STUDENT WORKBOOK

SRV02 Base Unit Experiment For Matlab®/Simulink® Users

Standardized for ABET Evaluation Criteria

Developed by:
Jacob Apkarian, Ph.D., Quanser
Michel Lévis, M.A.Sc., Quanser
Hakan Gurocak, Ph.D., Washington State University

SRV02 educational solutions are powered by:

Course material complies with:

CAPTIVATE. MOTIVATE. GRADUATE.

Acknowledgements

Quanser, Inc. would like to thank the following contributors:
Dr. Hakan Gurocak, Washington State University Vancouver, USA, for his help to include embedded outcomes assessment, and
Dr. K. J. Åström, Lund University, Lund, Sweden for his immense contributions to the curriculum content.

ROTARY SERVO BASE UNIT POSITION CONTROL

The objective of this laboratory is to develop feedback systems that control the position of the rotary servo load shaft. Using the proportional-integral-derivative (PID) family, controllers are designed to meet a set of specifications.

Topics Covered

- Design of a proportional-derivative (PD) controller for position control of the servo load shaft to meet certain time-domain requirements.
- Actuator saturation.
- Design of a proportional-integral-derivative (PID) controller to track a ramp reference signal.
- Simulation of the PI and PID controllers using the developed model of the plant to ensure the specifications are met without any actuator saturation.
- Implementation of the controllers on the Quanser Rotary Servo Base Unit device to evaluate their performance.

Prerequisites

- System has been setup and tested by going through the Rotary Servo Base Unit Quick Start Guide.
- Familiar with Transfer function fundamentals, e.g. obtaining a transfer function from a differential equation.
- Familiar with MATLAB® and SIMULINK® fundamentals.
- Integration laboratory experiment to get familiar with using QUARC® with the Rotary Servo Base Unit.

Caution

Make sure the system has been setup and tested by going through the Rotary Servo Base Unit Quick Start Guide before starting this experiment!

1 Background

1.1 Desired Position Control Response

The block diagram shown in Figure 1.1 is a general unity feedback system with compensator (controller) C(s) and a transfer function representing the plant, P(s). The measured output, Y(s), is supposed to track the reference signal R(s) and the tracking has to match to certain desired specifications.

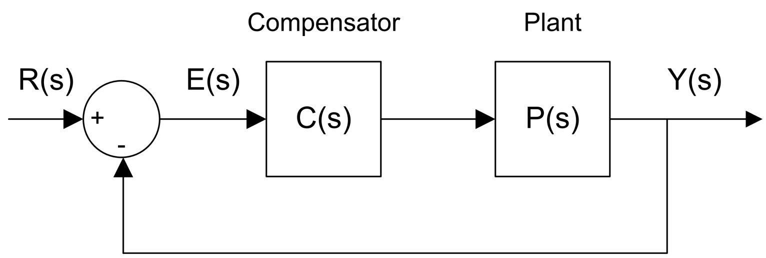


Figure 1.1: Unity feedback system.

The output of this system can be written as:

$$Y(s) = C(s)P(s)\left(R(s) - Y(s)\right) \tag{1.1}$$

By solving for $Y(s)$, we can find the closed-loop transfer function:

$$\frac{Y(s)}{R(s)} = \frac{C(s)P(s)}{1 + C(s)P(s)} \tag{1.2}$$

Recall in Integration laboratory experiment, the Rotary Servo Base Unit voltage-to-speed transfer function was derived. To find the voltage-to-position transfer function, we can put an integrator $(1/s)$ in series with the speed transfer function (effectively integrating the speed output to get position). Then, the resulting open-loop voltage-to-load gear position transfer function becomes:

$$P(s) = \frac{K}{s(\tau s + 1)} \tag{1.3}$$

As you can see from this equation, the plant is a second order system. In fact, when a second order system is placed in series with a proportional compensator in the feedback loop as in Figure 1.1, the resulting closed-loop transfer function can be expressed as:

$$\frac{Y(s)}{R(s)} = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \tag{1.4}$$

where ω_n is the natural frequency and ζ is the damping ratio. This is called the *standard second-order* transfer function. Its response depends on the values of ω_n and ζ.

1.1.1 Peak Time and Overshoot

Consider a second-order system as shown in Equation 1.4 subjected to a step input given by

$$R(s) = \frac{R_0}{s} \tag{1.5}$$

with a step amplitude of $R_0 = 1.5$. The system response to this input is shown in Figure 1.2, where the red trace is the response (output), $y(t)$, and the blue trace is the step input $r(t)$.

The maximum value of the response is denoted by the variable y_{max} and it occurs at a time t_{max}. The initial value of the response is denoted as y_0. For a response similar to Figure 1.2, the percent overshoot is found using

$$PO = \frac{100\left(y_{max} - R_0\right)}{R_0} \tag{1.6}$$

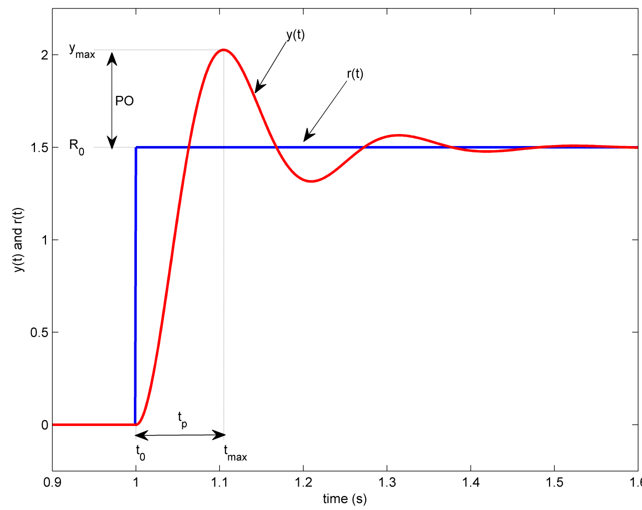


Figure 1.2: Standard second-order step response.

From the initial step time, t_0, the time it takes for the response to reach its maximum value is

$$t_p = t_{max} - t_0 \tag{1.7}$$

This is called the *peak time* of the system.

In a second-order system, the amount of overshoot depends solely on the damping ratio parameter and it can be calculated using the equation

$$PO = 100e^{\left(-\frac{\pi\zeta}{\sqrt{1-\zeta^2}}\right)} \tag{1.8}$$

The peak time depends on both the damping ratio and natural frequency of the system and it can be derived as:

$$t_p = \frac{\pi}{\omega_n\sqrt{1-\zeta^2}} \tag{1.9}$$

Generally speaking, the damping ratio affects the shape of the response while the natural frequency affects the speed of the response.

1.1.2 Steady State Error

Steady-state error is illustrated in the ramp response given in Figure Figure 1.3 and is denoted by the variable e_{ss}. It is the difference between the reference input and output signals after the system response has settled. Thus, for a time t when the system is in steady-state, the steady-state error equals

$$e_{ss} = r_{ss}(t) - y_{ss}(t) \tag{1.10}$$

where $r_{ss}(t)$ is the value of the steady-state input and $y_{ss}(t)$ is the steady-state value of the output.

We can find the error transfer function $E(s)$ in Figure 1.1 in terms of the reference $R(s)$, the plant $P(s)$, and the compensator $C(s)$. The Laplace transform of the error is

$$E(s) = R(s) - Y(s) \tag{1.11}$$

Solving for $Y(s)$ from Equation 1.3 and substituting it in Equation 1.11 yields

$$E(s) = \frac{R(s)}{1 + C(s)P(s)} \tag{1.12}$$

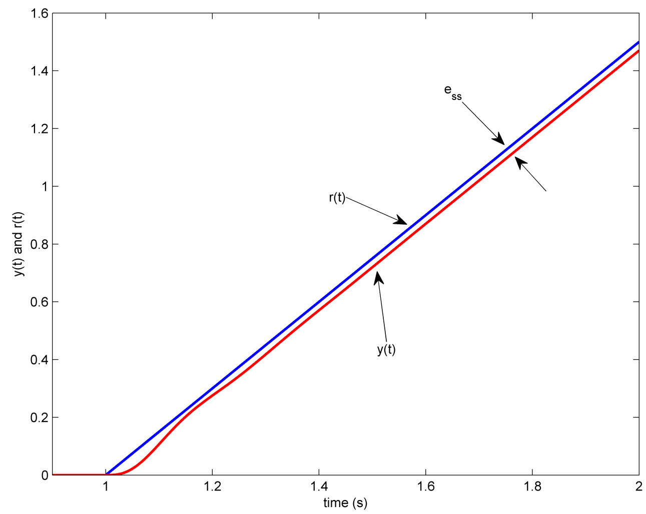


Figure 1.3: Steady-state error in ramp response.

We can find the the steady-state error of this system using the final-value theorem:

$$e_{ss} = \lim_{s \to 0} sE(s) \tag{1.13}$$

In this equation, we need to substitute the transfer function for E(s) from Equation 1.12. The E(s) transfer function requires, R(s), C(s) and P(s). For simplicity, let C(s)=1 as a compensator. The P(s) and R(s) were given by equations Equation 1.3 and Equation 1.5, respectively. Then, the error becomes:

$$E(s) = \frac{R_0}{s\left(1 + \frac{K}{s(\tau s+1)}\right)} \tag{1.14}$$

Applying the final-value theorem gives

$$e_{ss} = R_0 \left(\lim_{s \to 0} \frac{(\tau s + 1)\, s}{\tau s^2 + s + K} \right) \tag{1.15}$$

When evaluated, the resulting steady-state error due to a step response is

$$e_{ss} = 0 \tag{1.16}$$

Based on this zero steady-state error for a step input, we can conclude that the Rotary Servo Base Unit is a *Type 1* system.

1.1.3 Time-Domain Control Specifications

The desired time-domain specifications for controlling the position of the Rotary Servo Base Unit load shaft are:

$$e_{ss} = 0 \tag{1.17}$$

$$t_p = 0.20 \text{ s} \tag{1.18}$$

and

$$PO = 5.0 \text{ \%} \tag{1.19}$$

Thus, when tracking the load shaft reference, the transient response should have a peak time less than or equal to 0.20 seconds, an overshoot less than or equal to 5 %, and the steady-state response should have no error.

1.2 PD Controller Design

1.2.1 Closed Loop Transfer Function

The proportional-derivative (PD) compensator to control the position of the Rotary Servo Base Unit has the following structure

$$V_m(t) = k_p\left(\theta_d(t) - \theta_l(t)\right) - k_d\left(\frac{d}{dt}\theta_l(t)\right) \tag{1.20}$$

where k_p is the proportional control gain, k_d is the derivative control gain, $\theta_d(t)$ is the setpoint or reference load shaft angle, $\theta_l(t)$ is the measured load shaft angle, and $V_m(t)$ is the Rotary Servo Base Unit motor input voltage. The block diagram of the PD control is given in Figure 1.4.

Note: This is a variation of the classic PD control where the D-term is in the feedback path as opposed to in the forward path.

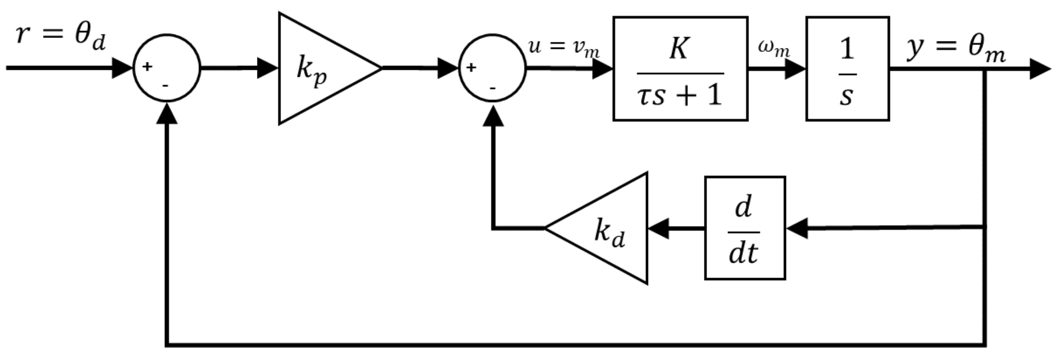


Figure 1.4: Block diagram of Rotary Servo Base Unit PD position control.

We need to find the closed-loop transfer function $\Theta_l(s)/\Theta_d(s)$ for the closed-loop position control of the Rotary Servo Base Unit. Taking the Laplace transform of Equation 1.20 gives

$$V_m(s) = k_p\left(\Theta_d(s) - \Theta_l(s)\right) - k_d s\Theta_l(s) \tag{1.21}$$

From the Plant block in Figure 1.4 and Equation 1.3, we can write

$$\frac{\Theta_l(s)}{V_m(s)} = \frac{K}{s(\tau s + 1)} \tag{1.22}$$

Substituting Equation 1.21 into Equation 1.22 and solving for $\Theta_l(s)/\Theta_d(s)$ gives the Rotary Servo Base Unit position closed-loop transfer function as:

$$\frac{\Theta_l(s)}{\Theta_d(s)} = \frac{Kk_p}{\tau s^2 + (1 + Kk_d)s + Kk_p} \tag{1.23}$$

1.2.2 Controller Gain Limits

In control design, a factor to be considered is saturation. This is a nonlinear element and is represented by a saturation block as shown in Figure 1.5. In a system like the Rotary Servo Base Unit, the computer calculates a numeric control voltage value. This value is then converted into a voltage, $V_{dac}(t)$, by the digital-to-analog converter of the data-acquisition device in the computer. The voltage is then amplified by a power amplifier by a factor of K_a. If the amplified voltage, $V_{amp}(t)$, is greater than the maximum output voltage of the amplifier or the input voltage limits of the motor (whichever is smaller), then it is saturated (limited) at V_{max}. Therefore, the input voltage $V_m(t)$ is the effective voltage being applied to the Rotary Servo Base Unit motor.

The limitations of the actuator must be taken into account when designing a controller. For instance, the voltage entering the Rotary Servo Base Unit motor should never exceed

$$V_{max} = 10.0 \text{ V} \tag{1.24}$$

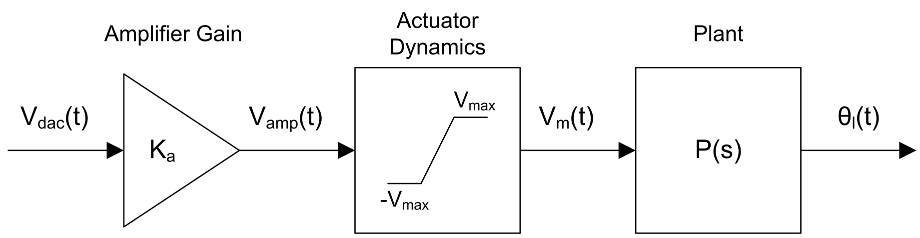


Figure 1.5: Actuator saturation.

1.2.3 Ramp Steady State Error Using PD Control

From our previous steady-state analysis, we found that the closed-loop Rotary Servo Base Unit system is a Type 1 system. In this section, we will investigate the steady-state error due to a *ramp* input when using PD controller.

Given the following ramp setpoint (input)

$$R(s) = \frac{R_0}{s^2} \tag{1.25}$$

we can find the error transfer function by substituting the Rotary Servo Base Unit closed-loop transfer function in Equation 1.23 into the formula given in Equation 1.11. Using the variables of the Rotary Servo Base Unit, this formula can be rewritten as $E(s) = \Theta_d(s) - \Theta_l(s)$. After rearranging the terms we find:

$$E(s) = \frac{\Theta_d(s)s\,(\tau s + 1 + Kk_d)}{\tau s^2 + s + Kk_p + Kk_d s} \tag{1.26}$$

Substituting the input ramp transfer function Equation 1.25 into the $\Theta_d(s)$ variable gives

$$E(s) = \frac{R_0\,(\tau s + 1 + Kk_d)}{s\,(\tau s^2 + s + Kk_p + Kk_d s)} \tag{1.27}$$

1.3 PID Controller Design

Adding integral control can help eliminate steady-state error. The proportional-integral-derivative (PID) algorithm to control the position of the Rotary Servo Base Unit is shown in Figure 1.6. The motor voltage will be generated by the PID according to:

$$V_m(t) = k_p\,(\theta_d(d) - \theta_l(t)) + k_i \int (\theta_d(t) - \theta_l(t))dt - k_d\left(\frac{d}{dt}\theta_l(t)\right) \tag{1.28}$$

where k_p is the proportional control gain, k_i is the integral gain, k_d is the derivative control gain, $\theta_d(t)$ is the setpoint or reference load shaft angle, $\theta_l(t)$ is the measured load shaft angle, and $V_m(t)$ is the Rotary Servo Base Unit motor input voltage.

Note: This is a variation of the standard PID control with the D-term in the feedback path as opposed to the feed forward path.

We need to find the closed-loop transfer function $\Theta_l(s)/\Theta_d(s)$ for the closed-loop position control of the Rotary Servo Base Unit. Taking the Laplace transform of Equation 1.28 gives

$$V_m(s) = \left(k_p + \frac{k_i}{s}\right)(\Theta_d(s) - \Theta_l(s)) - k_d s\Theta_l(s) \tag{1.29}$$

From the Plant block in Figure Figure 1.6 and Equation 1.3, we can write

$$\frac{\Theta_l(s)}{V_m(s)} = \frac{K}{(\tau s + 1)\,s} \tag{1.30}$$

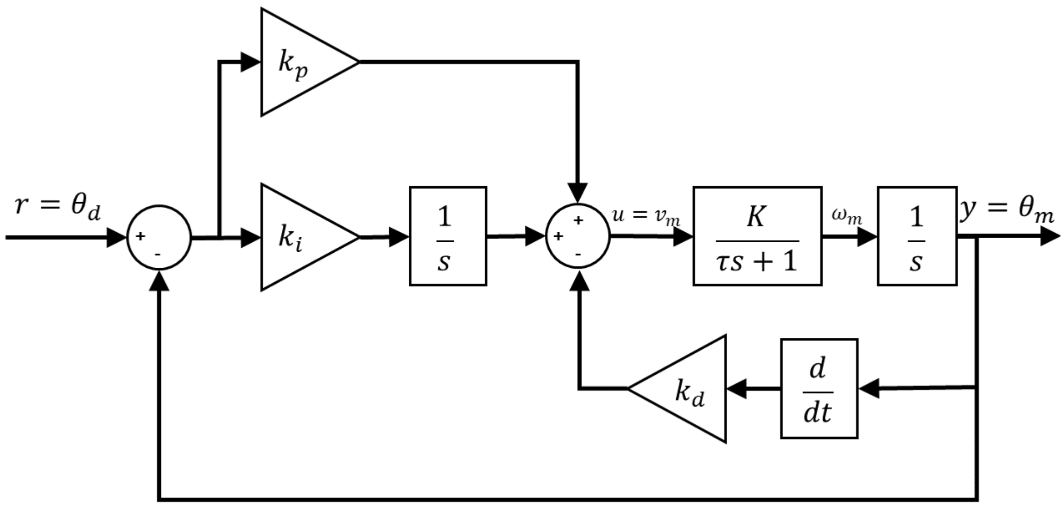


Figure 1.6: Block diagram of PID Rotary Servo Base Unit position control.

Substituting Equation 1.29 into Equation 1.30 and solving for $\Theta_l(s)/\Theta_d(s)$ gives the Rotary Servo Base Unit position closed-loop transfer function as:

$$\frac{\Theta_l(s)}{\Theta_d(s)} = \frac{K\left(k_p s + k_i\right)}{s^3\tau + (1 + Kk_d)s^2 + Kk_p s + Kk_i} \tag{1.31}$$

1.3.1 Ramp Steady-State Error using PID Controller

To find the steady-state error of the Rotary Servo Base Unit for a ramp input under the control of the PID substitute the closed-loop transfer function from Equation 1.31 into Equation 1.11

$$E(s) = \frac{\Theta_d(s)s^2\left(\tau s + 1 + Kk_d\right)}{s^3\tau + s^2 + Kk_p s + Kk_i + Kk_d s^2} \tag{1.32}$$

Then, substituting the reference ramp transfer function Equation 1.25 into the $\Theta_d(s)$ variable gives

$$E(s) = \frac{R_0\left(\tau s + 1 + Kk_d\right)}{s^3\tau + s^2 + Kk_p s + Kk_i + Kk_d s^2} \tag{1.33}$$

1.3.2 Integral Gain Design

It takes a certain amount of time for the output response to track the ramp reference with zero steady-state error. This is called the *settling time* and it is determined by the value used for the integral gain.

In steady-state, the ramp response error is constant. Therefore, to design an integral gain the velocity compensation (the V signal) can be neglected. Thus, we have a PI controller left as:

$$V_m(t) = k_p\left(\theta_d(t) - \theta_l(t)\right) + k_i \int \left(\theta_d(t) - \theta_l(t)\right) dt \tag{1.34}$$

When in steady-state, the expression can be simplified to

$$V_m(t) = k_p e_{ss} + k_i \int_0^{t_i} e_{ss} dt \tag{1.35}$$

where the variable t_i is the integration time.

2 Pre-Lab Questions

Before you start the lab experiments given in Section Section 3, you should study the background materials provided in Section Section 1 and work through the questions in this Section.

1. Calculate the maximum overshoot of the response (in radians) given a step setpoint of 45 degrees and the overshoot specification given in Section 1.1.3.
   Hint: By substituting $y_{max} = \theta(t_p)$ and step setpoint $R_0 = \theta_d(t)$ into Equation 1.6, we can obtain $\theta(t_p) = \theta_d(t)\left(1 + \frac{PO}{100}\right)$. Recall that the desired response specifications include 5% overshoot.
2. The Rotary Servo Base Unit closed-loop transfer function was derived in Equation 1.23 in Section 1.2.1. Find the control gains k_p and k_d in terms of ω_n and ζ. **Hint:** Remember the standard second order system equation.
3. Calculate the minimum damping ratio and natural frequency required to meet the specifications given in Section 1.1.3.
4. Based on the nominal Rotary Servo Base Unit model parameters, K and τ, found in Modeling laboratory experiment calculate the control gains needed to satisfy the time-domain response requirements given in Section 1.1.3.
5. In the PD controlled system, for a reference step of $\pi/4$ (i.e. 45 degree step) starting from $\Theta_l(t) = 0$ position, calculate the *maximum* proportional gain that would lead to providing the maximum voltage to the motor. Ignore the derivative control ($k_d = 0$). Can the desired specifications be obtained based on this maximum available gain and what you calculated in Question 4?
6. For the PD controlled closed-loop system, find the steady-state error and evaluate it numerically given a ramp with a slope of $R_0 = 3.36$ rad/s. Use the control gains found in Question 4.
7. What should be the integral gain k_i so that when the Rotary Servo Base Unit is supplied with the maximum voltage of $V_{max} = 10V$ it can eliminate the steady-state error calculated in Question 6 in 1 second? **Hint:** Start from Equation 1.35 and use $t_i = 1$, $V_m(t) = 10$, the k_p you found in Question 4 and e_{ss} found in Question 6. Remember that e_{ss} is constant.

3 Lab Experiments

The main goal of this laboratory is to explore position control of the Rotary Servo Base Unit load shaft using PD and PID controllers.

In this laboratory, you will conduct the following experiments:

1. Step response with PD controller,
2. Ramp response with PD controller, and
3. Ramp response with PID controller.

In each experiment, you will first simulate the closed-loop response of the system. Then, you will implement the controller using the Rotary Servo Base Unit hardware and software to compare the real response to the simulated one.

3.1 Step Response Using PD Controller

3.1.1 Simulation

First, you will simulate the closed-loop response of the Rotary Servo Base Unit with a PD controller to step input. Our goals are to confirm that the desired response specifications in an ideal situation are satisfied and to verify that the motor is not saturated. Then, you will explore the effect of using a high-pass filter, instead of a direct derivative, to create the velocity signal in the controller.

Experimental Setup

The s_servo_pos_cntrl SIMULINK® diagram shown in Figure 3.1 will be used to simulate the closed-loop position control response with the PD and PID controllers. The Rotary Servo Base Unit Model uses a *Transfer Fcn* block from the SIMULINK® library. The PID Control subsystem contains the PID controller detailed in Section 1.3. When the integral gain is set to zero, it essentially becomes a PD controller.

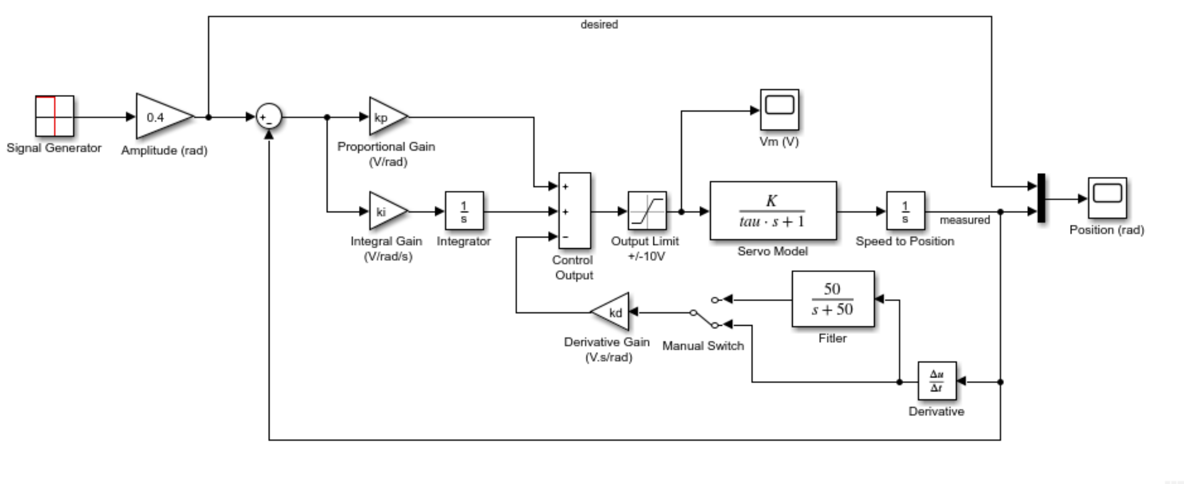
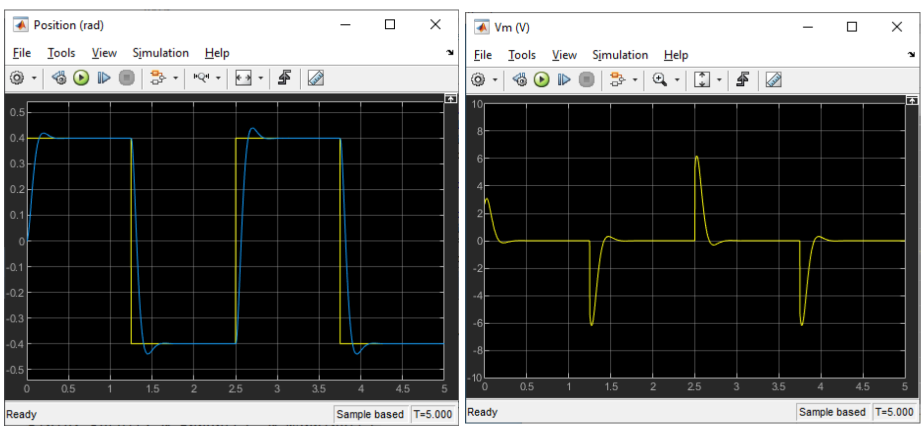


Figure 3.1: Simulink model used to simulate the Rotary Servo Base Unit closed-loop position response.

IMPORTANT: Before you can conduct these experiments, you need to make sure that the lab files are configured according to your Rotary Servo Base Unit setup. If they have not been configured already, then you need to go to Section 4.2 to configure the lab files first.

Closed-loop Response with the PD Controller

1. Enter the proportional and derivative control gains found in Pre-Lab Question 4 in MATLAB® as `kp` and `kd`.
2. Set the integral gain to 0, denoted as `ki` in MATLAB®.
3. To generate a step reference, ensure the *Signal Generator* is set to the following:
   - Signal type = *square*
   - Amplitude = 1
   - Frequency = 0.4 Hz
4. In the SIMULINK® diagram, set the Amplitude (rad) gain block to $0.4(rad)$ to generate a step with an amplitude of 45.8 degrees (i.e. square wave goes between ± 0.4 which results in a step amplitude of 0.8 rad) .
5. Set the Manual Switch such that the velocity of the motor load shaft is fed back directly.
6. Open the servo position Position (rad) scope and the motor input voltage Vm (V) scope.
7. Start the simulation. By default, the simulation runs for 5 seconds. The scopes should be displaying responses similar to Figure 3.2a and Figure 3.2b Note that in the *theta_l* (rad) scope, the yellow trace is the setpoint position while the purple trace is the simulated position (generated by the *Rotary Servo Base Unit Model* block). This simulation is called the *Ideal PD* response as it uses the PD compensator with the derivative block.



(a) Ideal PD position response.

(b) Ideal PD motor input voltage.

Figure 3.2: Simulated PD control response using direct derivative

8. Generate a MATLAB® figure showing the *Ideal PD* position response and the ideal input voltage. After each simulation run, each scope automatically saves their response to a variable in the MATLAB® workspace. That is, the *Position (rad)* scope saves its response to the variable called data_pos and the *Vm (V)* scope saves its data to the *data_vm* variable. The *data_pos* variable has the following structure: *data_pos(:,1)* is the time vector, *data_pos(:,2)* is the setpoint, and *data_pos(:,3)* is the simulated angle. For the *data_vm* variable, *data_vm(:,1)* is the time and *data_vm(:,2)* is the simulated input voltage.
9. Measure the steady-state error, the percent overshoot and the peak time of the simulated response. Does the response satisfy the specifications given in Section 1.1.3? **Hint:** Use the *Cursor Measurements* tool in the Simulink scopes.

Using a Filtered Derivative

When implementing a controller on actual hardware, it is generally not advised to take the direct derivative of a measured signal. Any noise or spikes in the signal becomes amplified and gets multiplied by a gain and fed into the motor which may lead to damage. To remove any high-frequency noise components in the velocity signal, a low-pass filter is placed in series with the derivative, i.e. taking the high-pass filter of the measured signal. However, as with a controller, the filter must also be tuned properly. In addition, the filter has some adverse affects.

1. Set the Manual Switch block to the down position to enable the derivative and filter.
2. Start the simulation. The response in the scopes should still be similar to Figure 3.2a and Figure 3.2b. This simulation is called the *Filtered PD* response as it uses the PD controller with the high-pass filter block.
3. Generate a MATLAB® figure showing the *Filtered PD* position and input voltage responses.
4. Measure the steady-state error, peak time, and percent overshoot. Are the specifications still satisfied without saturating the actuator? Recall that the peak time and percent overshoot should not exceed the values given in Section 1.1.3. Discuss the changes from the ideal response. **Hint:** The different in the response is minor. Make sure you use *Cursor Measurements* tool in the Simulink scope to take precise measurements.

3.1.2 Implementation

In this experiment, we will control the angular position of the Rotary Servo Base Unit load shaft, i.e. the disc load, using the PD controller. Measurements will then be taken to ensure that the specifications are satisfied.

Experimental Setup

The q_servo_pos_cntrl SIMULINK® diagram shown in Figure 3.3 is used to implement the position control experiments. The Rotary Servo Interface - Position subsystem contains QUARC blocks that interface with the DC motor and sensors of the Rotary Servo Base Unit system, as discussed in Integration laboratory experiment. The *PID Control* subsystem implements the PID controller detailed in Section 1.3, except a low-pass filter is placed in series with the derivative to remove the noise.

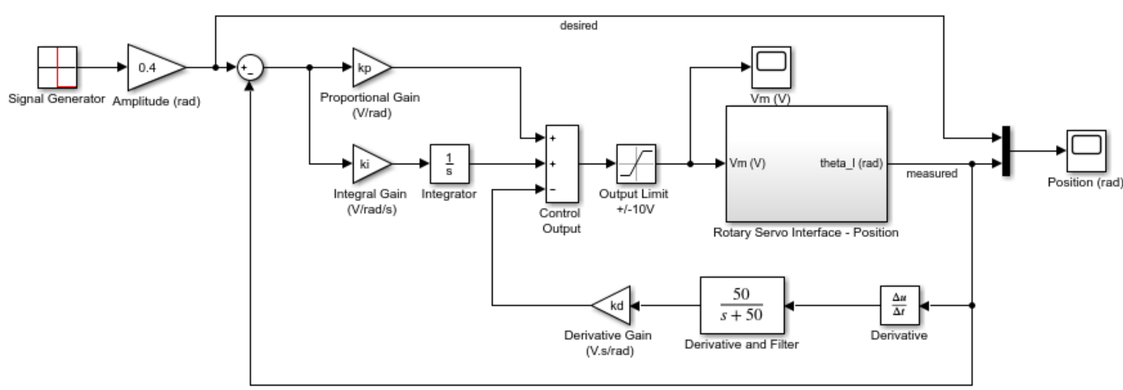


Figure 3.3: Simulink/QUARC model used with to run the PID position controller on the Rotary Servo Base Unit.

IMPORTANT: Before you can conduct these experiments, you need to make sure that the lab files are configured according to your Rotary Servo Base Unit setup. If they have not been configured already, then you need to go to Section 4.3 to configure the lab files first.

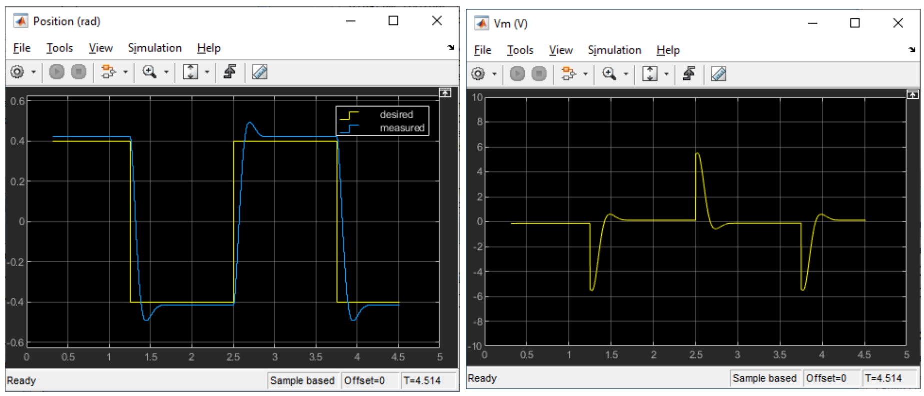
1. Run the setup_servo_pos_cntrl.m script.
2. Enter the proportional and derivative control gains found in Pre-Lab Question 4.
3. Set *Signal Type* in the Signal Generator block to *square* to generate a step reference.

4. Set the *Amplitude (rad)* gain block to 0.4 to generate a step with an amplitude of 45.8 degrees.

5. Open the load shaft position scope, *Position (rad)*, and the motor input voltage scope, *Vm (V)*.

6. Click on QUARC | Build to compile the SIMULINK® diagram.

7. Select QUARC | Start to begin running the controller. The scopes should display responses similar to Figure 3.4a and Figure 3.4b.



(a) Load gear position

(b) Input voltage.

Figure 3.4: PD control response on Rotary Servo Base Unit

8. 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. Generate a MATLAB® figure showing the PD position response and its input voltage.

   As in the s_servo_pos_cntrl Simulink diagram, when the controller is stopped each scope automatically saves their response to a variable in the MATLAB® workspace. Thus the *theta_l (rad)* scope saves its response to the *data_pos* variable and the *Vm (V)* scope saves its data to the *data_vm* variable.

9. Measure the steady-state error, the percent overshoot, and the peak time of the Rotary Servo Base Unit load gear. Does the response satisfy the specifications given in Section 1.1.3?

10. Click the Stop button on the SIMULINK® diagram toolbar (or select QUARC | Stop from the menu) to stop the experiment.

11. Turn off the power to the amplifier if no more experiments will be performed on the Rotary Servo Base Unit in this session.

3.2 Ramp Response Using PD Controller

3.2.1 Simulation

In this simulation, the goal is to verify that the system with the PD controller can meet the zero steady-state error specification without saturating the motor.

As in the Step Response experiment in Section 3.1, in this experiment you need to use the s_servo_pos_cntrl SIMULINK® diagram shown in Figure 3.1 in Section 3.1.1 again.

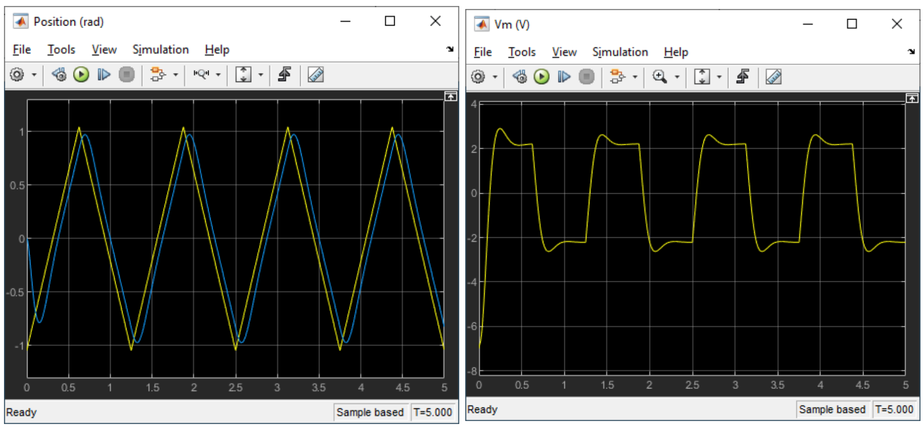
1. Enter the proportional and derivative control gains found in Pre-Lab Question 4and set the integral gain to 0.

2. Set the *Signal Generator* parameters to the following to generate a triangular reference (which corresponds to a ramp input):

    - Signal Type = triangle

- Amplitude = 1
- Frequency = 0.8 Hz

3. Setting the frequency to 0.8 Hz will generate an increasing and decreasing ramp signal with the same slope used in the Pre-Lab Question 6. The slope is calculated from the *Triangular Waveform* amplitude, *Amp*, and frequency, *f*, using the expression.
$$R_0 = 4Ampf \tag{3.1}$$
4. In the SIMULINK® diagram, set the Amplitude (rad) gain block to $\pi/3$.
5. Inside the *PID Control* subsystem, set the Manual Switch to the down position so that the *High-Pass Filter* block is used.
6. Open the load shaft position scope, *Position (rad)*, and the motor input voltage scope, *Vm (V)*.
7. Start the simulation. The scopes should display responses similar to Figure 3.5a and Figure 3.5b.



(a) Load gear position (b) Input voltage

Figure 3.5: Simulated ramp response using PD control

8. Generate a MATLAB® figure showing the *Ramp PD* position response and its corresponding input voltage trace.
9. Measure the steady-state error. Compare the simulation measurement with the steady-state error calculated in Pre-Lab Question 6.

3.2.2 Implementing

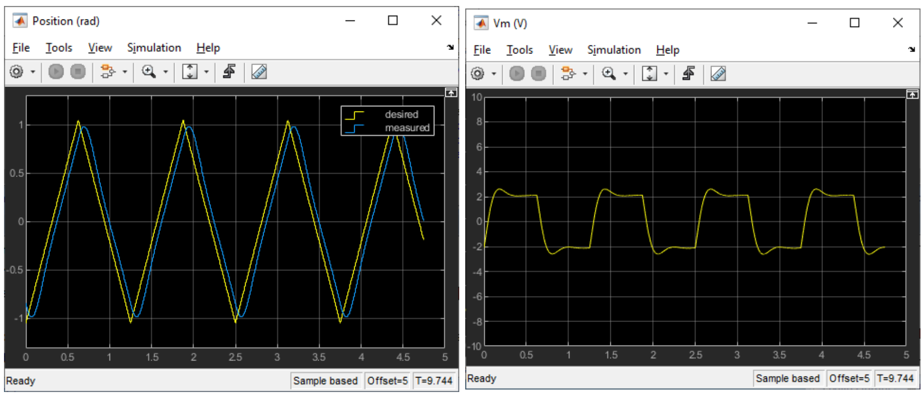
In this experiment, we will control the angular position of the Rotary Servo Base Unit load shaft, i.e. the disc load, using a PD controller. The goal is to examine how well the system can track a triangular (ramp) position input. Measurements will then be taken to ensure that the specifications are satisfied.

As in the Step Response experiment in Section 3.1, in this experiment you also need to use the q_servo_pos_cntrl SIMULINK® diagram shown in Figure 3.3 to implement the position control experiments.

1. Run the setup_servo_pos_cntrl.m script.
2. Enter the proportional and derivative control gains found in Pre-Lab Question 4.
3. Set the *Signal Generator* parameters to the following to generate a triangular reference (i.e. ramp reference):
   - Signal Type = triangle
   - Amplitude = 1

- Frequency = 0.8 Hz

4. In the SIMULINK® diagram, set the *Amplitude (rad)* gain block to $\pi/3$.
5. Open the load shaft position scope, *Position (rad)*, and the motor input voltage scope, *Vm (V)*.
6. Click on QUARC | Build to compile the SIMULINK® diagram.
7. Select QUARC | Start to run the controller. The scopes should display responses similar to Figure 3.6a and Figure 3.6b.



(a) Load gear position

(b) Input voltage

Figure 3.6: Ramp response using PD control on Rotary Servo Base Unit

8. Generate a MATLAB® figure showing the *Ramp PD* position response and its corresponding input voltage trace.
9. Measure the steady-state error and compare it with the steady-state error calculated in Pre-Lab Question 6.

3.3 Ramp Response with No Steady-State Error

Design an experiment to see if the steady-state error can be eliminated when tracking a ramp input. First simulate the response, then implement it using the Rotary Servo Base Unit system.

1. How can the PD controller be modified to eliminate the steady-state error in the ramp response? State your hypothesis and describe the anticipated cause-and-effect leading to the expected result. **Hint:** Look through Section 1.
2. List the independent and dependent variables of your proposed controller. Explain their relationship.
3. Your proposed control, like the PD compensator, are model-based controllers. This means that the control gains generated are based on mathematical representation of the system. Given this, list the assumptions you are making in this control design. State the reasons for your assumptions.
4. Give a brief, general overview of the steps involved in your experimental procedure for two cases: (1) Simulation, and (2) Implementation.
5. For each case, generate a MATLAB® figure showing the position response of the system and its corresponding input voltage.
6. In each case, measure the steady-state error.
7. For each case comment on whether the steady-state specification given in Section 1.1.3 was satisfied without saturating the actuator.

8. Click the Stop button on the SIMULINK® diagram toolbar (or select QUARC | Stop from the menu) to stop the experiment.

9. Turn off the power to the amplifier if no more experiments will be performed on the Rotary Servo Base Unit in this session.

3.4 Results

Fill out Table 3.1 below with your answers to the Pre-Lab questions and your results from the lab experiments.

Section / Question	Description	Symbol	Value	Unit
Question 4	**Pre-Lab: Model Parameters**			
	Open-Loop Steady-State Gain	K		rad/(V.s)
	Open-Loop Time Constant	τ		s
Question 4	**Pre-Lab: PD Gain Design**			
	Proportional gain	k_p		V/rad
	Derivative gain	k_d		V.s/rad
Question 5	**Pre-Lab: Control Gain Limits**			
	Maximum proportional gain	$k_{p,max}$		V/rad
Question 6	**Pre-Lab: Ramp Steady-State Error**			
	Steady-state error using PD	e_{ss}		rad
Question 7	**Pre-Lab: Integral Gain Design**			
	Integral gain	k_i		V/(rad.s)
Section 3.1.1	**Step Response Simulation using Ideal PD**			
	Peak time	t_p		s
	Percent overshoot	PO		%
	Steady-state error	e_{ss}		rad
Section 3.1.1	**Step Response Simulation Using Filtered PD**			
	Peak time	t_p		s
	Percent overshoot	PO		%
	Steady-state error	e_{ss}		rad
Section 3.1.2	**Step Response Implementation**			
	Peak time	t_p		s
	Percent overshoot	PO		%
	Steady-state error	e_{ss}		rad
Section 3.2.1	**Ramp Response Simulation with PD**			
	Steady-state error	e_{ss}		rad
Section 3.2.2	**PD Experimental Ramp Response**			
	Steady-state error	e_{ss}		rad
Section 3.3	**PID Simulated Ramp Response**			
	Steady-state error	e_{ss}		rad
Section 3.3	**PID Experimental Ramp Response**			
	Steady-state error	e_{ss}		rad

Table 3.1: Summary of results for the Rotary Servo Base Unit Position Control laboratory.

4 File Description and Configuration

4.1 Overview of Files

File Name	Description
Rotary Servo Base Unit Position Control - Student Manual.pdf	This laboratory guide contains pre-lab questions and lab experiments demonstrating how to design and implement a position controller on the Quanser Rotary Servo Base Unit rotary plant using QUARC.
setup_servo_pos_cntrl.m	The main MATLAB® script that sets the Rotary Servo Base Unit motor and sensor parameters as well as its configuration-dependent model parameters. **Run this file only to setup the laboratory.**
config_servo.m	Returns the configuration-based Rotary Servo Base Unit model specifications *Rm, kt, km, Kg, eta_g, Beq, Jeq,* and *eta_m,* the sensor calibration constants K_POT and K_ENC, and the amplifier limits VMAX_AMP and IMAX_AMP.
d_model_param.m	Calculates the Rotary Servo Base Unit model parameters *K* and *tau* based on the device specifications *Rm, kt, km, Kg, eta_g, Beq, Jeq,* and *eta_m.*
calc_conversion_constants.m	Returns various conversions factors.
s_servo_pos_cntrl	Simulink file that simulates a closed-loop PID controller for the Rotary Servo Base Unit system.
q_servo_pos_cntrl	Simulink file that implements a closed-loop PID position controller on the Rotary Servo Base Unit system using QUARC®.

Table 4.1: Files supplied with the Rotary Servo Base Unit Position Control laboratory.

4.2 Setup for Position Control Simulations

Follow these steps to configure the lab properly:

1. Load the MATLAB® software.
2. Browse through the Current Directory window in MATLAB® and find the folder that contains the Rotary Servo Base Unit position controller files, e.g. s_servo_pos_cntrl.
3. Double-click on the s_servo_pos_cntrl file to open the SIMULINK® diagram shown in Figure 3.1.
4. Double-click on the setup_servo_pos_cntrl.m file to open the setup script for the position control Simulink models.
5. **Configure setup script:** The controllers will be run on an Rotary Servo Base Unit in the high-gear configuration with the disc load. In order to simulate the Rotary Servo Base Unit properly, make sure the script is setup to match this configuration.
6. Set the CONTROL_TYPE to 'MANUAL' to calculate the control gains yourself.
7. Run the script. The messages shown below, should be generated in the MATLAB® Command Window. The model parameters and specifications are loaded **but the PID gains are all set to zero - they need to be changed**.

```
Servo model parameters:
   K = 1.53 rad/s/V
   tau = 0.0217 s
Specifications:
   tp = 0.2 s
   PO = 5 %
Calculated PD control gains:
   kp = 0 V/rad
   kd = 0 V.s/rad
Integral control gain for triangle tracking:
   ki = 0 V/rad/s
```

4.3 Setup for Position Control Implementation

Before beginning the lab experiments on the Rotary Servo Base Unit device, the q_servo_pos_cntrl SIMULINK® diagram and the setup_servo_pos_cntrl.m script must be configured.

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

1. Setup the Rotary Servo Base Unit in the high-gear configuration and with the disc load as described in Rotary Servo Base Unit User Manual.
2. Load the MATLAB® software.
3. Browse through the *Current Directory* window in MATLAB® and find the folder that contains the Rotary Servo Base Unit position control files, e.g. q_servo_pos_cntrl.
4. Double-click on the q_servo_pos_cntrl file to open the Position Control SIMULINK® diagram shown in Figure 3.1.
5. **Configure DAQ:** Double-click on the HIL Initialize block in the Rotary Servo Interface subsystem (which is located inside the Rotary Servo Interface - Position subsystem) and ensure it is configured for the DAQ device that is installed in your system. See QUARC® documentation for more information on configuring the HIL Initialize block.
6. Configure setup script: Set the parameters in the setup_servo_pos_cntrl.m script according to your system setup. See Section 4.2 for more details.

5 Lab Report

This laboratory contains three experiments, namely,

1. step response,
2. ramp response with PD controller, and
3. ramp response with no steady-state error.

When you are writing your lab report, follow the outline corresponding to the experiment you conducted to build the *content* of your report. Also, in Section 5.4 you can find some basic tips for the *format* of your report.

5.1 Template for Content (Step Response Experiment)

I. PROCEDURE

1. *Closed-loop response with the PD controller*
   - Briefly describe the main goal of the simulation.
   - Briefly describe the simulation procedure (Section 3.1.1)
2. *Step response with PD controller using high-pass filter*
   - Briefly describe the main goal of this simulation.
   - Briefly describe the simulation procedure (Section 3.1.1)
3. *Implementing Step Response using PD Controller*
   - Briefly describe the main goal of this experiment.
   - Briefly describe the experimental procedure (Section 3.1.2)

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

1. Response plot from step 8 in Section 3.1.1, *Simulated step response*
2. Response plot from step 3 in Section 3.1.1, *Filtered PD response*
3. Response plot from step 8 in Section 3.1.2, *Step response of implemented PD controller*
4. Provide applicable data collected in this laboratory (from Table 3.1).

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

1. Step 9 in Section 3.1.1, *Step response with PD controller*
2. Step 4 in Section 3.1.1, *Step response with PD controller using high-pass filter*
3. Step 9 in Section 3.1.2, *Step response with PD controller*

IV. CONCLUSIONS
Interpret your results to arrive at logical conclusions for the following:

1. Step 9 in Section 3.1.1, *Step response simulation with PD controller*
2. Step 4 in Section 3.1.1, *Step response simulation with PD controller using High-pass filter*
3. Step 9 in Section 3.1.2, *Step response with the implemented PD controller*

5.2 Template for Content (Ramp Response with PD)

I. PROCEDURE

1. *Ramp response with PD controller*
   - Briefly describe the main goal of this simulation.
   - Briefly describe the procedure (Section 3.2.1)
2. *Implementing Ramp Response Using PD*
   - Briefly describe the main goal of this experiment.
   - Briefly describe the experimental procedure (Section 3.2.2)

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

1. Response plot from step 8 in Section 3.2.1, *Simulated PD controller with ramp input*
2. Response plot from step 8 in Section 3.2.2, *Ramp response of implemented PD controller*
3. Provide applicable data collected in this laboratory (from Table 3.1).

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

1. Step 9 in Section 3.2.1, *Simulated PD controller with ramp input*
2. Step 9 in Section 3.2.2, *Ramp response of implemented PD controller*

IV. CONCLUSIONS
Interpret your results to arrive at logical conclusions for the following:

1. Step 9 in Section 3.2.2, *Ramp response with the implemented PD controller*

5.3 Template for Content (Ramp Response with No Steady-State Error)

I. PROCEDURE

1. State the hypothesis of your experiment and describe the anticipated cause-and-effect leading to the expected result (Step 1 in Section 3.3).

2. List the independent and dependent variables for the controller. Explain their relationship (Step 2 in Section 3.3).
3. List the assumptions you made in this experiment. State the reasons for your assumptions (Step 3 in Section 3.3).
4. Briefly list the steps of your experimental procedure for two cases: (1) Simulation, and (2) Implementation (Step 4 in Section 3.3).

II. RESULTS

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

1. Response plot from step 5 in Section 3.3, *Simulated controller with ramp input*
2. Response plot from step 5 in Section 3.3, *Implemented controller with ramp input*
3. Provide applicable data collected in this laboratory (from Table 3.1).

III. ANALYSIS

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

1. Step 6 in Section 3.3, *Simulated controller with ramp input*
2. Step 6 in Section 3.3, *Implemented controller with ramp input*

IV. CONCLUSIONS

Interpret your results to arrive at logical conclusions for the following:

1. Step 7 in Section 3.3, *for both simulated and implemented controllers*

5.4 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.

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

Waste Electrical and Electronic Equipment (WEEE)

This symbol indicates that waste products must be disposed of separately from municipal household waste, according to Directive 2002/96/EC of the European Parliament and the Council on waste electrical and electronic equipment (WEEE). All products at the end of their life cycle must be sent to a WEEE collection and recycling center. Proper WEEE disposal reduces the environmental impact and the risk to human health due to potentially hazardous substances used in such equipment. Your cooperation in proper WEEE disposal will contribute to the effective usage of natural resources. For information about the available collection and recycling scheme in a particular country, go to ni.com/citizenship/weee.

电子信息产品污染控制管理办法 （中国 RoHS）

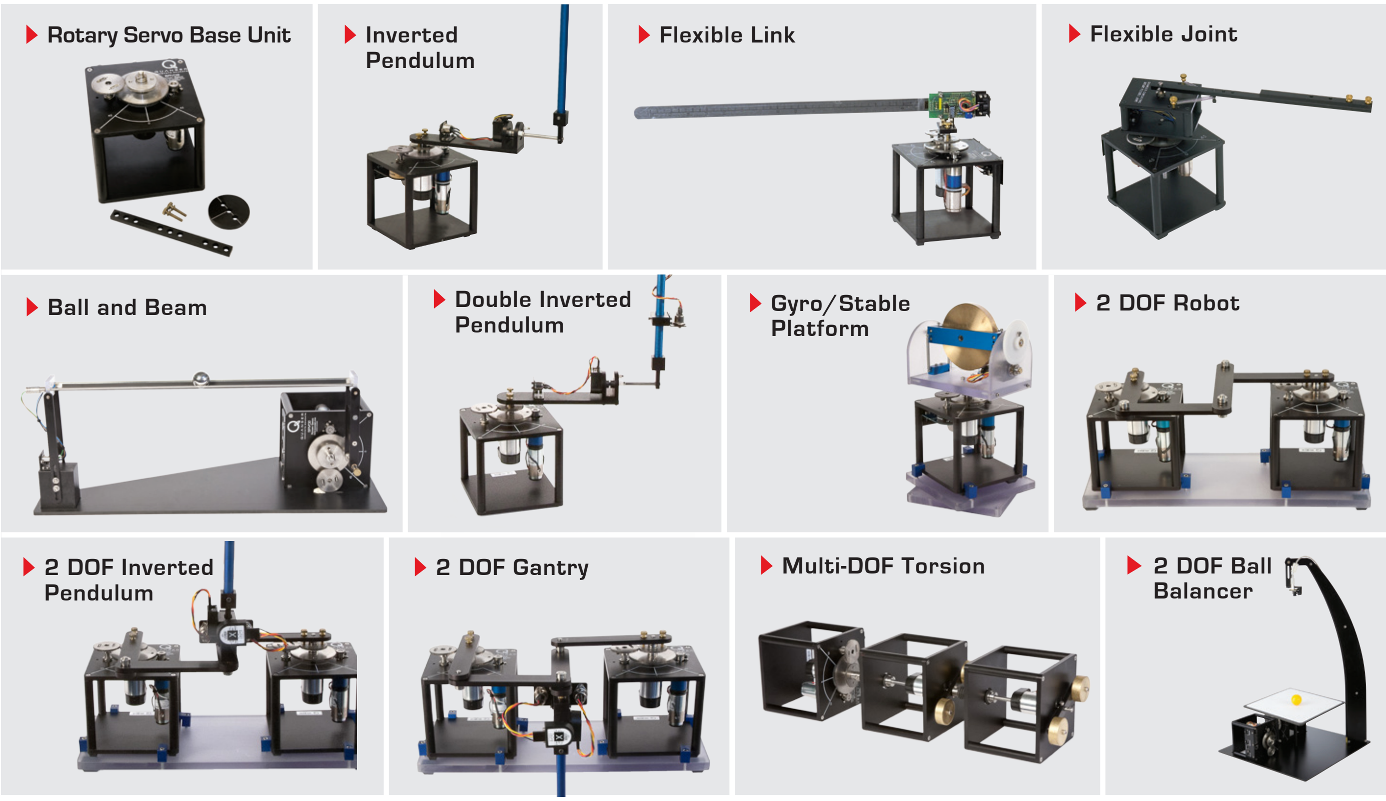
中国客户 National Instruments 符合中国电子信息产品中限制使用某些有害物质命令 （RoHS）。
关于National Instruments 中国 RoHS合规性信息，请登录 ni.com/environment/rohs_china
(For information about China RoHS compliance, go to ni.com/environment/rohs_china)

CE Compliance CE

This product meets the essential requirements of applicable European Directives as follows:

- 2006/95/EC; Low-Voltage Directive (safety)
- 2004/108/EC; Electromagnetic Compatibility Directive (EMC)

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.

QUANSER
INNOVATE · EDUCATE

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

Solutions for teaching and research. Made in Canada.