



# LABORATORY GUIDE

## Rotary Double Inverted Pendulum Experiment for LabVIEW™ Users

Developed by:  
Jacob Apkarian, Ph.D., Quanser  
Michel Lévis, M.A.SC., Quanser

Quanser educational solutions  
are powered by:



CAPTIVATE. MOTIVATE. GRADUATE.

© 2012 Quanser Inc., All rights reserved.

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>

This document and the software described in it are provided subject to a license agreement. Neither the software nor this document may be used or copied except as specified under the terms of that license agreement. All rights are reserved and no part may be reproduced, stored in a retrieval system or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of Quanser Inc.

# CONTENTS

<b>1</b>	<b>Introduction</b>	<b>4</b>
<b>2</b>	<b>Background</b>	<b>5</b>
	2.1 Modeling	5
	2.2 Control	8
<b>3</b>	<b>Lab Experiments</b>	<b>11</b>
	3.1 Simulation	11
	3.2 Implementation	13
<b>4</b>	<b>System Requirements</b>	<b>15</b>
	4.1 Overview of Files	16
	4.2 Setup for Simulation	16
	4.3 Setup for Running on DBPEN-ROT	17

# 1 INTRODUCTION

This laboratory manual describes how to design a state-feedback control system that balances a rotary double inverted pendulum and positions the rotary arm to a commanded angular position.

The plant has two main components: the Quanser SRV02 rotary motion plant and the Quanser Rotary Double Inverted Pendulum, i.e., DBPEN-ROT module. The double pendulum is composed of a rotary arm that attaches to the SRV02 system, a short 7-inch bottom blue rod, an encoder hinge, and the top 12-inch blue rod.

## Topics Covered

- Obtain a state-space representation of the open-loop system.
- Design a the state-feedback gain for the closed-loop system using Linear-Quadratic Regulator (LQR) optimization.
- Simulate the system and ensure it is stabilized using the designed state-feedback control.
- Implement the state-feedback controller on the DBPEN-ROT system and evaluate its actual performance.

## Prerequisites

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

1. Hardware and software requirements given in Section 4.
2. Modeling and state-space representation.
3. State-feedback design using Linear-Quadratic Regular (LQR) optimization.
4. Basics of [LabVIEW™](#) .
5. LabVIEW Integration lab detailed in Appendix A in the SRV02 Workbook [2].

# 2 BACKGROUND

## 2.1 Modeling

### 2.1.1 Model Convention

The rotary pendulum model is shown in Figure 2.1. The rotary arm pivot is attached to the SRV02 system and is actuated. The arm has a length of  $L_r$ , a moment of inertia of  $J_r$ , and its angle,  $\theta$ , increases positively when it rotates counter-clockwise (CCW). The servo (and thus the arm) should turn in the CCW direction when the control voltage is positive, i.e.,  $V_m > 0$ .

The double-pendulum assembly is connected to the end of the rotary arm. The short-sized, bottom pendulum has a total length of  $L_{p1}$  and a center of mass of  $l_{p1}$ . The moment of inertia about its center of mass is  $J_{p1}$  and it has a mass of  $M_{p1}$ . The top medium-sized pendulum has a total length of  $L_{p2}$ , a center of mass of  $l_{p2}$ , a moment of inertia of  $J_{p2}$ , and a mass  $M_{p2}$ . The short bottom pendulum angle,  $\alpha$ , and the medium top pendulum angle,  $\phi$ , are both zero when it is perfectly upright in the vertical position and they increase positively when rotated CCW. The hinge between the two pendulum has a mass of  $M_h$ .

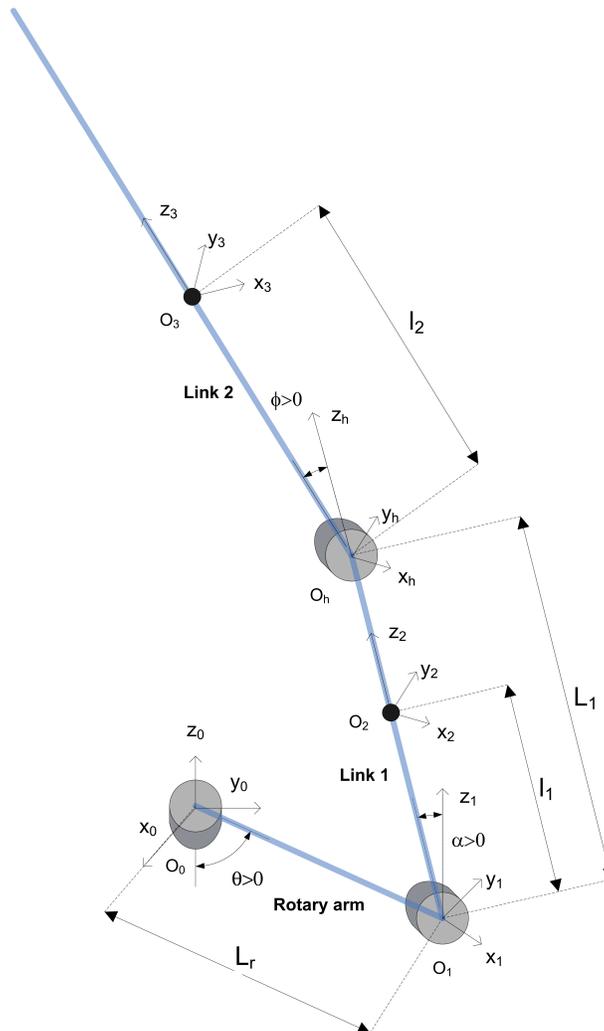


Figure 2.1: Rotary Double-Inverted Pendulum conventions

## 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.

More specifically, the equations that describe the motions of the rotary arm 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

$$q(t)^\top = [\theta(t) \quad \alpha(t) \quad \phi(t)]$$

where, as shown in Figure 2.1,  $\theta(t)$  is the rotary arm angle and  $\alpha(t)$  is the inverted pendulum angle. The corresponding velocities are

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

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

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

$$\begin{aligned} \frac{\partial^2 L}{\partial t \partial \dot{\theta}} - \frac{\partial L}{\partial \theta} &= Q_1 \\ \frac{\partial^2 L}{\partial t \partial \dot{\alpha}} - \frac{\partial L}{\partial \alpha} &= Q_2 \\ \frac{\partial^2 L}{\partial t \partial \dot{\phi}} - \frac{\partial L}{\partial \phi} &= Q_3 \end{aligned}$$

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 rotary arm is

$$Q_1 = \tau - D_r \dot{\theta},$$

and acting on the bottom and top pendulum are

$$Q_2 = -D_{p1} \dot{\alpha}$$

and

$$Q_3 = -D_{p2} \dot{\phi}.$$

See [1] for a description of the corresponding SRV02 parameters (e.g. such as the back-emf constant,  $k_m$ ). Our control variable is the input servo motor voltage,  $V_m$ . Opposing the applied torque is the viscous friction torque, or viscous damping, corresponding to the term  $D_r$ . Since the pendulum is not actuated, the only force acting on the link is the damping. The viscous damping coefficient of the short (bottom) and medium (top) pendulums are denoted by  $D_{p1}$  and  $D_{p2}$ .

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.**

The torque applied at the base of the rotary arm (i.e., at the load gear) is generated by the servo motor as described by the equation

$$\tau = \frac{\eta_g K_g \eta_m k_t (V_m - K_g k_m \dot{\theta})}{R_m}. \quad (2.1)$$

See [1] for a description of the corresponding SRV02 parameters (e.g. such as the back-emf constant,  $k_m$ ).

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

$$J\ddot{x} + b\dot{x} + g(x) = \tau_1$$

where  $x$  is an angular position,  $J$  is the moment of inertia,  $b$  is the damping,  $g(x)$  is the gravitational function, and  $\tau_1$  is the applied torque (scalar value).

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

$$D(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) = \tau \quad (2.2)$$

where  $D$  is the inertial matrix,  $C$  is the damping matrix,  $g(q)$  is the gravitational vector, and  $\tau$  is the applied torque vector.

### 2.1.3 Linearizing

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

$$z^T = [z_1 \ z_2]$$

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

$$z_0^T = [a \ b]$$

The linearized function is

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

### 2.1.4 Linear State-Space Model

The linear state-space equations are

$$\dot{x} = Ax + Bu \quad (2.3)$$

and

$$y = Cx + Du \quad (2.4)$$

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

$$x^T = [\theta \ \alpha \ \phi \ \dot{\theta} \ \dot{\alpha} \ \dot{\phi}]$$

and

$$y^T = [x_1 \ x_2 \ x_3].$$

After linearizing the nonlinear equations of motion, solving for the acceleration terms, i.e.,  $\ddot{\theta}$ ,  $\ddot{\alpha}$ , and  $\ddot{\phi}$ , and substituting the state given in , we obtain the following state-space matrices:

$$A = \frac{1}{J_T} \begin{bmatrix} 0 & 0 & & 0 & & 1 & 0 & 0 \\ 0 & 0 & & 0 & & 0 & 1 & 0 \\ 0 & 0 & & 0 & & 0 & 0 & 1 \\ 0 & a_{42} & & -M_{p1}l_{p1}L_rM_{p2}g(-l_{p1} + L_{p1}) & & 0 & 0 & 0 \\ 0 & a_{52} & & -M_{p2}g(-L_r^2L_{p1}M_{p1} + L_{p1}L_r^2M_{p1} + J_rL_{p1}) & & 0 & 0 & 0 \\ 0 & a_{62} & & a_{63} & & 0 & 0 & 0 \end{bmatrix}$$

where

$$\begin{aligned} a_{42} &= L_r g (M_{p1}^2 l_{p1}^2 + 2M_{p1} l_{p1} M_h L_{p1} M_h L_{p1}^2 M_{p2} M_{p1} l_{p1}^2 M_{p2} M_h^2 L_{p1}^2) \\ a_{52} &= g (L_r^2 L_{p1} M_{p1} M_{p2} + L_{p1} L_r^2 M_h M_{p2} + L_{p1} L_r^2 M_h^2 + L_r^2 L_{p1} M_{p1}^2 + J_r L_{p1} M_h + J_r L_{p1} M_{p1} \\ &\quad + L_{p1} L_r^2 M_h M_{p1} + L_r^2 L_{p1} M_h M_{p1}) \\ a_{62} &= -\frac{g}{L_{p2}} \left( L_{p1} L_r^2 L_{p1} M_{p1}^2 + L_{p1} L_r^2 L_{p2} M_h M_{p1} + L_r^2 L_{p1} L_{p2} M_h M_{p1} + L_{p1} L_r^2 L_{p2} M_h^2 \right. \\ &\quad + L_r^2 L_{p1} L_{p2} M_{p1}^2 - M_{p1} L_{p1}^2 L_r^2 M_{p2} + L_r^2 L_{p1} L_{p2} M_{p1} M_{p2} + L_{p1} L_r^2 L_{p2} M_h M_{p2} + J_r L_{p1} L_{p2} M_h + J_r L_{p1} L_{p2} M_{p1} \\ &\quad \left. - M_{p1} L_{p1}^2 J_r + L_{p1} J_r L_{p1} M_{p1} - M_{p1} L_{p1}^2 M_h L_r^2 + L_r^2 L_{p1} M_{p2} M_{p1} L_{p1} + M_{p1} L_{p1} L_r^2 M_h L_{p1} - L_r^2 M_{p1}^2 L_{p1}^2 \right) \\ a_{63} &= \frac{g}{L_{p2}} \left( J_r L_{p1} L_{p2} M_{p2} + J_r L_{p1}^2 M_{p2} + L_{p1}^2 L_r^2 M_{p1} M_{p2} + M_{p1} L_{p1}^2 L_r^2 M_{p2} + L_{p1} L_r^2 L_{p2} M_{p1} M_{p2} \right. \\ &\quad - L_r^2 L_{p1} L_{p2} M_{p1} M_{p2} + M_{p1} L_{p1}^2 J_r + M_h L_{p1}^2 J_r + M_h L_{p1}^2 M_{p1} L_r^2 + M_{p1} L_{p1}^2 M_h L_r^2 \\ &\quad \left. - 2L_r^2 L_{p1} M_{p2} M_{p1} L_{p1} - 2M_{p1} L_{p1} L_r^2 M_h L_{p1} \right) \end{aligned}$$

and

$$B = \frac{1}{J_T} \begin{bmatrix} 0 \\ 0 \\ 0 \\ M_h L_{p1}^2 + M_{p1} L_{p1}^2 \\ L_r (M_h L_{p1} + M_{p1} L_{p1}) \\ -\frac{L_r}{L_{p2}} (-M_{p1} L_{p1}^2 + M_{p1} L_{p1} L_{p2} + M_h L_{p1} L_{p2} + M_{p1} L_{p1} L_{p1}) \end{bmatrix}$$

In the output equation, only the position of the servo and link angles are being measured. Based on this, the  $C$  and  $D$  matrices in the output equation are

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \quad (2.5)$$

and

$$D = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}. \quad (2.6)$$

**Note:** In the above model, we ignore the viscous damping parameters  $D_r$ ,  $D_{p1}$ , and  $D_{p2}$ . For the full state-space system, see the Maple worksheet.

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

## 2.2 Control

In Section 2.1, we found a linear state-state space model that represents the Rotary Double Inverted Pendulum 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 ([4]):

- 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

$$\det(sI - A) = 0 \tag{2.7}$$

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  $i = 1 \dots n$ , from an initial state to a final state then the system is controllable, otherwise it is uncontrollable ([4]).

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

$$T = [B \ AB \ A^2B \ \dots \ A^nB] \tag{2.8}$$

equals the number of states in the system,

$$\text{rank}(T) = n. \tag{2.9}$$

### 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 2.3, find a control input  $u$  that minimizes the cost function

$$J = \int_0^\infty x(t)'Qx(t) + u(t)'Ru(t) dt, \tag{2.10}$$

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 2.3 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 stabilize the servo to a desired position,  $\theta_d$ , while minimizing the deflection of the pendulum.

The reference state is defined

$$x_d = [\theta_d \ 0 \ 0 \ 0 \ 0 \ 0]$$

and the controller is

$$u = K(x_d - x). \tag{2.11}$$

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

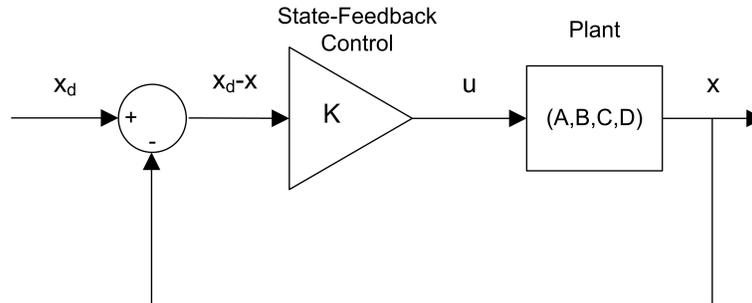


Figure 2.2: State-feedback control loop

To eliminate 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

$$\eta^T = [\theta \ \alpha \ \phi \ \dot{\theta} \ \dot{\alpha} \ \dot{\phi} \ \int \theta \, dt]$$

and

$$\eta_d^T = [\theta_d \ 0 \ 0 \ 0 \ 0 \ 0 \ \int \theta_d \, dt].$$

This introduces the integration terms  $\eta_7(t) = \int \theta \, dt$  to the feedback controller

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

# 3 LAB EXPERIMENTS

## 3.1 Simulation

In this section we will use the VI shown in Figure 3.1 to simulate the closed-loop control of the Rotary Double Inverted Pendulum system. The system is simulated using the linear model summarized in Section 2.1. The VI uses the state-feedback control described in Section 2.2.4. The feedback gain  $K$  is found using the LQR command from the *Control Design and Simulation Toolkit* (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 inverted), tracks the reference servo position, and does not saturate the dc motor.

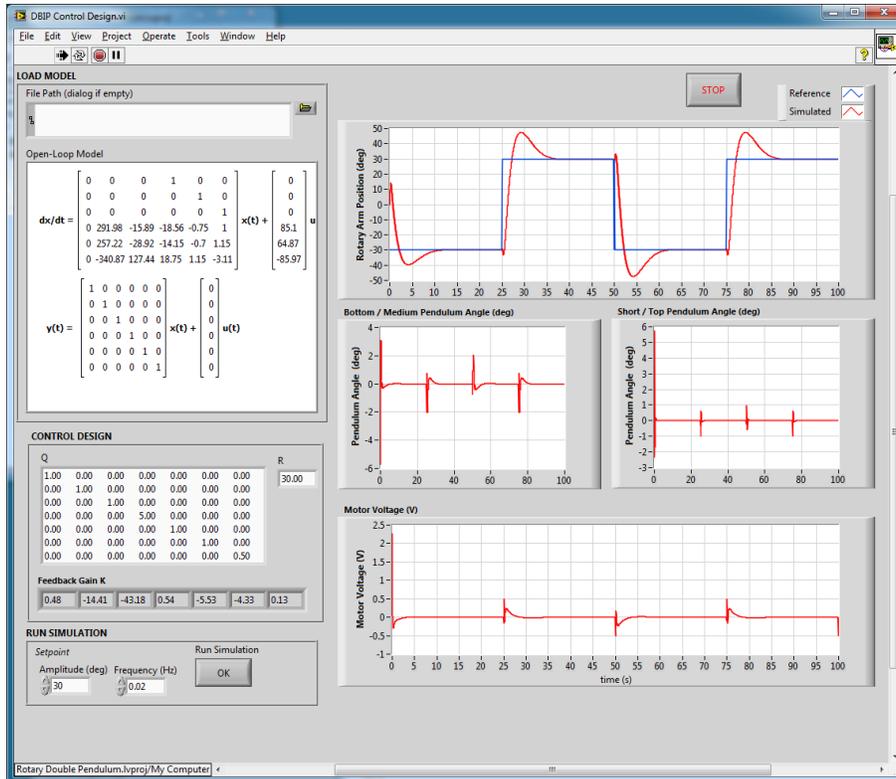


Figure 3.1: VI used to simulate Rotary Double Inverted Pendulum.

**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. Open and run the *DBIP Control Design.vi* as described in Section 4. Make sure you choose your model file using the *File Path* control.

2. By default, the Q matrix is set to identity matrix. Set the LQR weighting matrices in to

$$Q = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 5 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.5 \end{bmatrix}$$

and

$$R = 30.$$

3. This automatically generates the gain

$$K = [0.48 \quad -14.4 \quad -43.2 \quad 0.54 \quad -5.53 \quad -4.33 \quad 0.13].$$

**LQR Tuning:** When tuning the LQR, we start with the identity matrix. To put more emphasis on the top / medium pendulum angle  $\phi$ , we set  $Q(4, 4) = 5$ . The last diagonal element,  $Q(7, 7)$  is set to 0.5 to generate a small integral gain for the servo tracking.

4. In the *Run Simulation* section, set the *Amplitude* to 30 deg and the *Frequency* to 0.02 Hz. This will generate a  $\pm 30$  deg 0.02 Hz square wave reference.
5. Click on *Run Simulation OK* button. By default, the simulation runs for 100 seconds. The scopes should be displaying responses similar to Figure 3.2. Note that in the servo scope, the blue trace is the setpoint position and the red trace is the simulated position.

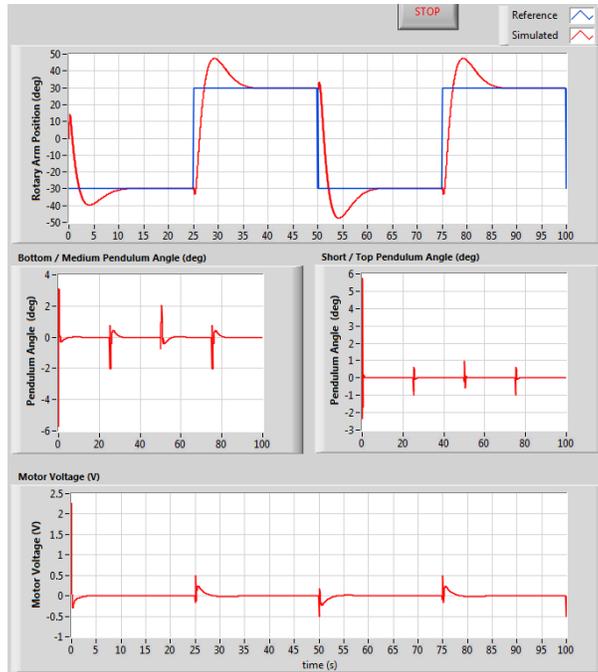


Figure 3.2: Simulated closed-loop response.

6. This is an interactive design process. You can go back and change your Q and R matrices, acquire a new control gain  $K$ , and then run the simulation again by pressing the OK button.
7. Click on the *STOP* button to stop running the VI.

## 3.1.2 Analysis

As shown by the response in Figure 3.2, the pendulum maintains its balance about the upright vertical position while tracking the  $\pm 30$  degree servo angle.

## 3.2 Implementation

The *DBIP Control* VI shown in Figure 3.3 is used to perform the balance control on the DBPEN-ROT. The VI contains **Quanser Rapid Control Prototyping Toolkit**<sup>®</sup> blocks that interface with the dc motor and sensors of the DBPEN-ROT system. The VI enables the balance control when the pendulum angles are within the  $\pm 10$  degrees of the inverted vertical position.

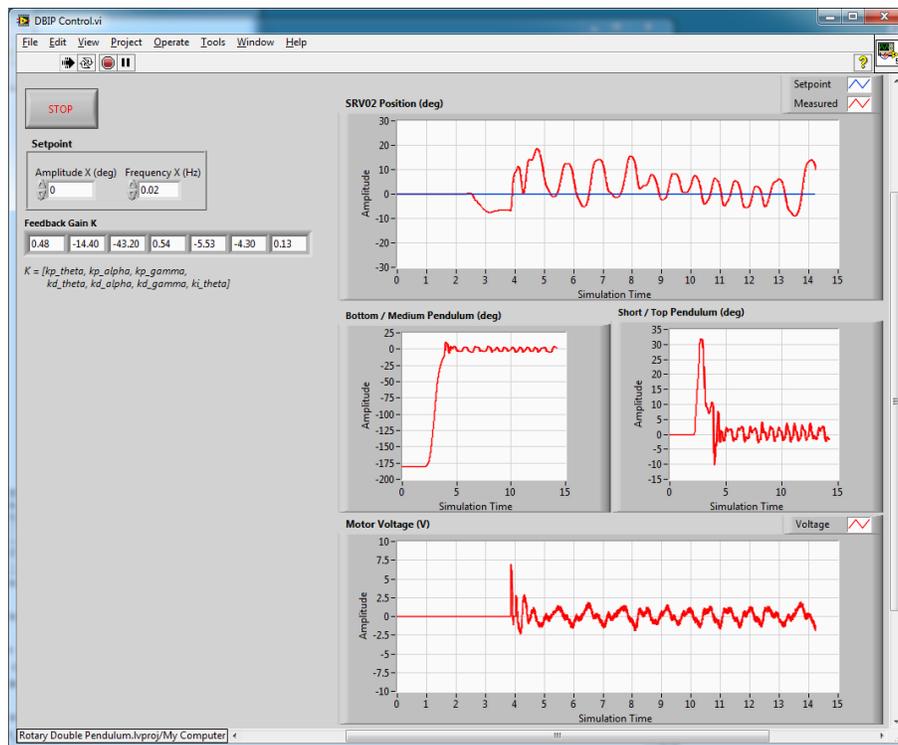


Figure 3.3: VI used to run controller on the DBPEN-ROT.

**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. Set the *Amplitude* control to 0 to turn off the servo setpoint.
2. Make sure gain K is set to the gain you found and simulated in Section 3.1.
3. **Make sure the pendulum assembly is in the hanging down position and motionless.** The Simulink diagram will measure the short pendulum angle,  $\alpha$ , as -180 degrees.
4. Run the VI.

- Manually rotate the pendulum assembly to the upright vertical position. Do this slowly and keep the bottom and top pendulums straight and in-line with each other. Once the control engages, immediately release the pendulum and let the servo balance the links. The scopes should be displaying responses similar to Figure 3.4. Note that in the  $\theta_x$  (deg) and  $\theta_y$  (deg) scopes, the blue trace is the setpoint position while the red trace is the measured position.

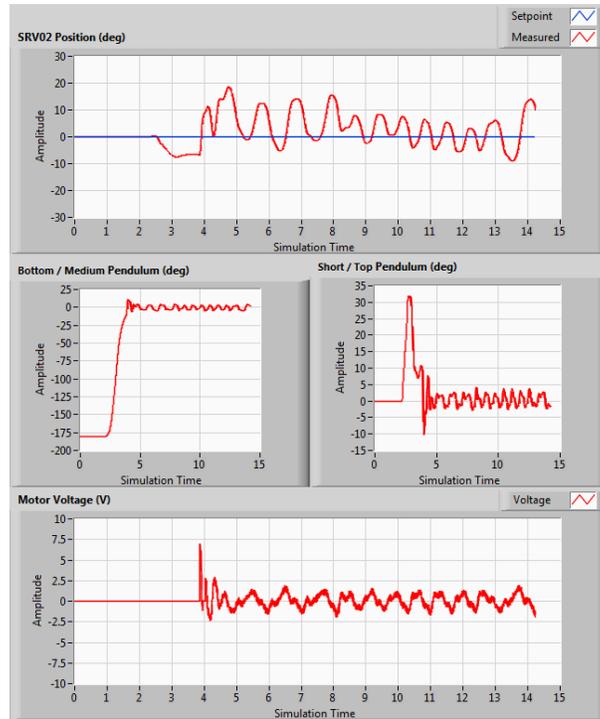


Figure 3.4: Typical response when running control on DBPEN-ROT system

- To stop the experiment, click on the *Stop* button but make sure you catch the pendulum before it swings down. This will prevent the assembly from hitting the table surface.

### 3.2.2 Analysis

The balance control response is shown in Figure 3.4. Due to the friction in the system, the servo oscillates back-and-forth approximately  $\pm 10$  degrees to balance the double pendulum. Because of the integrator, the servo eventually stabilizes about the 0 degree setpoint. In the  $\alpha$  (deg) scope, the pendulum angle goes up from -180 degrees to 0 at which point the balance control is engaged (around the 2.5 second mark).

# 4 SYSTEM REQUIREMENTS

## Required Software

Make sure **LabVIEW™** is installed with the following required add-ons:

1. **LabVIEW™**
2. NI-DAQmx
3. NI **LabVIEW™** Control Design and Simulation Module
4. NI **LabVIEW™** MathScript RT Module
5. **Quanser Rapid Control Prototyping Toolkit®**

**Note:** Make sure the Quanser Rapid Control Prototyping (RCP) Toolkit is installed after LabVIEW. See the RCP Toolkit Quick Start Guide for more information.

## Required Hardware

- Data acquisition (DAQ) device **with 3x encoder inputs** and that is compatible with **Quanser Rapid Control Prototyping Toolkit®**.
- Quanser SRV02-ET rotary servo.
- Quanser Rotary Double Inverted Pendulum (attached to the load gear of the SRV02).
- Quanser VoltPAQ-X1 power amplifier, or equivalent.

## Before Starting Lab

Before you begin this laboratory make sure:

- **LabVIEW™** is installed on your PC.
- DAQ device has been successfully tested (e.g., using the test software in the Quick Start Guide or the *Analog Loopback Demo*).
- Rotary Double Inverted Pendulum and amplifier are connected to your DAQ board as described its User Manual [3].

## 4.1 Overview of Files

File Name	Description
Rotary Double Inverted Pendulum User Manual.pdf	This manual describes the hardware of the DBPEN-ROT system and explains how to setup and wire the system for the experiments.
Rotary Double Inverted Pendulum 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 DBPEN-ROT plant using LabVIEW™.
Rotary Double Pendulum.lvproj	LabVIEW project that contains all the VIs required for the lab.
DBIP Modeling.vi	VI used to generate the linear state-space model of the DBPEN-ROT system.
DBIP Control Design.vi	VI used to design the LQR state-feedback gain and simulate the DBPEN-ROT system.
DBIP Control.vi	VI that implements the state-feedback control on the DBPEN-ROT system.
dbpen-rot.mws	Maple worksheet used to develop the model for the DBPEN-ROT experiment. Waterloo Maple 9, or a later release, is required to open, modify, and execute this file.
dbpen-rot.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 DBPEN-ROT

## 4.2 Setup for Simulation

Before beginning the in-lab procedure outlined in Section 3.1, the modeling and control design VIs must be configured.

Follow these steps:

1. Load LabVIEW™.
2. Open the *Rotary Double Pendulum.lvproj* LabVIEW project, shown in Figure 4.1.
3. Open the *DBIP Modeling.vi* shown in Figure 4.2.
4. The pendulum and SRV02 parameters are already set, by default. Run the VI to generate the linear state-space model.
5. In *Model Name*, enter the name of the model you and click on OK. This will save the state-space model under the folder *DBIP Model*. You can close this VI now.
6. Open the *DBIP Control Design* VI, shown in Figure 3.1.
7. Using the *File Path* control, select the model file.
8. Run the VI. The state-space model should load. You are now ready to design your LQR control and simulate the closed-loop response.

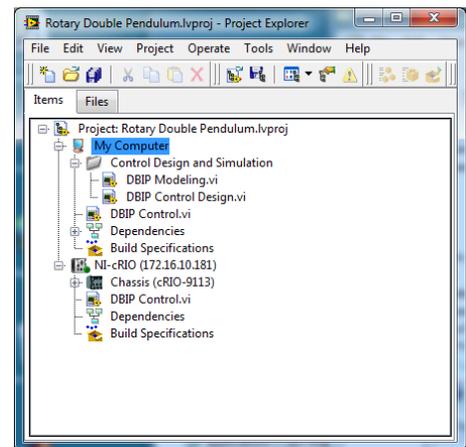


Figure 4.1: LabVIEW Rotary Double Inverted Pendulum Project

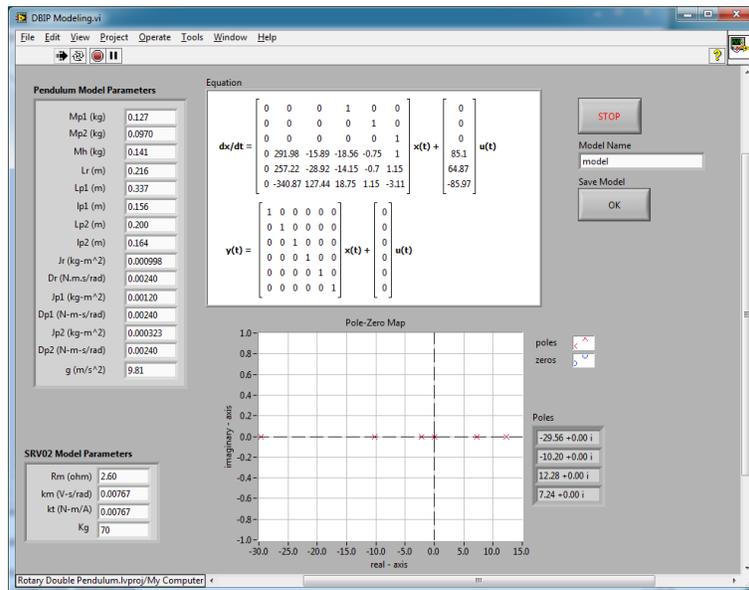


Figure 4.2: Rotary Double Inverted Pendulum Modeling VI

## 4.3 Setup for Running on DBPEN-ROT

Before performing the in-lab exercises in Section 3.2, the DBPEN-ROT system and the *DBIP Control.vi* must be configured properly.

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

1. Setup the SRV02 with the DBPEN-ROT module as detailed in the DBPEN-ROT User Manual [3].
2. **Make sure the pendulum is in the hanging, downward position.** For more information, go to the DBPEN-ROT User Manual [3].
3. Open the *DBIP Control.vi*, shown in Figure 3.3.
4. Set gain  $K$  control in the VI to the value found in Section 4.2 (or another gain you want to test on the system).
5. **Configure DAQ:** Ensure the HIL Initialize block is configured for the DAQ device that is installed in your system. To do this, go to the block diagram (CTRL-E) and double click on the **HIL Initialize** Express VI shown in Figure 4.3.

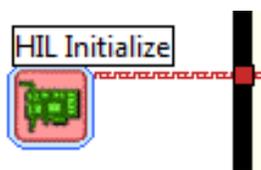


Figure 4.3: HIL Initialize Express VI

6. Under the Main tab, select the data acquisition device that is installed on your system in the *Board type* section. For example, in Figure 4.4 the Q8-USB is chosen.

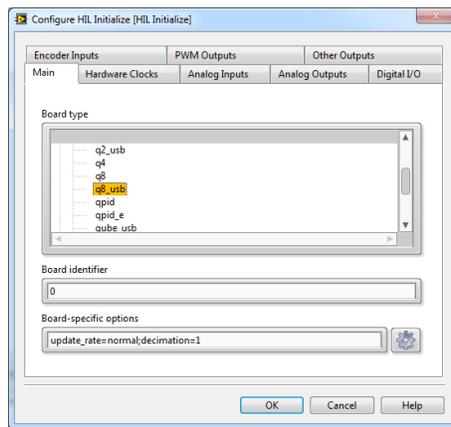


Figure 4.4: Select DAQ board that will be used to control system

# REFERENCES

- [1] Quanser Inc. *SRV02 User Manual*, 2009.
- [2] Quanser Inc. SRV02 lab manual. 2011.
- [3] Quanser Inc. *Rotary Double Inverted Pendulum User Manual*, 2012.
- [4] Norman S. Nise. *Control Systems Engineering*. John Wiley & Sons, Inc., 2008.

## Over ten rotary experiments for teaching fundamental and advanced controls concepts



Quanser's rotary collection allows you to create experiments of varying complexity – from basic to advanced. Your lab starts with the Rotary Servo Base Unit and is designed to help engineering educators reach a new level of efficiency and effectiveness in teaching controls in virtually every engineering discipline including electrical, computer, mechanical, aerospace, civil, robotics and mechatronics. For more information please contact [info@quanser.com](mailto:info@quanser.com)

©2012 Quanser Inc. All rights reserved.