state given in Equation 2.1.4, we obtain the following state-space matrices:

$$A = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & \dfrac{K_s}{m_c} & 0 & -\dfrac{B_{eq}}{m_c} & 0 & 0 \\ 0 & -\dfrac{K_s}{m_c} - \dfrac{K_s}{m_{jc}} & \dfrac{m_p g}{m_{jc}} & \dfrac{B_{eq}}{m_c} - \dfrac{B_{eq_{jc}}}{m_{jc}} & -\dfrac{B_e q_{jc}}{m_{jc}} & -\dfrac{B_p}{l_p m_{jc}} \\ 0 & -\dfrac{K_s}{l_p m_{jc}} & \dfrac{g(m_{jc} + m_p)}{l_p m_{jc}} & -\dfrac{B_{eq_{jc}}}{l_p m_{jc}} & -\dfrac{B_{eq_{jc}}}{l_p m_{jc}} & -\dfrac{B_p(m_{jc} + m_p)}{m_p l_p^2 m_{jc}} \end{bmatrix}$$

and

$$B = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \dfrac{1}{m_c} \\ -\dfrac{1}{m_c} \\ 0 \end{bmatrix}.$$

In the output equation, the position of the IP02 cart, the LFJC cart, and pendulum angle are being measured. Based on this, the C and D matrices in the output equation are

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \tag{2.7}$$

and

$$D = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}. \tag{2.8}$$

Note: The velocities of the servos and pendulum angle can be computed in the digital controller, e.g., by taking the derivative and filtering the result though a high-pass filter.

2.2 Control

In Section 2.1, we found a linear state-state space model that represents the LFJC-PEN-E system. This model is used to investigate the stability properties of the system in Section 2.2.1. In Section 2.2.2, the notion of controllability is introduced. Using the Linear Quadratic Regular algorithm, or LQR, is a common way to find the control gain and is discussed in Section 2.2.3. Lastly, Section 2.2.4 describes the state-feedback control used to control the servo position while minimizing link deflection.

2.2.1 Stability

The stability of a system can be determined from its poles ([7]):

- Stable systems have poles only in the left-hand plane.

- Unstable systems have at least one pole in the right-hand plane and/or poles of multiplicity greater than 1 on the imaginary axis.

- Marginally stable systems have one pole on the imaginary axis and the other poles in the left-hand plane.

The poles are the roots of the system's characteristic equation. From the state-space, the characteristic equation of the system can be found using

$$\det(sI - A) = 0 \qquad (2.9)$$

where *det()* is the determinant function, s is the Laplace operator, and I the identity matrix. These are the *eigenvalues* of the state-space matrix A.

2.2.2 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 ([7]).

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

$$T = \begin{bmatrix} B \ AB \ A^2 B \ldots A^n B \end{bmatrix} \qquad (2.10)$$

equals the number of states in the system,

$$\text{rank}(T) = n. \qquad (2.11)$$

2.2.3 Linear Quadratic Regular (LQR)

If (A,B) are controllable, then the Linear Quadratic Regular optimization method can be used to find a feedback control gain. Given the plant model in Equation 2.5, find a control input u that minimizes the cost function

$$J = \int_0^\infty x(t)'Qx(t) + u(t)'Ru(t)\,dt, \qquad (2.12)$$

where Q and R are the weighting matrices. The weighting matrices affect how LQR minimizes the function and are, essentially, tuning variables.

Given the control law $u = -Kx$, the state-space in Equation 2.5 becomes

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

2.2.4 Feedback Control

The feedback control loop that in Figure 2.2 is designed to balance the pendulum by regulating the pendulum angle, α.

The reference state is defined

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

and the controller is therefore
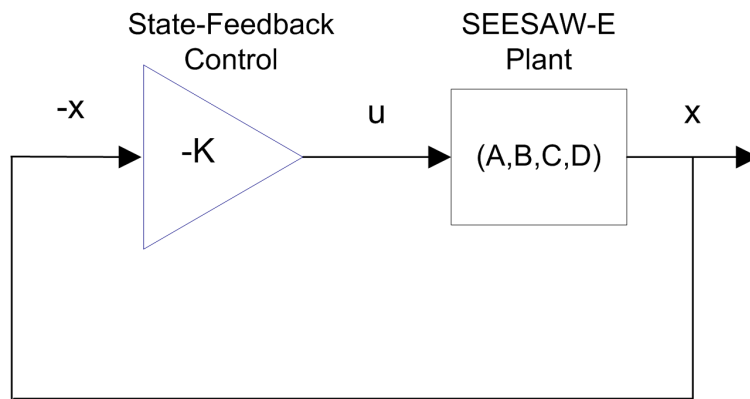
$$u = -Kx, \qquad (2.13)$$

Figure 2.2: State-feedback control loop

which is the control used in the LQR algorithm.

To eliminate IP02 servo tracking error, we can augment the system to include an integrator such that

$$\dot{\eta} = \begin{bmatrix} A & 0 \\ 1 & 0 \end{bmatrix} \eta + \begin{bmatrix} B \\ 0 \end{bmatrix} u$$

where A and B are the state-space matrices defined in Section 2.1.4 and the states are

$$\eta^\top = \begin{bmatrix} x_c & x_{jc} & \alpha & \dot{x}_c & \dot{x}_j c & \dot{\alpha} & \int x_c \, \mathrm{d}t \end{bmatrix}$$

This introduces the integration terms $\eta_7(t) = \int \theta \, \mathrm{d}t$ to the feedback controller

$$u = K(-\eta).$$

3   LAB EXPERIMENTS

3.1   Simulation

In this section we will use the Simulink diagram shown in Figure 3.1 to simulate the closed-loop control of the LFJC-PEN-E system. The system is simulated using the linear model summarized in Section 2.1. The Simulink model uses the state-feedback control described in Section 2.2.4. The feedback gain K is found using the Matlab LQR command (LQR is described briefly in Section 2.2.3). The goal is to make sure the gain used successfully stabilizes the system (i.e., keeps it balanced), minimizes the deflection of the spring, and does not saturate the dc motor.
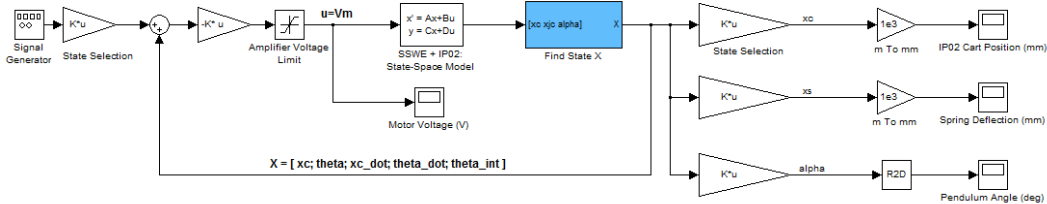

Figure 3.1: Simulink model used to simulate LFJC-PEN-E.

The state-feedback controller has proportional-derivative (PD) action and the integral (I) action. The *Find State X* block shown in Figure 3.1 creates the complete state vector using a differentiating filter and integral block. The position and velocity states of the model are then multiplied by the vector gain k computed in earlier in, i.e., $k(1 : 5) = [k1, k2, k3, k4, k5, k6, k7]$. The *Signal Generator* introduces a small disturbance into the system.

IMPORTANT: Before you can conduct these experiments, you need to make sure that the lab files are configured according to your setup. If they have not been configured already, then you need to go to Section 4 to configure the lab files first.

3.1.1   Procedure

Follow these steps to simulate the system:

1. Make sure the LQR weighting matrices in setup_ip02_lfjc_pen.m are set to

$$
Q = \begin{bmatrix}
4000 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 400 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 3000 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 100
\end{bmatrix}
$$

   and

$$ R = 0.25. $$

2. Run the script to generate the gain

$$ K = \begin{bmatrix} -138.93 & -709.32 & 304.16 & -95.71 & -116.64 & 50.05 & -20 \end{bmatrix}. $$

   LQR Tuning: When tuning the LQR, we start with the identity matrix. To put more emphasis on the IP02 cart position, and pendulum angle α, we set $Q(1,1) = 4000$ and $Q(3,3) = 3000$ respectively. The second diagonal element, $Q(2,2)$ is set to 400 to regulate the spring deflection as a secondary control goal. Finally, a value of 100 is added to the last diagonal element, $Q(7,7)$ as an integral gain for the IP02 cart to account for any steady-state error in the IP02 cart position.

3. To generate a disturbance, ensure the *Signal Generator* is set to the following:

    - Wave Form = Square
    - Amplitude = 0.02 m
    - Frequency = 0.5 Hertz

4. Open the IP02 cart position scope, *IP02 Cart Position (mm)*, the spring deflection scope, *Spring Deflection (mm)*, the pendulum angle scope, *Pendulum Angle (deg)*, and the motor input voltage scope, *Motor Voltage (V)*.

5. Start the simulation. The scopes should be displaying responses similar to Figure 3.2.
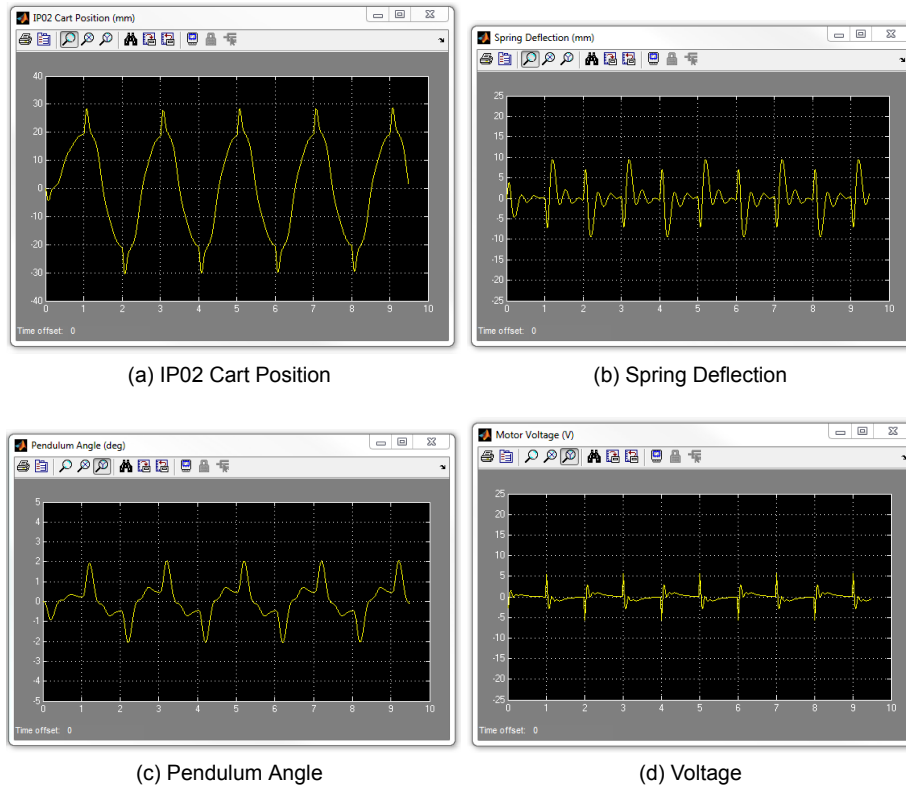

(a) IP02 Cart Position

(b) Spring Deflection

(c) Pendulum Angle

(d) Voltage

Figure 3.2: Simulated closed-loop response.

3.1.2 Analysis

The feedback response is shown in Figure 3.3. You can generate this figure by running the plot command.
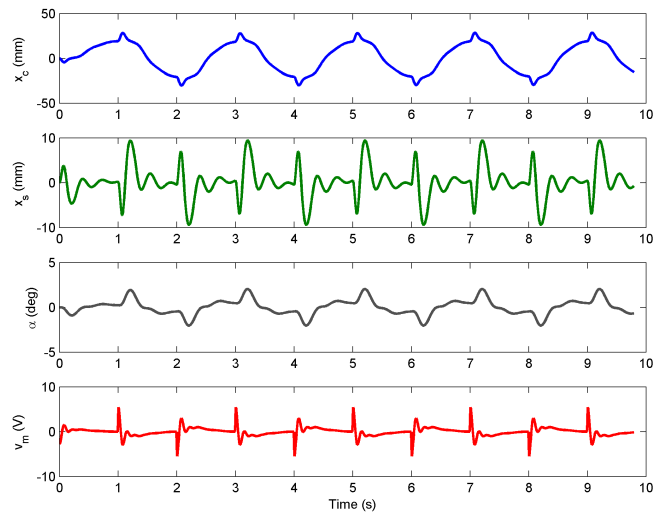

Figure 3.3: Simulated LFJC-PEN-E feedback response.

As shown by the response in Figure 3.3, the pendulum maintains its balanced vertical position without saturating the motor voltage.

Generating the Matlab figure: After each simulation run, each scope automatically saves their response to a variable in the Matlab® workspace. The *IP02 Cart Position (mm)* scope saves its response to the variable called *data_xc*, the *Spring Deflection (mm)* scope saves its data to the *data_xs* variable, the *Pendulum Angle (deg)* scope saves its data to the *data_alpha* variable, and the *Motor Voltage (V)* scope saves its plot to the *data_vm* variable.

3.2 Implementation

The q_sswe Simulink diagram shown in Figure 3.4 is used to perform the balance control on the LFJC-PEN. The *IP02 and LFJC-PEN-E* subsystem contains QUARC® blocks that interface with the DC motor and sensors of the LFJC-PEN system.
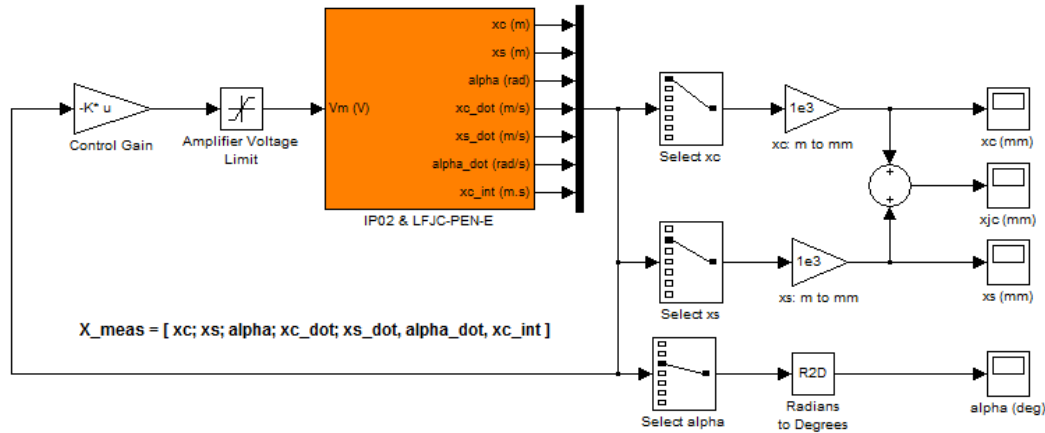

Figure 3.4: Simulink model used with QUARC® to run controller on the LFJC-PEN.

IMPORTANT: Before you can conduct these experiments, you need to make sure that the lab files are configured according to your setup. If they have not been configured already, then you need to go to Section 4 to configure the lab files first.

3.2.1 Procedure

Follow this procedure:

1. Run the setup_ip02_lfjc_pen.m script using the LQR weighting matrices that you used in the simulation in Section 3.1.

2. Open the IP02 cart position scope, *xc (mm)*, the spring deflection scope, *xs (mm)*, the pendulum angle scope, *alpha (deg)*, and the motor voltage scope, *V Command (V)*, located in the *IP02 and LFJC-PEN-E* subsystem.

3. In the Simulink diagram, go to QUARC | Build.

4. **Make sure that the pendulum is stationary in the downward (gantry) position, and the system is in the centre of the track.**

5. Click on QUARC | Start to run the controller.

6. Once the model is running, manually bring the pendulum up to its upright vertical position. You should feel the motor voltage kick-in when it is within the range where the balance control engages. The scopes should be displaying responses similar to Figure 3.5.

   Note: Once the controller has engaged, do not attempt to manually lower the pendulum. If the pendulum or carts move outside of a safe workspace, the system watchdog should halt the controller automatically.

7. If your specifications have not been met, you can finely tune the LQR weighting matrices on-the-fly using the following command:

   ```
   >> K = lqr(A,B, diag([Q(1,1),Q(1,1),Q(3,3),Q(4,4),Q(5,5),Q(6,6),Q(7,7)]),R)
   ```

8. To stop the experiment, click on the QUARC | Stop button but **make sure you catch the pendulum before it swings down**.
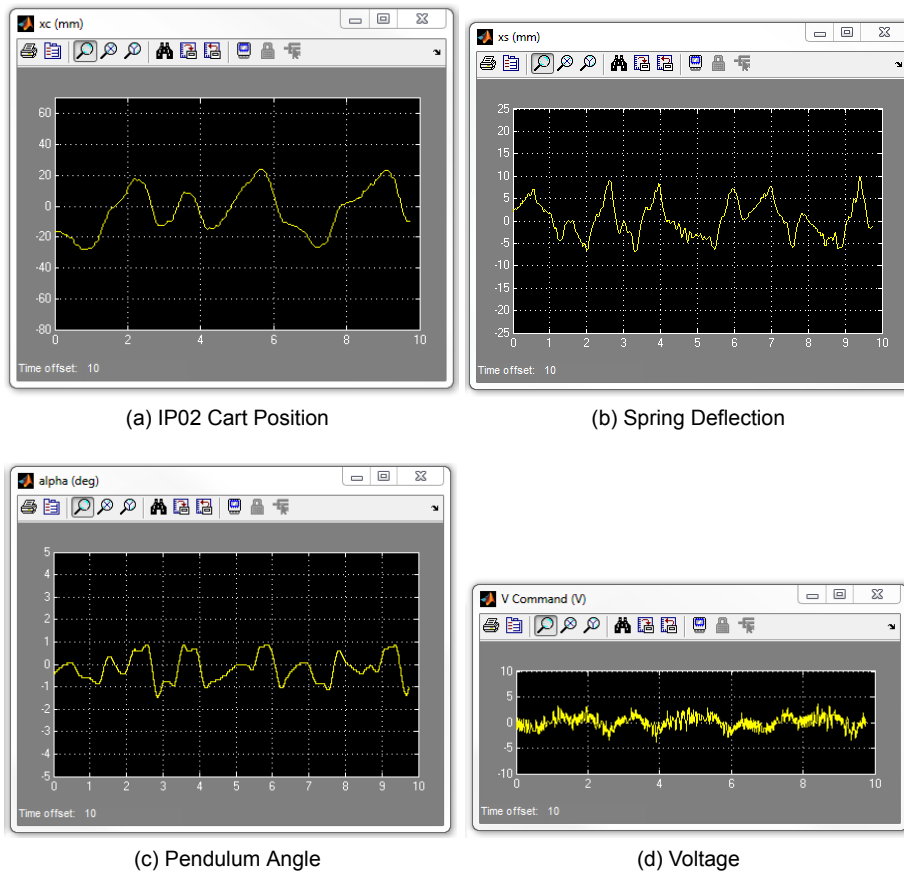
(a) IP02 Cart Position

(b) Spring Deflection

(c) Pendulum Angle

(d) Voltage

Figure 3.5: Typical response when balancing the LFJC-PEN system

3.2.2 Analysis

An example of the closed-loop balance response is shown in Figure 3.6. You can generate this using the plot command after running the q_lfjc_pen_ip02 QUARC controller.

Due to the friction in the system, the IP02 servo oscillates back-and-forth approximately ± 20 mm to balance the pendulum. The pendulum angle does not exceed 1.5 degrees when balanced, and the deflection of the spring does not exceed ± 10 mm. Because of the integrator, the IP02 cart eventually returns to the initial 0 mm setpoint.
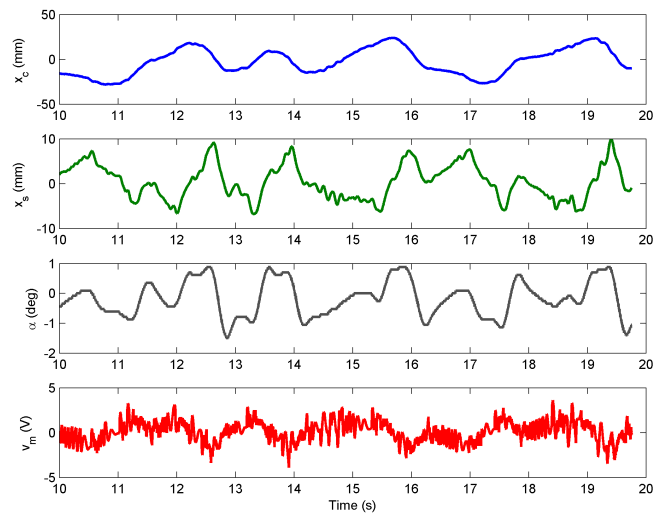
Figure 3.6: LFJC-PEN balance control response

4 SYSTEM REQUIREMENTS

Required Software

- Microsoft Visual Studio (MS VS)
- Matlab® with Simulink®, Real-Time Workshop, and the Control System Toolbox
- QUARC®

See the QUARC® software compatibility chart in [4] to see what versions of MS VS and Matlab are compatible with your version of QUARC and for what OS.

Required Hardware

- Data acquisition (DAQ) device **with 3x encoder inputs** and that is compatible with QUARC®. This includes Quanser DAQ boards such as Q8-USB, QPID, and QPIDe and some National Instruments DAQ devices. For a full listing of compliant DAQ cards, see Reference [1].
- Quanser IP02 linear servo.
- Quanser LFJC-PEN-E (attached to the IP02).
- Quanser VoltPAQ-X1 power amplifier, or equivalent.

Before Starting Lab

Before you begin this laboratory make sure:

- QUARC® is installed on your PC, as described in [3].
- DAQ device has been successfully tested (e.g., using the test software in the Quick Start Guide or the *QUARC Analog Loopback Demo*).
- LFJC-PEN-E and amplifier are connected to your DAQ board as described its User Manual [6].

4.1 Overview of Files

File Name	Description
LFJC-PEN-E User Manual.pdf	This manual describes the hardware of the LFJC-PEN system and explains how to setup and wire the system for the experiments.
LFJC-PEN-E Laboratory Guide.pdf	This document demonstrates how to obtain the linear state-space model of the system, simulate the closed-loop system, and implement controllers on the LFJC-PEN plant using QUARC®.
setup_ip02_lfjc_pen.m	The main Matlab script that sets the IP02 and LFJC-PEN motor, sensor, and configuration-dependent parameters. Run this file only to setup the laboratory.
config_ip02.m	Returns the configuration-based IP02 model, sensor, and motor specifications.
config_lfjc.m	Returns the LFJC-PEN model parameters.
config_sp.m	Returns the pendulum model parameters.
s_lfjc_pen.mdl	Simulink file that simulates the closed-loop control of a LFJC-PEN-E system using state-feedback control.
q_lfjc_pen.mdl	Simulink file that implements the state-feedback control on the LFJC-PEN-E system using QUARC®.
LFJC-PEN-E.mws	Maple worksheet used to develop the model for the LFJC-PEN experiment. Waterloo Maple 9, or a later release, is required to open, modify, and execute this file.
LFJC-PEN-E.html	HTML presentation of the Maple Worksheet. It allows users to view the content of the Maple file without having Maple 9 installed. No modifications to the equations can be performed when in this format.

Table 4.1: Files supplied with the LFJC-PEN

4.2 Setup for Simulation

Before beginning the in-lab procedure outlined in Section 3.1, the s_lfjc_pen Simulink diagram and the setup_ip02_lfjc_pen.m script must be configured.

Follow these steps:

1. Load the Matlab software.

2. Browse through the *Current Directory* window in Matlab and find the folder that contains the file *setup_ip02_lfjc_pen.m*.

3. Open the *setup_ip02_lfjc_pen.m* script.

4. **Configure setup_ip02_lfjc_pen script**: When used with the LFJC-PEN, the IP02 has the additional weight, the LFJC-PEN has two weights, and the long pendulum is used. Make sure the script is setup to match this setup:

   - IP02_WEIGHT_TYPE to 'WEIGHT'
   - LFJC_WEIGHT_TYPE to 'TWO_WEIGHT';
   - PEND_TYPE to 'LONG_24IN';
   - K_AMP to 1 (unless your amplifier gain is different)
   - AMP_TYPE to your amplifier type (e.g., VoltPAQ).

- Ensure other parameters such as VMAX_DAC match your system configuration.

5. Run setup_ip02_lfjc_pen.m to setup the Matlab workspace.

6. Open the *s_lfjc_pen.mdl* Simulink diagram, shown in Figure 3.1.

4.3 Setup for Running on LFJC-PEN

Before performing the in-lab exercises in Section 3.2, the s_lfjc_pen Simulink diagram and the setup_ip02_lfjc_pen.m script must be configured.
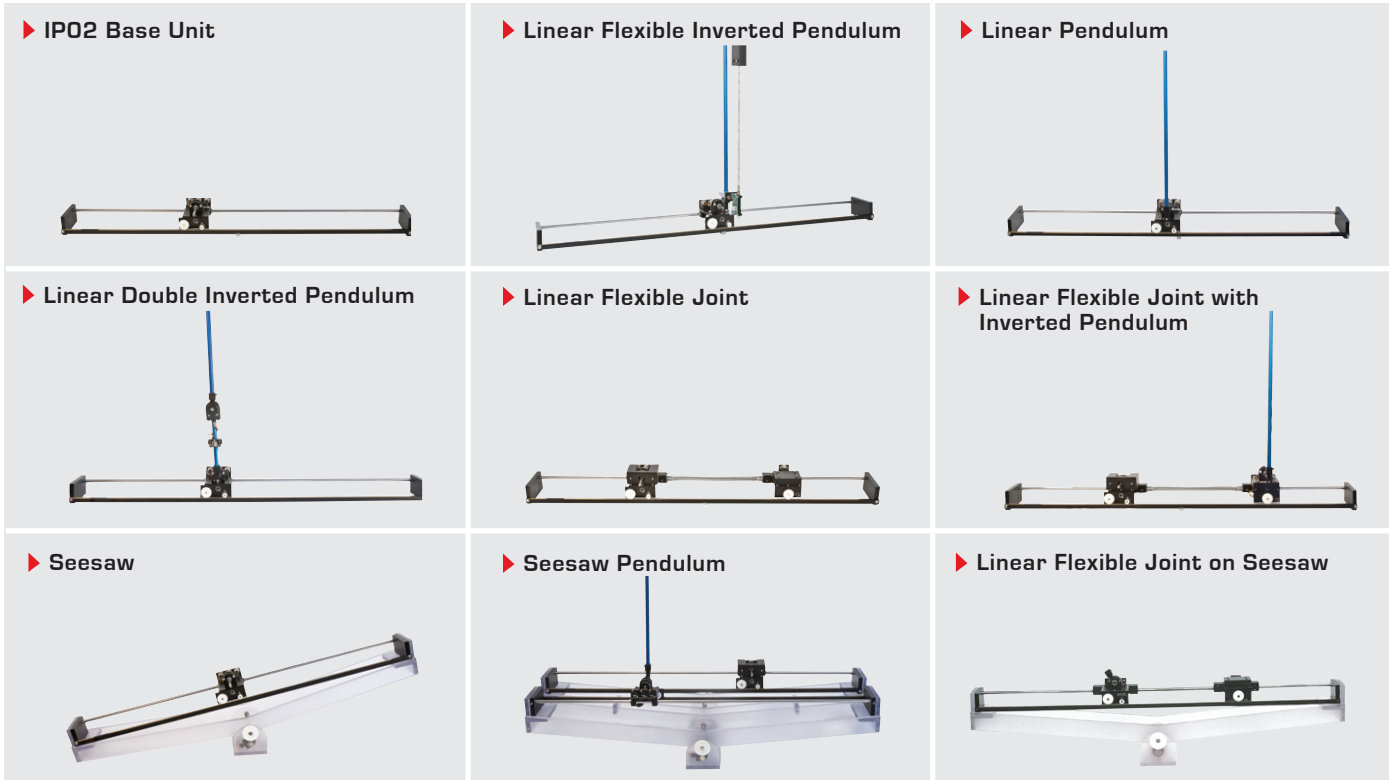
Follow these steps to get the system ready for this lab:

1. Setup the IP02 with the LFJC-PEN module as detailed in the LFJC-PEN User Manual [6].

2. **Make sure that the pendulum is stationary in the downward (gantry) position, and the system is in the centre of the track.**

3. Configure and run setup_ip02_lfjc_pen.m as explained in Section 4.2.

4. Open the *q_lfjc_pen.mdl* Simulink diagram, shown in Figure 3.4.

5. **Configure DAQ:** Ensure the HIL Initialize block in the *IP02 and LFJC-PEN-E* subsystem is configured for the DAQ device that is installed in your system. See Reference [1] for more information on configuring the HIL Initialize block.

REFERENCES

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

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

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

[4] Quanser Inc. *QUARC Compatibility Table*, 2010.

[5] Quanser Inc. *IP02 Lab Workbook (QUARC)*, 2012.

[6] Quanser Inc. *SEESAW-E User Manual*, 2012.

[7] Norman S. Nise. *Control Systems Engineering*. John Wiley & Sons, Inc., 2008.

Nine linear motion plants for teaching fundamental and advanced controls concepts

▶ **IP02 Base Unit**

▶ **Linear Flexible Inverted Pendulum**

▶ **Linear Pendulum**

▶ **Linear 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.

Solutions for teaching and research. Made in Canada.