QUANSER
INNOVATE. EDUCATE.

2 DOF Ball Balancer

Workbook

2DBB

Student Version

Quanser Inc.
2023

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

ACKNOWLEDGEMENTS

Quanser, Inc. would like to thank Dr. Hakan Gurocak, Washington State University Vancouver, USA, for his help to include embedded outcomes assessment.

CONTENTS

1 INTRODUCTION

The challenge of the 2 DOF Ball Balancer experiment, i.e., 2DBB, is to design a PID-based controller that stabilizes the ball to a desired position on the balance plate. This involves controlling the position of the rotary servos attached to the bottom of the plate based on the X-Y position of the ball measured by the overhead camera.

Topics Covered

- Model the dynamics of the ball from first-principles.
- Obtain a transfer function representation of the system.
- Design a proportional-derivative (PD) control that positions the ball to a desired X-Y position on the plate. The control is designed according to given specifications.
- Simulate the control on a single-axis 2 DOF Ball Balancer system.
- Implement the control on the actual Quanser 2 DOF Ball Balancer device using QUARC®.

Prerequisites

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

- Transfer function fundamentals, e.g., obtaining a transfer function from a differential equation.
- Basics of MATLAB® and SIMULINK®.
- Rotary Servo Base Unit Integration Laboratory in the Rotary Servo Base Unit Workbook ([4]) in order to be familiar using QUARC® with the servo.
- Modeling and Position Control Laboratory given in the Rotary Servo Base Unit Workbook ([4]).

2 MODELING

2.1 Background

Since the 2 DOF Ball Balancer uses two Rotary Servo Base Unit (SRV02) devices and the table is symmetrical, it is assumed that the dynamics of each axis is the same. The 2 DOF Ball Balancer is therefore modeled as two de-coupled "ball and beam" systems where we assume the angle of the x-axis servo only affects the ball movement in the x direction. Similarly for the y ball motion. The equation of motion representing the ball's motion along the x axis relative to the plate angle is developed in Section 2.1.1. The servo angle is introduced into the model in Section 2.1.2 and is then represented as a transfer function in Section 2.1.3.

2.1.1 Nonlinear Equation of Motion

The free body diagram of the Ball and Beam is illustrated in Figure 2.1. Using this diagram, the equation of motion, or *EOM* for short, relating the motion of the ball, x, to the angle of the beam, α, can be found. Based on Newton's First Law of Motion, the sum of forces acting on the ball along the beam equals

$$m_b \ddot{x}(t) = \sum F = F_{x,t} - F_{x,r} \tag{2.1}$$

where m_b is the mass of the ball, x is the ball displacement, $F_{x,r}$ is the force from the ball's inertia, and $F_{x,t}$ is the translational force generated by gravity. Friction and viscous damping are neglected.

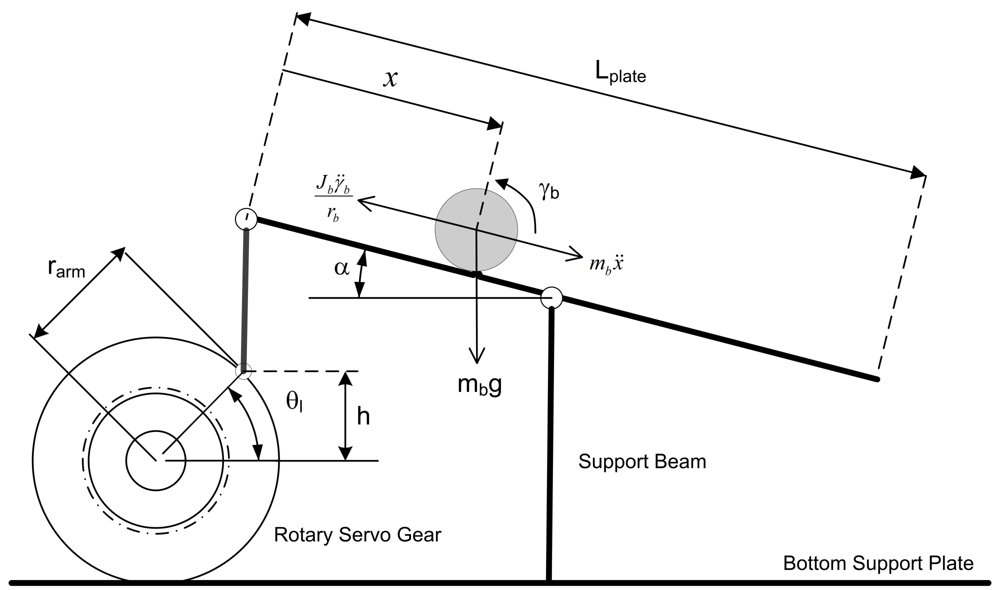


Figure 2.1: Modeling ball on plate in one dimension.

Modeling Conventions:

- Applying a positive voltage causes the servo load gear to move in the positive, counter-clockwise (CCW) direction. This moves the beam upwards and causes the ball to roll in the positive direction (i.e., away from the servo towards the left). Thus $V_m > 0 \rightarrow \dot{\theta}_l > 0 \rightarrow \dot{x} > 0$.
- Ball position is zero, $x = 0$, when located in the center of the beam.
- Servo angle is zero, $\theta_l = 0$, when the beam is parallel to the ground, $\alpha = 0$.

For the ball to be stationary at a certain moment, i.e., be in equilibrium, the force from the ball's momentum must be equal to the force produced by gravity. As illustrated in Figure 2.1, the force $F_{x,t}$ in the x direction (along the beam) that is caused by gravity can be found as:

$$F_{x,t} = m_b\, g\, \sin\alpha(t)$$

The force caused by the rotation of the ball is

$$F_{x,r} = \frac{\tau_b}{r_b}$$

where r_b is the radius of the ball and τ_b is the torque which equals

$$\tau_b = J_b \ddot{\gamma}_b(t)$$

where γ_b is the ball angle. Using the sector formula, $x(t) = \gamma_b(t)\, r_b$, we can convert between linear and angular displacement. Then, the force acting on the ball in the x direction from its momentum becomes:

$$F_{x,r} = \frac{J_b \ddot{x}(t)}{r_b^2}.$$

Now, by substituting the rotational and translational forces into Equation 2.1, we can get the nonlinear equation of motion for the ball and beam as:

$$m_b \ddot{x}(t) = m_b g \sin\alpha(t) - \frac{J_b \ddot{x}(t)}{r_b^2}.$$

Solving for the linear acceleration gives:

$$\ddot{x}(t) = \frac{m_b g r_b^2}{m_b r_b^2 + J_b} \sin\alpha(t). \tag{2.2}$$

2.1.2 Relative to Servo Angle

In this section, the equation of motion representing the position of the ball relative to the angle of the servo load gear is derived. The obtained equation will be nonlinear (includes a trigonometric term). Therefore, it will have to be linearized to use in control design.

Let's look at how we can find the relationship between the servo load gear angle, θ_l, and the beam angle, α. Using the schematic given in Figure 2.1, consider the beam and servo angles required to change the height of the beam by h. Taking the sine of the beam angle gives the expression

$$\sin\alpha(t) = \frac{2h}{L_{plate}}$$

and taking the sine of the servo load shaft angle results in the equation

$$\sin\theta_l(t) = \frac{h}{r_{arm}}.$$

From these we can obtain the following relationship between the beam and servo angle

$$\sin\alpha(t) = \frac{2r_{arm}}{L_{plate}} \sin\theta_l(t). \tag{2.3}$$

To find the equation of motion that represent the ball's motion with respect to the servo angle θ_l we need to linearize the equation of motion about the servo angle $\theta_l(t) = 0$. Insert the servo and plate angle relationship, Equation 2.3, into the nonlinear EOM found in 2.2

$$\ddot{x}(t) = \frac{2 m_b g r_{arm} r_b^2}{L_{plate}(m_b r_b^2 + J_b)} \sin\theta_l(t). \tag{2.4}$$

About angle zero, the sine function can be approximated by $\sin\theta_l(t) \approx \theta_l(t)$. Applying this to the nonlinear EOM gives the *linear* equation of motion of the ball

$$\ddot{x}(t) = \frac{2 m_b g r_{arm} r_b^2}{L_{plate}(m_b r_b^2 + J_b)} \theta_l(t). \tag{2.5}$$

2.1.3 Obtaining the Transfer Function

The complete open-loop system of the 2 DOF Ball Balancer is represented by the block diagram shown in Figure 2.2. The Rotary Servo Base Unit (SRV02) transfer function $P_s(s)$ represents the dynamics between the servo input motor voltage and the resulting load angle. The dynamics between the angle of the servo load gear and the position of the ball is described by transfer function $P_{bb}(s)$. This is a decoupled model, e.g., it is assumed the x-axis servo does not affect the y-axis response.

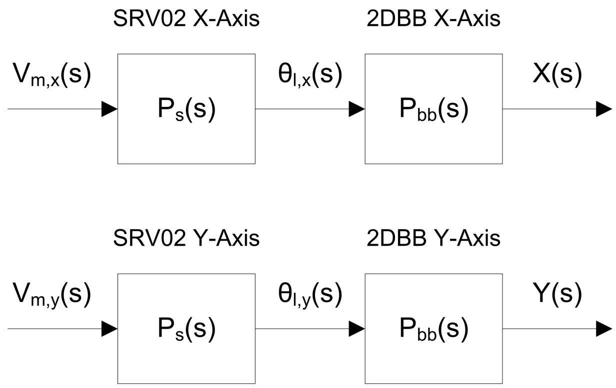


Figure 2.2: 2 DOF Ball Balancer open-loop block diagram

The block diagram for a single-axis of the 2 DOF Ball Balancer, denoted 1DBB, is shown in Figure 2.3.

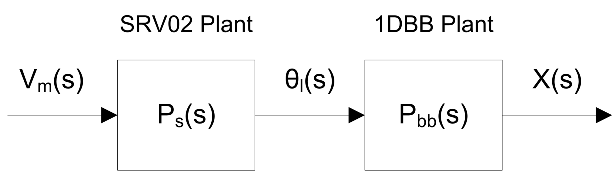


Figure 2.3: 1D open-loop block diagram of 2 DOF Ball Balancer

This section will describe how to obtain the 1DBB transfer function

$$P(s) = P_{bb}(s)P_s(s)$$

where

$$P_{bb}(s) = \frac{X(s)}{\Theta_l(s)}$$

is the *servo angle to ball position* transfer function and

$$P_s(s) = \frac{\Theta_l(s)}{V_m(s)}$$

is the *voltage to servo angle* transfer function.

The SRV02 transfer function models the servo load gear position, $\theta_l(t)$, with respect to the servo input voltage, $V_m(t)$. Recall that in Modeling Laboratory ([4]), this transfer function was found to be:

$$P_s(s) = \frac{K}{s(\tau s + 1)}. \tag{2.6}$$

The nominal model parameters, K and τ, of the SRV02 with no load and in high-gear configuration are:

$$K = 1.53 \text{ rad/(V-s)} \tag{2.7}$$

and

$$\tau = 0.0211 \text{ s} \tag{2.8}$$

Note: These parameters are different than the those found in the Modeling Laboratory because it does not include the inertial disc load.

The servo angle to ball position transfer function, $P_{bb}(s)$, can be found by taking the Laplace transform of the linear equation of motion in 2.5 as:

$$P_{bb}(s) = \frac{X(s)}{\Theta_l(s)} = \frac{K_{bb}}{s^2} \tag{2.9}$$

As illustrated in Figure 2.3, both systems are in series. By inserting the plate position transfer function, 2.9, and the voltage-servo transfer function, 2.6, into Equation 2.10, we can derive the complete process transfer function $P(s)$ as:

$$P(s) = \frac{X(s)}{V_m(s)} = \frac{K_{bb}K}{s^3(\tau s + 1)} \tag{2.10}$$

This is the servo voltage to ball displacement transfer function.

2.2 Pre-Lab Exercises

1. Find the moment of inertia of a *hollow* ball with mass m_b and radius r_b (do not evaluate it numerically). Apply it to the equation of motion in Equation 2.5 and simplify the expression. If the ball doubled in size and mass, how would it affect the equation?

2. Lump the coefficient parameters of $\alpha(t)$ in Equation 2.5 into parameter K_{bb}, the model gain for single-axis 2DBB system. Evaluate the gain given the plate length of 27.5 cm and the arm radius of 2.54 cm (all parameters given in the 2 DOF Ball Balancer User Manual [5]).

3 SENSOR CALIBRATION

3.1 Background

The (x, y) position of the ball is measured using the overhead camera. In the 2 DOF Ball Balancer User Manual ([5]), the camera is adjusted such that the plate fits inside its viewing area. The width and height resolutions of the camera are set to the same value (i.e., square image) and denoted by the variable *res*.

The camera has an image resolution of 720 by 540. The full camera view is shown in Figure 3.1 as *Original image*. The ball coordinate system, $[X_b, Y_b]$, is rotated by -90 deg and the image is resized, i.e., cropped, to fit the plate width and height. This is the plate or *target image*. The ball is measured through an image processing object detection that outputs the row and column of the detected object. The x-axis ball position is measured from the image column, col_b and the y-axis ball position is measured form the image row, row_b.

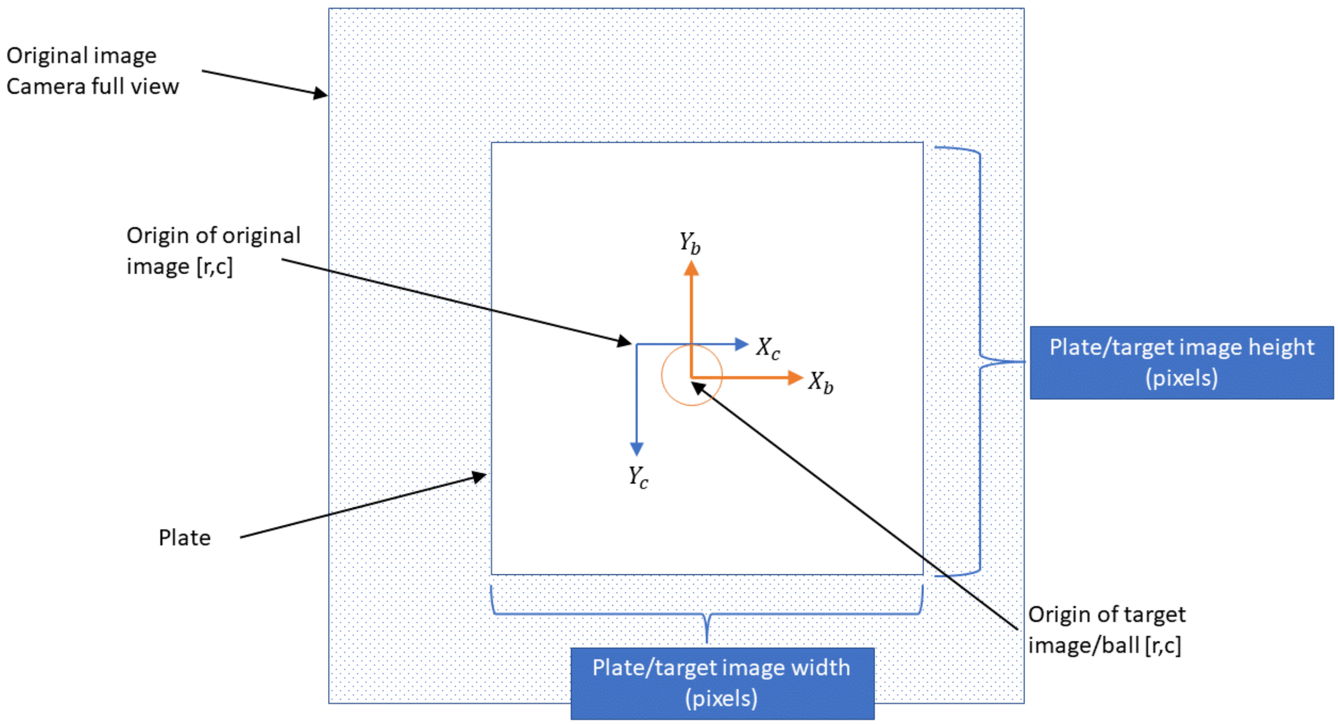


Figure 3.1: Position of ball on plate shown in terms of raw pixel camera measurements

3.2 Pre-Lab Exercises

1. The ball is to be controlled relative to the coordinate axis $[X_b, Y_b]$ in metric units. Find the functions that describe the position of the ball relative to the ball position coordinate system $[X_b, Y_b]$ from the rows and columns outputted by the object detection sofware, row_b and col_b. The plate height and width is the same size and denoted as variable w_{plate} in pixels and L_{tbl} in meters. The rows and columns outputted by the software are with respect to the *original camera based axes*, $[X_c, Y_c]$. Note that the y-axis is the inverse of the ball position axes. The rows, which are used to find the y ball position, increases as you go from the top of the original image to the bottom, e.g., 45 pixels at the top of the plate and 425 pixels at the bottom. The columns used to measure the x position are in-line.

2. In 2 DOF Ball Balancer User Manual ([5]), the image viewed by the camera has to be customized in order to view the entire plate. Assume the plate/target image has a resolution of 440 pixels (i.e., width and height of image is 440). Calculate the position of the ball if the camera reads $row_b = 160$ and $col_b = 220$.

3.3 Lab Experiments

In Section 3.2 we found equations that describe the ball position from the raw camera pixel measurements. In this lab, the software controller is calibrated and tested to ensure the ball is being measured according to the conventions given in Figure 3.1.

Experimental Setup

The q_2dbb_camera_calib Simulink diagram shown in Figure 3.2 is used to configure the *Camera Calibration* subsystem and ensure the correct X-Y ball position is being measured.

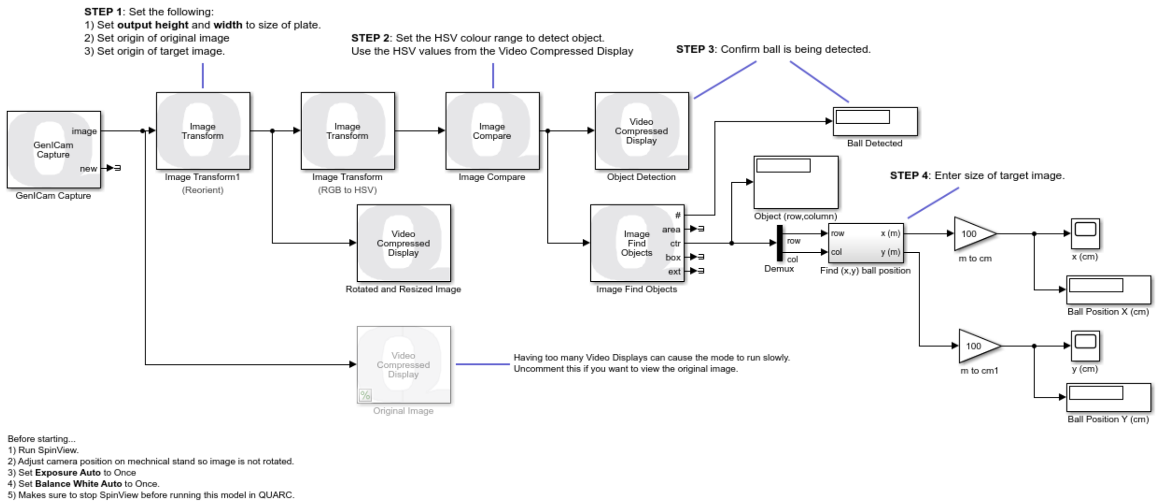
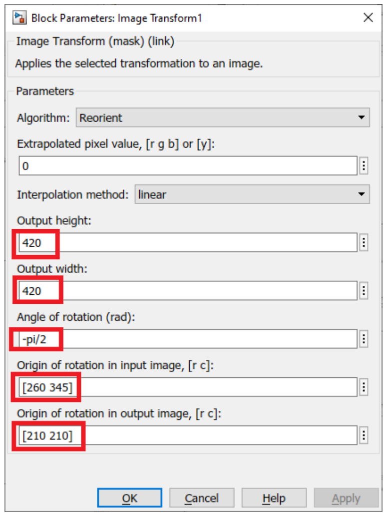


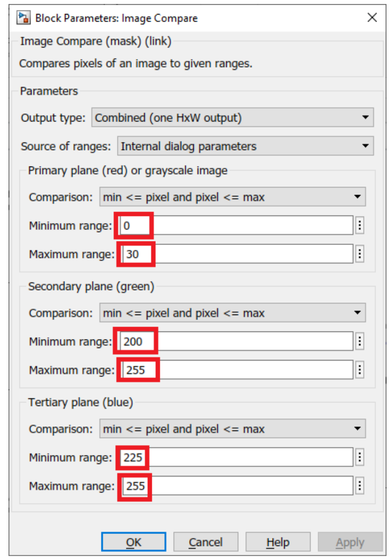
Figure 3.2: Simulink diagram used to calibrate camera using QUARC®

IMPORTANT: Before you can conduct this experiment, you need to make sure that the are configured according to your system setup. **If they have not been configured already, then you need to go to Section 5.2 to configure the lab files first.**

1. Place the ping pong ball in the middle of the table (approximately).
2. Go through the camera calibration procedure outlined in the 2 DOF Ball Balancer User Manual ([5]) using the SpinView software. Make sure you set the **Exposure Auto** setting to *Once* and the **Balance While Auto** to *Once*. This will ensure the image has the correct balance and be able to measure objects more accurately.
3. Close the Spinview software. **Make sure this software is closed to avoid obtaining an error when running the QUARC controller.**
4. Open the **Image Transform** block. As shown in 3.3a, set the **Output height**, **Output width**, **Origin of rotation in input image**, and **Origin of rotation** in output image. The origin of rotation of the original input camera image can be found using the mouse cursor (it only need to be approximate). The origin of rotation of the target image is half the resolution of the resized image, e.g., if you target image size is 420 then the origin for the rotation axes is [210,210].
5. Open the *Image Compare* block, as shown in Figure 3.3b. The HSV threshold values are set for an orange ball but this will have be tuned depending on the lighting and other factors. You can view the HSV values for the ball directly from the Spinview software in the calibration procedure (in the 2 DOF Ball Balancer User Manual). Place the mouse cursor over the ball and record the HSV values showed at the top of Spinview software in the title bar.
6. Click on the *Monitor & Tune* button in the HARDWARE or QUARC tab in the *q_2dbb_camera_calib* Simulink diagram. The QUARC model should now be running.



(a) Typical settings for the *Image Transform* block



(b) Typical *Image Compare* block parameters for an orange ball

Figure 3.3: Settings for the *Image Transform* and *Image Compare* blocks

7. Open the *Rotated and Resized Image* Video Compressed Display block and confirm that you see only the plate in the target image.
8. Click on the **Stop** button.
9. Open the *Find (x,y) ball position* block and enter the size of your target image, set in the *Image Transform* block earlier, in the *Image Size (pixels)* block. In this case, the target image size of the plate is 420 pixels.

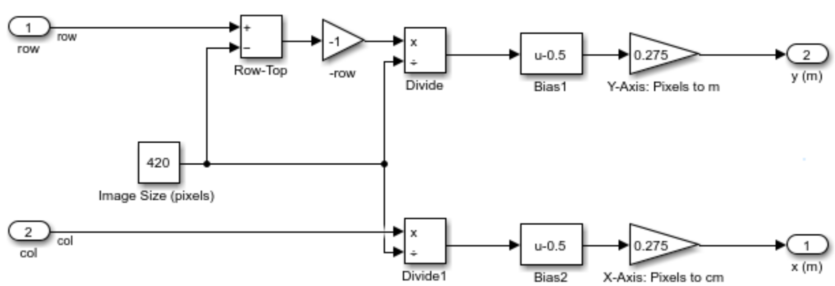
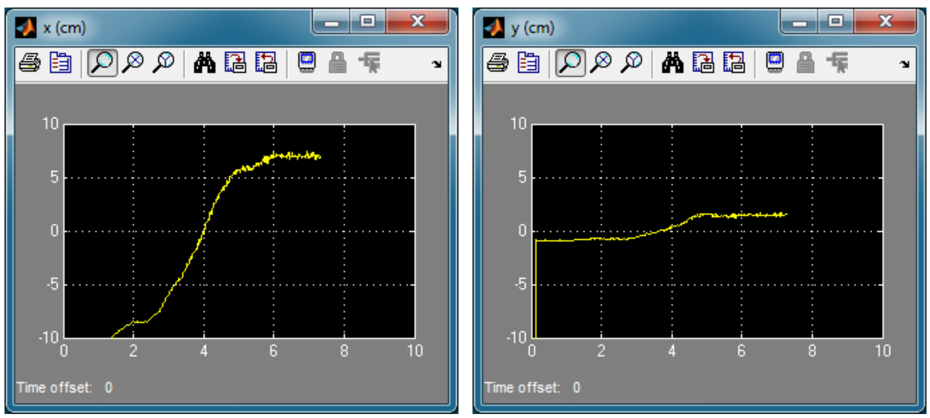


Figure 3.4: Converts measured row and column into x and y ball position

10. Click on the *Monitor & Tune* button in the HARDWARE or QUARC tab again.
11. Move the ball around the plate, the scope and numeric displays should read different positions. The *x (cm)* and *y (cm)* scopes in Figure 3.5 should a typical response when moving the ball along the x-axis in the positive direction.
12. Move the ball along the axes and determine, from the responses, if the measured ball position follows the conventions presented in Section 3.1. Briefly explain the tests you used to validate the ball position measurements.
13. Go to QUARC | Stop to stop running the controller.



(a) X-Axis

(b) Y-Axis

Figure 3.5: Ball position response when moving ball from left to right along x-axis

3.4 Results

Fill out Table 3.1 with your answers from your modeling lab results - both simulation and implementation.

Description	Value	Unit
Calibration		
Image Pos Left		pixels
Image Pos Top		pixels
Image Pos Width		pixels
Image Pos Height		pixels

Table 3.1: Results

4 CONTROL DESIGN

4.1 Specifications

The time-domain requirements for controlling the position of the ball - for both the x and y axes - on the 2 DOF Ball Balancer are:

Specification 1: 4% settling time: $t_s \leq 3.0$ s

Specification 2: Percentage overshoot: $PO \leq 10$ %

Specification 3: Steady-state error: $|e_{ss}| \leq 5$ mm

Given a step reference, the peak position of the ball should not overshoot over 10%. After 3.0 seconds, the ball should settled within 4% of its steady-state value (i.e., not the reference) and the steady-state should be within 5 mm of the desired position.

4.2 Background

The control that will be used for each axis of the 2 DOF Ball Balancer system is illustrated by the block diagram in Figure 4.1. Based on the measured ball position $X(s)$, the outer ball control loop computes the servo load shaft angle $\Theta_d(s)$ to attain the desired ball position $X_d(s)$. The inner loop controls the servo position using a proportional gain $k_{p,s}$.

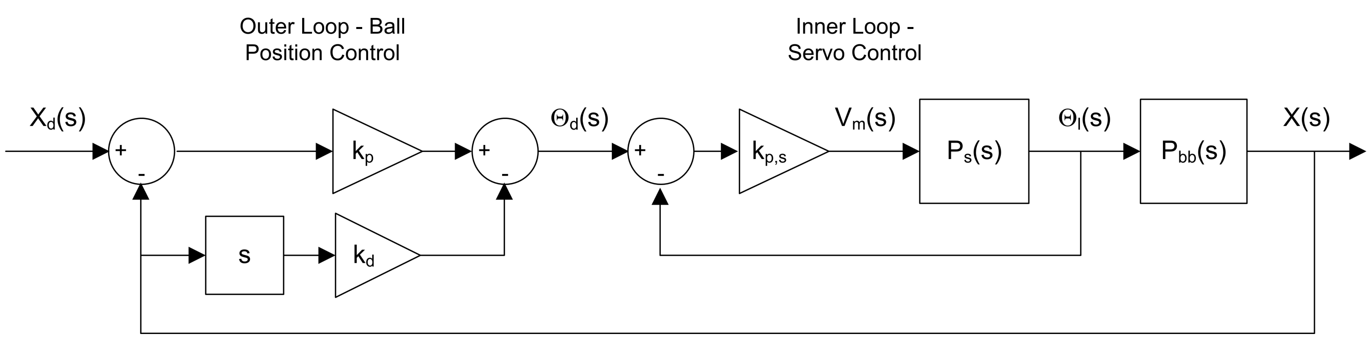


Figure 4.1: Ball position control system for one axis on the 2 DOF Ball Balancer

To design the ball position control, assume that the inner servo loop is ideal and therefore

$$\theta_l(t) = \theta_d(t) \tag{4.1}$$

As shown in the block diagram, the outer-loop is a PV control

$$\Theta_d(s) = k_p\,(X_d(s) - X(s)) - k_d\,s\,X(s). \tag{4.2}$$

Given the assumption that $\Theta_l(t) = \Theta_d(t)$, substitute the PV control 4.2 into the plant transfer function 2.9 and solve for $X(s)/X_d(s)$ to obtain the *2 DOF Ball Balancer closed-loop transfer function*

$$\frac{X(s)}{X_d(s)} = \frac{K_{bb}\,k_p}{s^2 + K_{bb}\,k_d\,s + K_{bb}\,k_p} \tag{4.3}$$

As shown in Figure 4.1, k_p and k_d are the proportional and derivative gains, respectively. We can compute the values for these meet the settling time and overshoot specifications.

The transfer function of a standard second-order system is given by

$$\frac{Y(s)}{R(s)} = \frac{\omega_n^2}{s^2 + 2\,\zeta\,\omega_n\,s + \omega_n^2} \tag{4.4}$$

For a second-order under damped system, the settling time and percent overshoot equations are

$$t_s = -\frac{\ln\left(c_{ts}\sqrt{1-\zeta^2}\right)}{\zeta\,\omega_n} \tag{4.5}$$

and

$$PO = 100 e^{\frac{-\pi\,\zeta}{\sqrt{1-\zeta^2}}} \tag{4.6}$$

where ω_n is the natural frequency ζ is the damping ratio.

4.3 Pre-Lab Questions

1. Find the natural frequency and damping ratio required to match the settling time and overshoot specifications given in Section 4.1.
2. The closed-loop transfer function given in Equation 4.3 is second-order. Find the proportional and derivative gains needed to match the standard second-order transfer function given in Equation 4.4.
3. Using the model gain, K_{bb}, found in Section 2.2, find the control gains needed to satisfy the requirements given in Section 4.1.

4.4 Lab Experiments

The control gains found in Section 4.3 are tested in simulation first. Once it is validated that the gains satisfy the requirements, it is implemented on the actual Quanser 2 DOF Ball Balancer system.

4.4.1 Control Simulation

Using the linear model of the system and the designed control gain, the closed-loop response can be simulated. This way, we can test the controller and see if it satisfies the given specifications before running it on the hardware platform.

Experiment Setup

The *s_2dbb* Simulink diagram shown in Figure 4.2 is used to simulate the closed-loop response of the 2 DOF Ball Balancer using the control developed in Section 4.3.

The *Signal Generator* block generates a 0.08 Hz square wave (with amplitude of 1). The *Amplitude (deg)* gain block is used to change the desired ball position command. The PD gains, k_p and k_d, are set in the *Ball Balancer Control* subsystem and is read from the Matlab workspace. The dynamics of the servo *voltage to angle* and the servo angle to ball position are modeled for a single-axis.

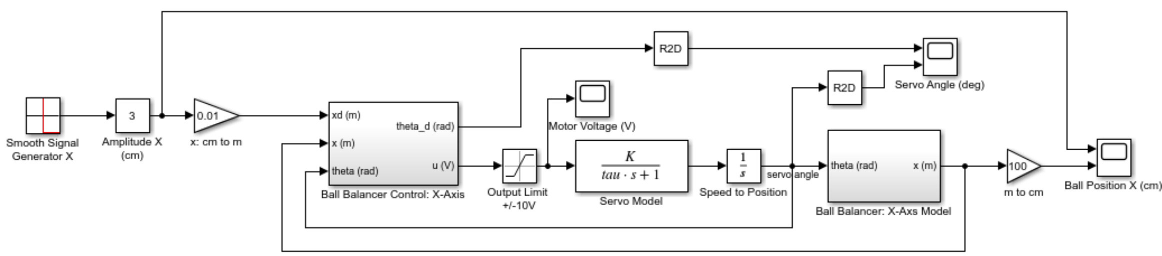
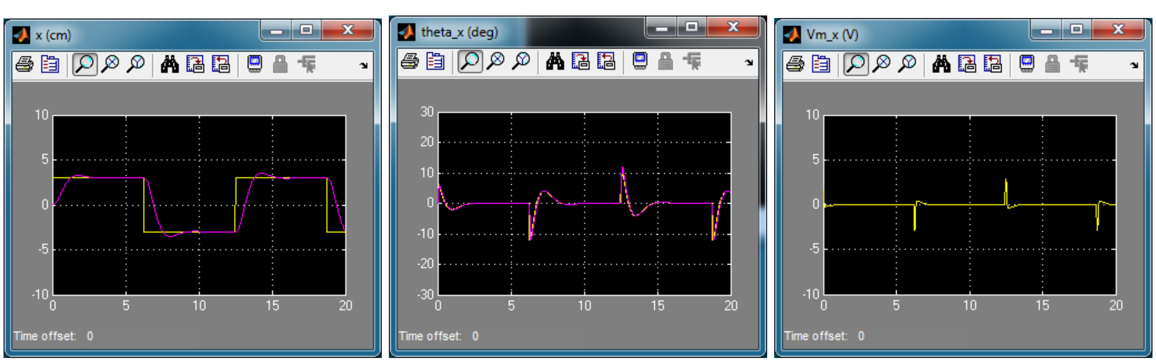


Figure 4.2: Simulink diagram used to simulate the PD response

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 5.3 to configure the lab files first.**

1. Set the *Amplitude X* Slider Gain to 3 cm.
2. Run *s_2dbb* to simulated the closed-loop response with these gains. Figure 4.3 shows a typical response.



(a) Ball Position (b) Servo Angle (c) Voltage

Figure 4.3: Simulated closed-loop response of x-axis

3. Plot the responses from the *x (cm)*, *theta (deg)*, and *Vm (V)* scopes in a Matlab figure.

   Saved response data: When the QUARC controller is stopped, these scopes automatically save the last 20 seconds of their response data to the variables *data_x*, *data_theta_x*, and *data_vm_x*. For *data_x*, the time is in *data_x(:,1)*, the setpoint (i.e., desired ball position) is in *data_x(:,2)*, and the simulated ball position is in *data_x(:,3)*. The same conventions apply for the servo angle data, which is saved in *data_theta_x*. In the *data_vm_x* variable, the (:,1) holds the time vector and the (:,2) holds the input voltage.

4. Measure the settling time, percent overshoot, and steady-state error of the simulated response. Does the response satisfy the specifications given in Section 4.1?

4.4.2 Control Implementation

In this section, the control designed in Section 4.3 and tested in simulation in Section 4.4.1 is implemented on the actual 2 DOF Ball Balancer device. Measurements will then be taken to ensure that the specifications are satisfied.

Experiment Setup

The *q_2dbb* Simulink diagram shown in Figure 4.4 is used to run the state-feedback control on the Quanser 2 DOF Ball Balancer system. The *SRV02-ET+2DBB* subsystem contains QUARC blocks that interface with the servo DC motors, servo encoders, and overhead camera.

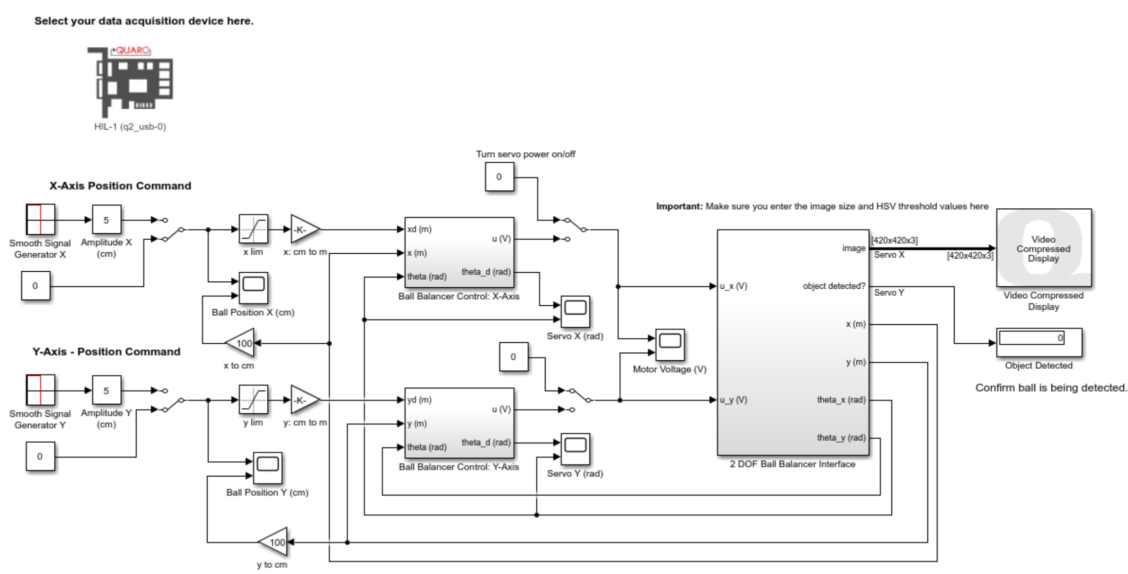
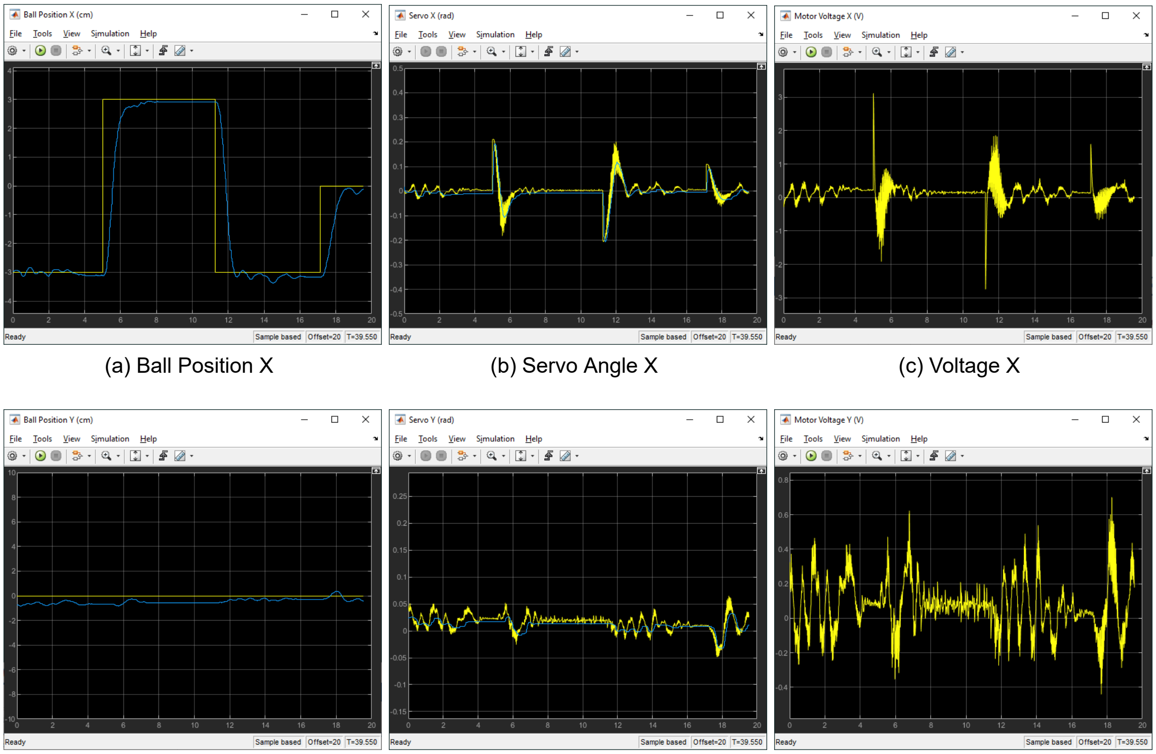


Figure 4.4: Simulink diagram that runs the control on 2 DOF Ball Balancer system

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 5.4 to configure the lab files first.**

1. Place the ball in the middle of the table.
2. Click on the **Monitor & Tune** button in the HARDWARE or QUARC tab of the Simulink model to build and run the QUARC model. Depending where the ball is initially located, the servos may rotate to position the ball in the center (desired ball position initially set to (0,0)).
3. Go to the *Setpoints* subsystem and set the *Amplitude X* Slider Gain to 3 cm. The ball should travel $\pm$3cm along the x-axis of the plate.



(a) Ball Position X (b) Servo Angle X (c) Voltage X

(d) Ball Position Y (e) Servo Angle Y (f) Voltage Y

Figure 4.5: Running PD control on 2 DOF Ball Balancer

4. Figure 4.5 depicts a typical response (both the x and y axes scopes are shown).
5. Stop the controller once you have obtained a representative response.
6. Plot the responses from the *x (cm)*, *theta_x (deg)*, and *Vm_x (V)* scopes in a MATLAB figure. Similarly as described in Section 4.4.1, the response data is saved in variables *data_x*, *data_theta_x*, and *data_vm_x*.
7. Measure the settling time, percent overshoot, and steady-state error of the measured response. Does the response satisfy the specifications given in Section 4.1?

4.5 Results

Fill out Table 4.1 with your answers from your control lab results - both simulation and implementation.

Description	Symbol	Value	Unit
Simulation			
Proportional Gain	k_p		rad/m
Derivative Gain	k_d		rad/m/s
Settling time	t_s		s
Percentage overshoot	PO		%
Steady-state error	e_{ss}		cm
Implementation			
Proportional Gain	k_p		rad/m
Derivative Gain	k_d		rad/m/s
Settling time	t_s		s
Percentage overshoot	PO		%
Steady-state error	e_{ss}		cm

Table 4.1: Results

5 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 [3] 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) card that is compatible with QUARC. This includes Quanser Hardware-in-the-loop (HIL) boards such as:
  - Q2-USB
  - Q8-USB
  - QPID
  - QPIDe

  and some National Instruments DAQ devices (e.g., NI USB-6251, NI PCIe-6259). For a full listing of compliant DAQ cards, see Reference [1].
- Two Quanser Rotary Servo Base Unit (SRV02-ET).
- Quanser 2 DOF Ball Balancer (attaches to SRV02 units).
- Quanser VoltPAQ-X2 power amplifier, or equivalent.

Before Starting Lab

Before you begin this laboratory make sure:

- QUARC® is installed on your PC, as described in [2].
- DAQ device has been successfully tested (e.g., using the test software in the Quick Start Guide or the *QUARC Analog Loopback Demo*).
- 2 DOF Ball Balancer and amplifier are connected to your DAQ board as described Reference [5].

5.1 Overview of Files

File Name	Description
2 DOF Ball Balancer User Manual.pdf	This manual describes the hardware of the Quanser 2 DOF Ball Balancer system and explains how to setup and wire the system for the experiments.
2 DOF Ball Balancer Workbook (Student).pdf	Courseware that contains pre-lab questions and lab experiments demonstrating how to design and implement a ball balancing and positioning controller on the Quanser 2 DOF Ball Balancer plant using QUARC®.
setup_2dbb.m	Main Matlab script that sets the various system and control parameters in Matlab for the 2 DOF Ball Balancer experiments. **Run this file to setup the laboratory.**
config_srv02.m	Returns the configuration-based SRV02 model specifications Rm, kt, km, Kg, eta_g, Beq, Jeq, and eta_m, the sensor calibration constants K_POT, K_ENC, and K_TACH, and the amplifier limits VMAX_AMP and IMAX_AMP.
config_2dbb.m	Returns the parameters associated with the 2DBB system.
calc_conversion_constants.m	Returns various conversions factors.
s_2dbb.mdl	Simulink file that simulates one-axis of the 2DBB system using a PD control.
q_2dbb_camera_calib.mdl	When ran with QUARC®, this Simulink model can be used to test the camera and validate the ball position measurements .
q_2dbb.mdl	Simulink file that implements the full ball positioning controller on the actual 2 DOF Ball Balancer system using QUARC®.

Table 5.1: Files supplied with the Quanser 2 DOF Ball Balancer.

5.2 Setup for Camera Calibration

Before beginning in-lab procedure outlined in Section 3.3, the q_2dbb_camera_calib Simulink diagram must be properly configured.

Follow these steps:

1. Make sure the 2 DOF Ball Balancer hardware (including the camera) has been set up correctly, as documented in the 2 DOF Ball Balancer User Manual.
2. Load MATLAB®.
3. Browse through the *Current Directory* window in Matlab and find the folder that contains the file *q_2dbb_camera_calib.mdl*.
4. Open the *q_2dbb_camera_calib.mdl* Simulink diagram, shown in Figure 3.2.

5.3 Setup for Control Simulation

Before going through the control simulation in Section 4.4.1, the s_2dbb Simulink diagram and the setup_2dbb.m script must be configured.

Follow these steps to configure the lab properly:

1. Load MATLAB®.
2. Browse through the *Current Directory* window in Matlab and find the folder that contains the 2 DOF Ball Balancer files, e.g., s_2dbb.mdl.
3. Open *setup_2dbb.m*.
4. Set CONTROL_TYPE = 'MANUAL'.
5. **Configure setup script**: When used with the 2DBB, the SRV02 has no load (i.e., no disc or bar) and has to be in the high-gear configuration. Make sure the script is setup to match this setup:
   - EXT_GEAR_CONFIG to 'HIGH'
   - LOAD_TYPE to 'NONE'
   - Ensure ENCODER_TYPE, TACH_OPTION, K_CABLE, AMP_TYPE, and VMAX_DAC parameters are set according to the SRV02 system that is to be used in the laboratory.
6. Run *setup_2dbb.m*.
7. Enter the 2 DOF Ball Balancer model gain as variable *K_bb* in Matlab.
8. Enter the proportional and derivative gains you found in Section 4.3 as *kp* and *kd* in Matlab.
9. Open s_2dbb.mdl Simulink diagram shown in Figure 4.2.

5.4 Setup for Control Implementation

Before beginning the in-lab exercises given in Section 4.4.2, the q_2dbb Simulink diagram and the setup_2dbb.m script must be setup.

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

1. Make sure the 2 DOF Ball Balancer hardware (including the camera) has been set up correctly, as documented in the 2 DOF Ball Balancer User Manual.
2. Open the q_2dbb Simulink diagram. Make sure the *Manual Switch* is set to the ON (upward) position.
3. Ensure the Camera Calibration subsystem has been properly set, as performed in Section 3.3.
4. **Configure DAQ:** Configure the HIL Initialize block for the DAQ device that is installed in your system. By default, the block is setup for the Quanser Q2-USB board. See Reference [1] for more information on configuring the HIL Initialize block.
5. **Configure setup script:** Set the parameters in the *setup_2dbb.m* script according to your system setup. See Section 5.3 for more details.
6. Go through the simulation laboratory described in Section 4.4.1 to test the model and control gains and ensure they meet the requirements. Make sure you run setup_2dbb.m and enter the 2 DOF Ball Balancer model and controls gains *K_bb*, *kp*, and *kd*.
7. Turn ON the power amplifier.

REFERENCES

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

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

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

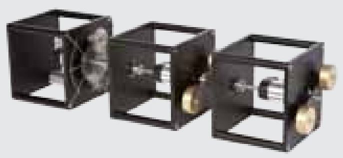
[4] Quanser Inc. SRV02 lab manual. 2011.

[5] Quanser Inc. *2 DOF Ball Balancer User Manual*, 2013.

Over ten rotary experiments for teaching fundamental and advanced controls concepts

Rotary Servo Base Unit

Multi-DOF Torsion



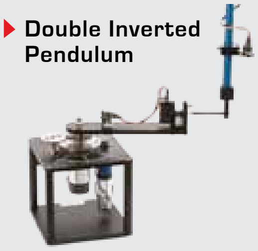
Inverted Pendulum

Flexible Link

Flexible Joint

Ball and Beam

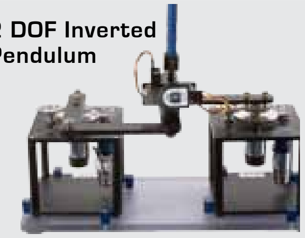
Double Inverted Pendulum



Gyro/Stable Platform

2 DOF Robot

2 DOF Inverted Pendulum



2 DOF Gantry

2 DOF Ball Balancer

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.

QUANSER
INNOVATE. EDUCATE.