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.

ROTARY SERVO BASE UNIT SPEED CONTROL

The objective of this laboratory is to develop feedback systems that control the speed of the rotary servo load shaft. A proportional-integral (PI) controller and a lead compensator are designed to regulate the shaft speed according to a set of specifications.

Topics Covered

- Design of a proportional-integral (PI) controller that regulates the angular speed of the servo load shaft.
- Design of a lead compensator.
- Simulation of the PI and lead controllers using the plant model to ensure the specifications are met without any actuator saturation.
- Implemention 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 Response

1.1.1 Speed Control Specifications

The time-domain requirements for controlling the speed of the Rotary Servo Base Unit load shaft are:

$$e_{ss} = 0 \tag{1.1}$$

$$t_p \leq 0.1 \text{ s, and} \tag{1.2}$$

$$PO \leq 5 \text{ \%} \tag{1.3}$$

Thus, when tracking the load shaft reference, the transient response should have a peak time less than or equal to 0.1 seconds, an overshoot less than or equal to 5 %, and zero steady-state error.

In addition to the above time-based specifications, the following frequency-domain requirements are to be met when designing the *Lead Compensator*:

$$PM \geq 72.5 \text{ deg} \tag{1.4}$$

and

$$\omega_g = 25.0 \text{ rad/s} \tag{1.5}$$

The phase margin mainly affects the shape of the response. Having a higher phase margin implies that the system is more stable and the corresponding time response will have less overshoot. The overshoot will not go beyond 5% with a phase margin of at least 72.5 degrees.

The crossover frequency is the frequency where the gain of the Bode plot is 1 (or 0 dB). This parameter mainly affects the speed of the response, thus having a larger ω_g decreases the peak time. With a crossover frequency of 25.0 radians the resulting peak time will be less than or equal to 0.1 seconds.

1.1.2 Peak Time and Overshoot

Consider a second-order system step response when the following reference signal is applied

$$R(s) = \text{Amp}\frac{1}{s} + \text{Off}$$

with a step amplitude of Amp $= 1.5$ and an offset of Off $= 0.5$. The system response to this input is shown in Figure 1.1, 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 1.1, the percent overshoot is found using

$$PO = \frac{100\,(y_{max} - r_{max})}{r_{max} - r_0} \tag{1.6}$$

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.

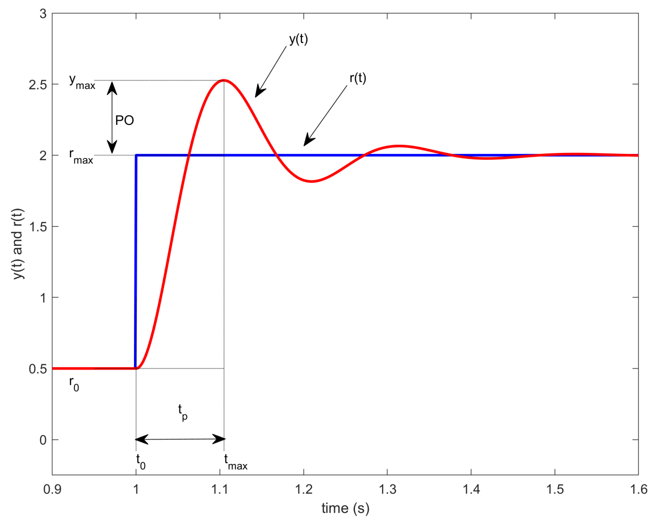


Figure 1.1: Standard second-order step response.

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.

In this laboratory we will use a step setpoint (input):

$$\omega_d(t) = \begin{cases} 3.5 \text{ rad/s} & t \leq t_0 \\ 6.5 \text{ rad/s} & t > t_0 \end{cases} \tag{1.10}$$

where t_0 is the time the step is applied. Initially, the Rotary Servo Base Unit should be running at 3.5 rad/s and after the step time it should jump up to 6.5 rad/s. From the standard definition of overshoot in step response, we can calculate the maximum overshoot of the response (in radians):

$$\omega(t_p) = \omega_d(t_0) + (\omega_d(t) - \omega_d(t_0))\left(1 + \frac{PO}{100}\right) \tag{1.11}$$

with the given values the maximum overshoot of the response is

$$\omega(t_p) = 3.5 + 3.0(1 + 0.05) = 6.65 \text{ rad/s} \tag{1.12}$$

The closed-loop speed response should therefore not exceed the value given in Equation 1.12.

1.1.3 Steady State Error

Consider the speed control system with unity feedback shown in Figure 1.2. Let the compensator be $C(s) = 1$.

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

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

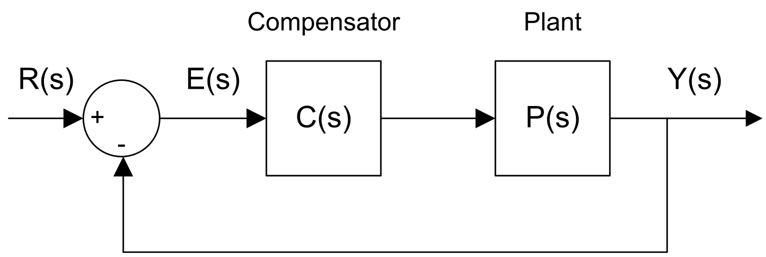


Figure 1.2: Unity feedback loop.

where

$$E(s) = \frac{R(s)}{1 + C(s)P(s)} \tag{1.14}$$

The voltage-to-speed transfer function for the Rotary Servo Base Unit, found in Modeling laboratory experiment, as:

$$P(s) = \frac{K}{\tau s + 1} \tag{1.15}$$

Substituting $R(s) = \frac{R_0}{s}$ and $C(s) = 1$ gives:

$$E(s) = \frac{R_0}{s\left(1 + \frac{K}{\tau s+1}\right)} \tag{1.16}$$

Applying the final-value theorem to the system gives

$$e_{ss} = R_0 \left(\lim_{s \to 0} \frac{\tau s + 1}{\tau s + 1 + K}\right) \tag{1.17}$$

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

$$e_{ss} = \frac{R_0}{1 + K} \tag{1.18}$$

1.2 PI Control Design

1.2.1 Closed Loop Transfer Function

The proportional-integral (PI) compensator used to control the velocity of the Rotary Servo Base Unit has the following structure:

$$V_m(t) = k_p\left(b_{sp}\omega_d(t) - \omega_l(t)\right) - k_i \int (\omega_d(t) - \omega_l(t))dt \tag{1.19}$$

where k_p is the proportional control gain, k_i is the integral control gain, $\omega_d(t)$ is the setpoint or reference angular speed for the load shaft, $\omega_l(t)$ is the measured load shaft angular speed, b_{sp} is the setpoint weight, and $V_m(t)$ is the voltage applied to the Rotary Servo Base Unit motor. The block diagram of the PI control is given in Figure 1.3.

We can take Laplace transform of the controller given in Equation 1.19:

$$V_m(s) = k_p\left(b_{sp}\Omega_d(s) - \Omega_l(s)\right) + \frac{k_i\left(\Omega_d(s) - \Omega_l(s)\right)}{s} \tag{1.20}$$

To find the closed-loop speed transfer function, $\Omega_l(s)/\Omega_d(s)$, we can use the process transfer function from Equation 1.15 and solve for $\Omega_l(s)/\Omega_d(s)$ as:

$$\frac{\Omega_l(s)}{\Omega_d(s)} = \frac{K\left(k_p s b_{sp} + k_i\right)}{s^2\tau + (1 + Kk_p)\,s + Kk_i} \tag{1.21}$$

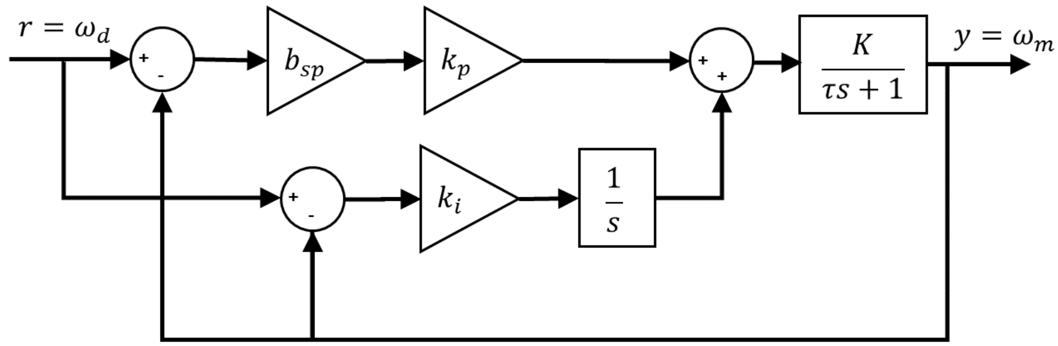


Figure 1.3: Block diagram of Rotary Servo Base Unit PI speed control.

1.2.2 Finding PI Gains to Satisfy Specifications

In this section, we will first calculate the minimum damping ratio and natural frequency required to meet the specifications given in Section 1.1.1. Then, using these values we will calculate the necessary control gains k_p and k_i to achieve the desired performance with a PI controller.

The minimum damping ratio and natural frequency needed to satisfy a given percent overshoot and peak time are:

$$\zeta = -\ln\left(\frac{PO}{100}\right)\sqrt{\frac{1}{\ln\left(\frac{PO}{100}\right)^2 + \pi^2}} \tag{1.22}$$

and

$$\omega_n = \frac{\pi}{t_p\sqrt{1-\zeta^2}} \tag{1.23}$$

Substituting the percent overshoot specifications given in Equation 1.3 into Equation 1.22 gives the required damping ratio

$$\zeta = 0.690 \tag{1.24}$$

Then, by substituting this damping ratio and the desired peak time, given in Equation 1.2, into Equation 1.23, the minimum natural frequency is found as:

$$\omega_n = 43.4 \text{ rad/s} \tag{1.25}$$

Now, let's look at how we can calculate the gains. When the setpoint weight is zero, i.e. $b_{sp} = 0$, the closed-loop Rotary Servo Base Unit speed transfer function has the structure of a *standard second-order system*. We can find expressions for the control gains k_p and k_i by equating the characteristic equation (denominator) of the Rotary Servo Base Unit closed-loop transfer function to the *standard characteristic equation*: $s^2 + 2\zeta\omega_n s + \omega_n^2$.

The denominator of the transfer function can be re-structured into the following:

$$s^2 + \frac{(1+Kk_p)\,s}{\tau} + \frac{Kk_i}{\tau} \tag{1.26}$$

equating the coefficients of this equation to the coefficients of the standard characteristic equation gives:

$$\frac{Kk_i}{\tau} = \omega_n^2 \tag{1.27}$$

and

$$\frac{1+Kk_p}{\tau} = 2\zeta\omega_n \tag{1.28}$$

Then, the proportional gain k_p can be found as:

$$k_p = \frac{-1+2\zeta\omega_n\tau}{K} \tag{1.29}$$

and the integral gain k_i is

$$k_i = \frac{\omega_n^2\tau}{K} \tag{1.30}$$

1.3 Lead Control Design

Alternatively, a lead or lag compensator can be designed to control the speed of the servo. The lag compensator is actually an approximation of a PI control and this, at first, may seem like the more viable option. However, due to the saturation limits of the actuator the lag compensator cannot achieve the desired zero steady-state error specification. Instead, a lead compensator with an integrator, as shown in Figure 1.4, will be designed.

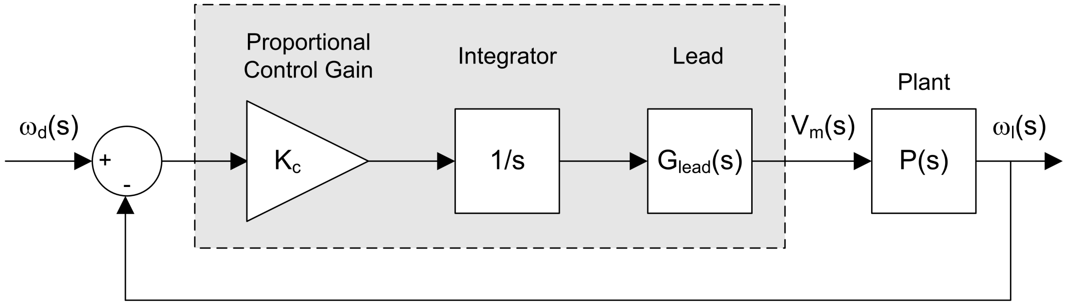


Figure 1.4: Closed-loop Rotary Servo Base Unit speed control with lead compensator.

To obtain zero steady-state error, an integrator is placed in series with the plant. This system is denoted by the transfer function

$$P_i(s) = \frac{P(s)}{s} \tag{1.31}$$

where $P(s)$ is the plant transfer function in Equation 1.15.

The phase margin and crossover frequency specifications listed in equations Equation 1.4 and Equation 1.5 of Section 1.1.1 can then be satisfied using a proportional gain K_c and the lead transfer function

$$G_{lead}(s) = \frac{1 + aTs}{1 + Ts} \tag{1.32}$$

The a and T parameters change the location of the pole and the zero of the lead compensator which changes the gain and phase margins of the system. The design process involves examining the stability margins of the *loop transfer function*

$$L(s) = C(s)P(s) \tag{1.33}$$

where the compensator is given by:

$$C(s) = \frac{K_c\,(1 + aTs)}{(1 + Ts)\,s} \tag{1.34}$$

1.3.1 Finding Lead Compensator Parameters

The Lead compensator is an approximation of a proportional-derivative (PD) control. A PD controller can be used to add damping to reduce the overshoot in the transient of a step response and effectively making the system more stable. In other words, it increases the phase margin. In this particular case, the lead compensator is designed for the following system:

$$L_p(s) = \frac{K_c P(s)}{s} \tag{1.35}$$

The proportional gain K_c is designed to attain a certain crossover frequency. Increasing the gain crossover frequency essentially increases the bandwidth of the system which decreases the peak time in the transient response (i.e. makes the response faster). However, as will be shown, adding a gain $K_c > 1$ makes the system less stable. The phase margin of the $L_p(s)$ system is therefore lower than the phase margin of the $P_i(s) = P(s)/s$ system and this translates to having a large overshoot in the response. The lead compensator is used to dampen the overshoot and increase the overall stability of the system, i.e increase its phase margin.

The frequency response of the lead compensator given in Equation 1.32 is

$$G_{lead}(\omega j) = \frac{1 + aT\omega j}{1 + T\omega j} \tag{1.36}$$

and its corresponding magnitude and phase equations are

$$|G_{lead}(\omega j)| = \sqrt{\frac{T^2\omega^2 a^2 + 1}{1 + T^2\omega^2}} \tag{1.37}$$

and

$$\phi_G = \text{arctan}\, aT\omega - \text{arctan}\, T\omega \tag{1.38}$$

The Bode plot of the lead compensator is shown in Figure 1.5.

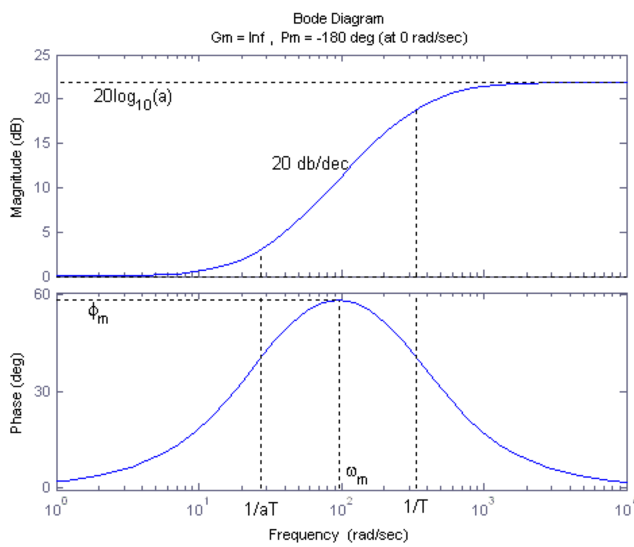


Figure 1.5: Bode of lead compensator.

1.3.2 Lead Compensator Design using MATLAB

In this section, we will use MATLAB® to design a lead compensator that will satisfy the frequency-based specifications given in Section 1.1.1.

1. **Bode plot of the open-loop uncompensated system**, $P_i(s)$, must first be found. To generate the Bode plot of $P_i(s)$, enter the following commands in MATLAB®. **NOTE:** If your system has not been set up yet, then you need to first run the the setup_servo_spd_cntrl.m script. This script will store the model parameter *K* and *tau* in the MATLAB® workspace. These parameters are used with the commands *tf* and *series* to create the $P_i(s)$ transfer function. The *margin* command generates a Bode plot of the system and it lists the gain and phase stability margins as well as the phase and gain crossover frequencies.

```
% Plant transfer function
P = tf([K],[tau 1]);
% Integrator transfer function
I = tf([1],[1 0]);
% Plant with Integrator transfer function
Pi = series(P,I);
% Bode of Pi(s)
figure(1)
margin(Pi);
set (1,'name','Pi(s)');
```

The entire Lead compensator design is given in the *d_lead.m* script file. Run this script after running the setup_servo_spd_cntrl.m script when CONTROL_TYPE = 'AUTO' to generate a collection of Bode diagrams including the Bode of $P_i(s)$ given in Figure 1.6.

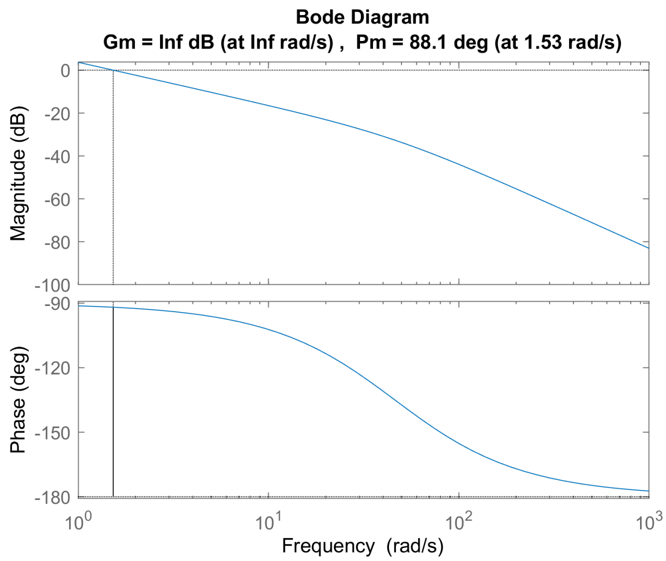


Figure 1.6: Bode of $P(s)/s$ system.

2. **Find how much more gain is required** such that the gain crossover frequency is 25.0 rad/s (you can click directly on the MATLAB® Figure to get data point). As mentioned before, the lead compensator adds gain to the system and will increases the phase as well.

   As given in Figure 1.6, the crossover frequency of the uncompensated system is 1.53 rad/s. To move the crossover frequency to 25.0 rad/s, a gain of

$$K_c = 25.4 \text{ dB} \tag{1.39}$$

   or

$$K_c = 18.6 \text{ V/rad} \tag{1.40}$$

   in the linear range is required. The Bode plot of the loop transfer function $L_p(s)$ (from Section 1.3) is given in Figure 1.7. This initial estimate of the gain was found using the *ginput* command. The gain was then adjusted according to the crossover frequency calculated in the generated Bode plot of the $L_p(s)$ system. The commands used to generate the Bode plot are given in the d_lead.m script.

3. **Gain needed for specified phase margin** must be found next so that the lead compensator can achieve the specified phase margin of 72.5 degrees.

   To attain the necessary phase margin, the maximum phase of the lead can be calculated using

$$\phi_m = PM_{des} - PM_{meas} \tag{1.41}$$

   Given that the desired phase margin in Equation 1.4 and the phase margin of $L_p(s)$ is

$$PM_{meas} = 61.5 \text{ deg} \tag{1.42}$$

   the maximum lead phase has to be about

$$\phi_m = 11.0 \text{ deg} \tag{1.43}$$

   or

$$\phi_m = 0.192 \text{ rad} \tag{1.44}$$

   The lead compensator, as explained in Section 1.3.1, has two parameters: a and T. To attain the maximum phase ϕ_m shown in Figure 1.5, the Lead compensator has to add $20\log_{10}(a)$ of gain. This is determined using the equation

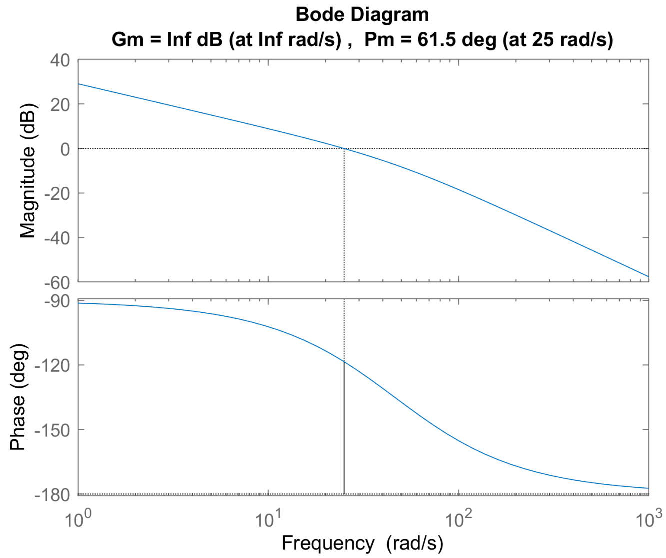


Figure 1.7: Bode of $L_p(s) = K_c P(s)/s$ system.

$$a = -\frac{1 + \sin(\phi_m)}{-1 + \sin(\phi_m)} \tag{1.45}$$

The gain needed is found by inserting the max phase into this equation to get

$$a = 1.472 \tag{1.46}$$

which is

$$20 \log_{10}(a) = 3.36 \text{ dB} \tag{1.47}$$

4. **The frequency at which the lead maximum phase occurs** must be placed at the new gain crossover frequency $\omega_{g,new}$. This is the crossover frequency after the lead compensator is applied. As illustrated in Figure 1.5, ω_m occurs halfway between 0 dB and $20 \log_{10}(a)$, i.e. at $10 \log_{10}(a)$. So, the new gain crossover frequency in the $L_p(s)$ system will be the frequency where the gain is $-10 \log_{10}(a)$.

   From Figure 1.7, it is found that the frequency where the $-10 \log_{10}(a) = -1.68$ dB gain in the $L_p(s)$ system occurs is at about 29.2 rad/s. Thus, the maximum phase of the lead will be set to

$$\omega_m = 29.2 \text{ rad/s} \tag{1.48}$$

   As illustrated earlier in Figure 1.5 in Section 1.3.1, the maximum phase occurs at the maximum phase frequency ω_m. Parameter T given by:

$$T = \frac{1}{\omega_m \sqrt{a}} \tag{1.49}$$

   is used to attain a certain maximum phase frequency. This changes where the Lead compensator breakpoint frequencies $1/(a * T)$ and $1/T$ shown in Figure 1.5 occur. The slope of the lead compensator gain changes at these frequencies. We can find the parameter T by substituting $\omega_m = 80.9$ and the lead gain value from Equation 1.46 into Equation 1.49:

$$T = 0.0282 \text{ s/rad} \tag{1.50}$$

   Therefore, the lead breakpoint frequencies are:

$$\frac{1}{aT} = 24.1 \text{ rad/s} \tag{1.51}$$

   and

$$\frac{1}{T} = 35.4 \text{ rad/s} \tag{1.52}$$

5. **Bode plot of the lead compensator** $C_{lead}(s)$, defined in Equation 1.32 can be generated using the *d_lead.m* script.

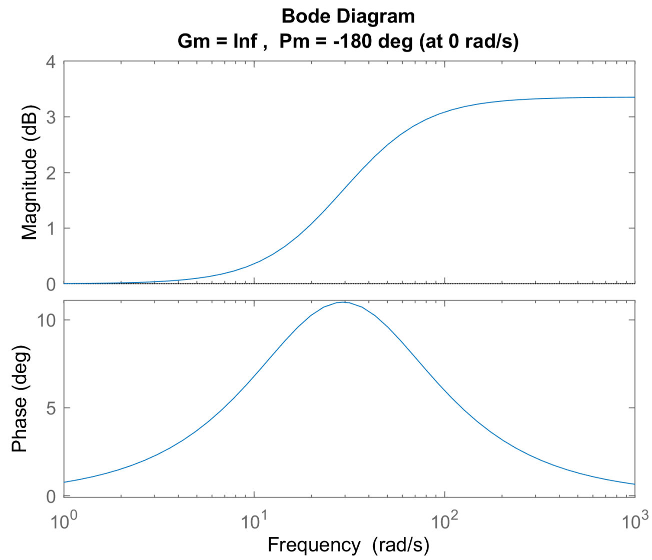


Figure 1.8: Bode of lead compensator $C_{lead}(s)$.

6. **Bode plot of the loop transfer function** $L(s)$, as described in Equation 1.33, can be generated using the d_lead.m script. The phase margin of $L(s)$ is 68.1 degrees and is below the desired phase margin of 75.0 degrees, as specified in Section 1.1.

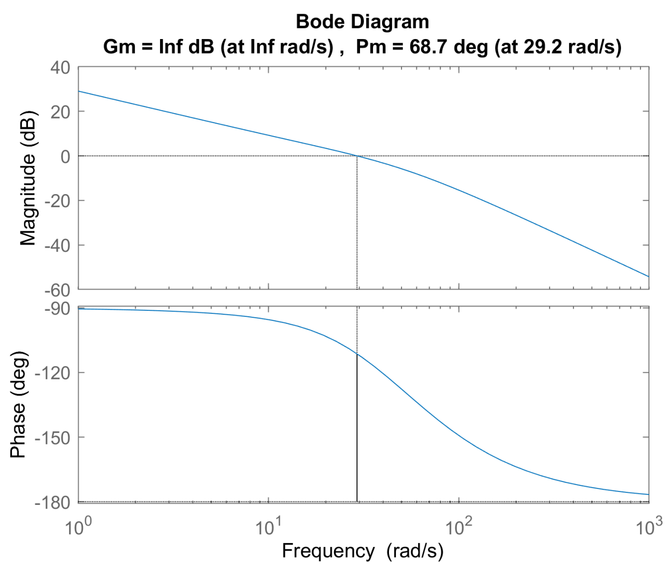


Figure 1.9: Bode of loop transfer function $L(s)$.

7. **Check response** by simulating the system to make sure that the time-domain specifications are met. Keep in mind that the goal of the lead design is the same as the PI control, the response should meet the desired steady-state error, peak time, and percentage overshoot specifications given in Section 1.1. Thus, if the crossover frequency and/or phase margin specifications are not quite satisfied, the response should be simulated to verify if the time-domain requirements are satisfied. If so, then the design is complete. If not, then the lead design needs to be re-visited.

   You will work on this later in the laboratory as described in Section 3.2.1.

2 Pre-Lab Questions

1. Based on the steady-state error result of a step response from Equation Equation 1.17 ,what *type* of system is the Rotary Servo Base Unit when performing speed control (Type 0, 1, or 2) and why?
2. Calculate the PI control gains needed to satisfy the time-domain response requirements. Use the nominal Rotary Servo Base Unit model parameters, K and τ, found in Rotary Servo Base Unit in Modeling laboratory experiment.
3. Find the frequency response magnitude, $|P_i(\omega)|$, of the transfer function $P_i(s)$ given in Equation Equation 1.31.
4. Calculate the DC gain of $P_i(s)$ given in Equation Equation 1.31. **Hint:** The DC gain is the gain when the frequency is zero, i.e. $\omega = 0\, rad/s$. However, because of its integrator, $P_i(s)$ has a singularity at zero frequency. Therefore, the DC gain is not technically defined for this system. Instead, approximate the DC gain by using $\omega = 1\, rad/s$. Make sure the DC gain estimate is evaluated numerically in dB using the nominal model parameters, K and τ, found in Modeling laboratory experiment).
5. The gain crossover frequency, ω_g, is the frequency at which the gain of the system is 1 or 0 dB. Express the crossover frequency symbolically in terms of the Rotary Servo Base Unit model parameters K and τ. Then, evaluate the expression using the nominal Rotary Servo Base Unit model parameters you found in Modeling laboratory experiment).

3 Lab Experiments

The main goal of this laboratory is to explore closed-loop speed control of the Rotary Servo Base Unit load shaft.

In this laboratory you will conduct two experiments:

1. Step response with PI control, and
2. Step response with Lead control

In each of the experiments, 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 with PI Control

3.1.1 Simulation

First you will simulate the closed-loop speed response of the Rotary Servo Base Unit with a PI controller to step input. Our goals are to confirm that the desired reponse specifications in an ideal situation are satisfied and to verify that the motor is not saturated. Then, you will explore the effect of the setpoint weight.

Experimental Setup

The s_servo_speed_cntrl SIMULINK® diagram shown in Figure 3.1 is used to simulate the closed-loop speed response of the Rotary Servo Base Unit when using either the PI or Lead controls. The Rotary Servo Base Unit Model uses a Transfer Fcn block from the SIMULINK® library to simulate the system. The PI compensator subsystem contains the PI control detailed in Section 1.2 and the *Lead Compensator* block has the compensator described in Section 1.3.

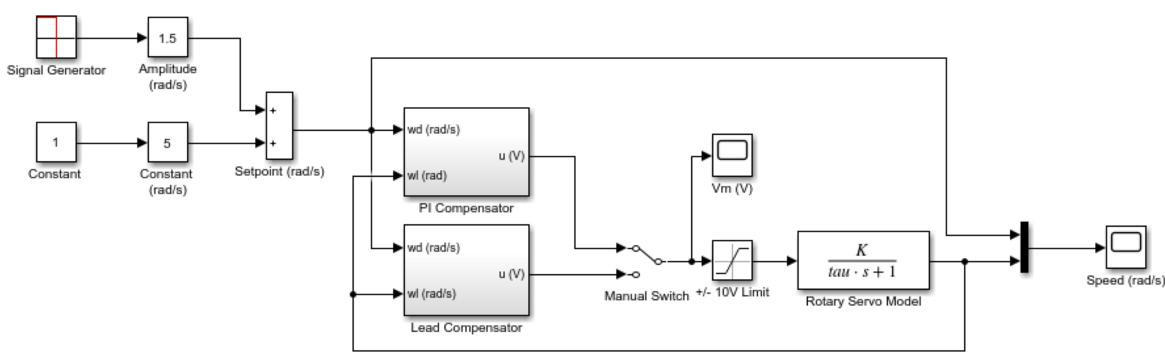


Figure 3.1: Simulink diagram used to simulate the closed-loop Rotary Servo Base Unit speed response.

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.

1. Enter the proportional and integral control gains found in Section 1.2.2 as `kp` and `ki` in MATLAB®.
2. The speed reference signal is to be a 0.4 Hz square wave that goes between 3.5 rad/s and 6.5 rad/s. Set the Signal Generator block parameters to the following:
   - Signal type = *square*

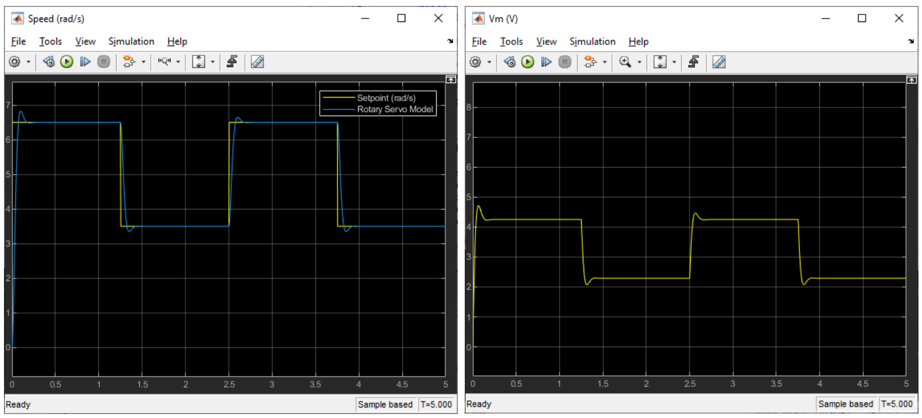
- Amplitude = 1
- Frequency = 0.4 Hz

3. In the s_servo_speed_cntrl SIMULINK® model, set the *Amplitude (rad/s)* gain block to 1.5 rad/s and the *Offset (rad/s)* block to 5.0 rad/s.

4. Set the Manual Switch to the upward position to activate the PI control.

5. Open the load gear angular velocity scope, *Speed (rad/s)*, and the motor input voltage scope, *Vm (V)*.

6. 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 *Speed (rad/s)* scope, the yellow trace is the setpoint velocity and the blue trace is the simulated speed generated by the Rotary Servo Model block.



(a) Angular rate of load gear

(b) Motor input voltage.

Figure 3.2: Simulated PI speed response

7. Generate a MATLAB® figure showing the simulated PI speed response and its input voltage. After each simulation run, each scope automatically saves their response to a variable in the Matlab workspace. The *Speed (rad/s)* scope saves its response to the variable called *data_spd* and the *Vm (V)* scope saves its data to the *data_vm* variable. The *data_spd* variable has the following structure: *data_spd(:,1)* is the time vector, *data_spd(:,2)* is the setpoint, and *data_spd(:,3)* is the simulated angular speed. For the *data_vm* variable, *data_vm(:,1)* is the time and *data_vm(:,2)* is the simulated input voltage.

8. 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 1.1.1?

3.1.2 Implementing PI Speed Control

Experimental Setup

The q_servo_speed_cntrl SIMULINK® diagram shown in Figure 3.3 is used to perform the speed control exercises in this laboratory. It contains QUARC® blocks that interface with the DC motor and sensors of the Rotary Servo Base Unit system, similarly as discussed in Integration laboratory experiment. The PI control subsystem implements the PI control detailed in Section 1.2 and the Lead Compensator block implements the lead control described in Section 1.3.

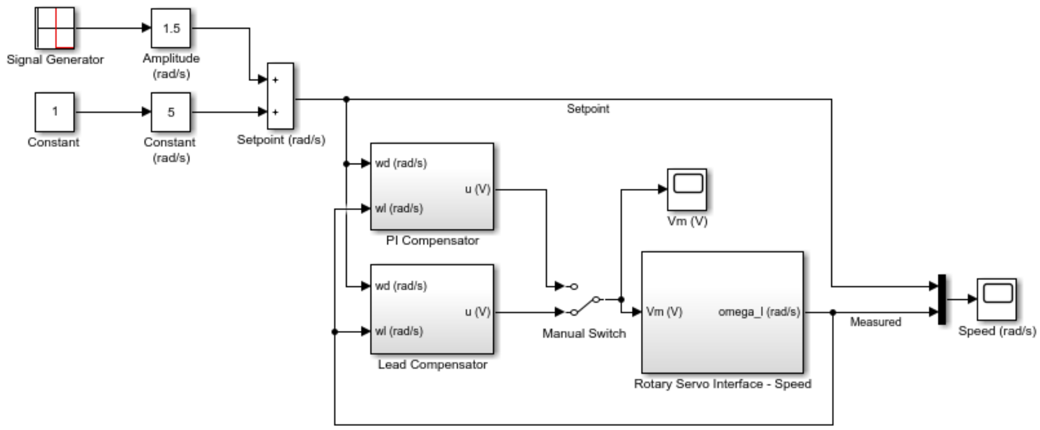
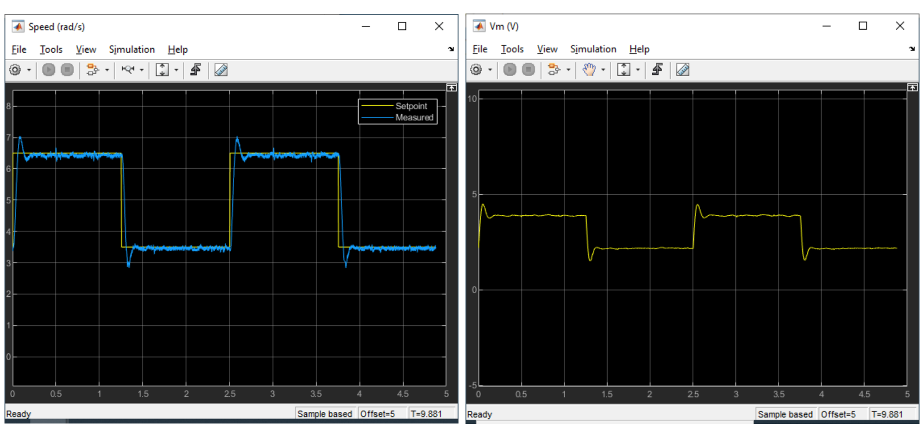


Figure 3.3: Simulink model used with QUARC to run the PI and lead speed controllers on the Rotary Servo Base Unit.

1. Run the `setup_servo_spd_cntrl.m` script.
2. Enter the proportional and integral control gains found in Section 2 as *kp* and *ki* in MATLAB®.
3. To generate a square reference speed signal, set the *Signal Generator* block parameters to the following:
   - Signal type = *square*
   - Amplitude = 1
   - Frequency = 0.4 Hz
4. In the Speed Control SIMULINK® model, set the *Amplitude (rad/s)* gain block to 1.5 rad/s and the *Offset (rad/s)* constant block to 5.0 rad/s.
5. Open the load shaft speed scope, *Speed (rad/s)*, and the motor input voltage scope, *Vm (V)*.
6. Set the *Manual Switch* to the upward position to activate the PI control.
7. Click on QUARC | Build to compile the SIMULINK® diagram.
8. Select QUARC | Start to run the controller. The scopes should be displaying responses similar to figures Figure 3.4a and Figure 3.4b. Note that in the *Speed (rad/s)* scope, the yellow trace is the setpoint speed while the blue trace is the measured speed.



(a) Angular rate of load gear. (b) Motor input voltage.

9. When a suitable response is obtained, click on the *Stop* button in the SIMULINK® diagram toolbar (or select QUARC | Stop from the menu) to stop running the code. Generate a MATLAB® figure showing the PI speed response and its input voltage. As in the s_servo_speed_cntrl SIMULINK® diagram, when the controller is stopped each scope automatically saves their response to a variable in the MATLAB® workspace. Thus, the *Speed (rad/s)* scope saves its response to the *data_spd* variable and the *Vm (V)* scope saves its data to the *data_vm* variable.

10. Measure the percent overshoot and the peak time of the Rotary Servo Base Unit load gear step response. Does the response satisfy the specifications given in Section 1.1.1?

    Note: There is noise when estimation the velocity from the encoder, known as the ripple velocity, that is assessed in Integration laboratory experiment. This effect the response measurements.

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

12. 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 Step Response with LEAD Control

3.2.1 Simulation

You will simulate the closed-loop speed response of the Rotary Servo Base Unit with a Lead controller to step input. Our goals are to confirm that the desired reponse specifications in an ideal situation are satisfied and to verify that the motor is not saturated.

As in the step response with PI control experiment in Section 3.1.1, in this experiment you need to use thes_servo_speed_cntrl SIMULINK® diagram shown in Figure 3.3 again.

1. Enter the Lead control parameters found in Section 1.3.2. These are denoted as *Kc*, *a*, and *T* in MATLAB®.
2. Set the *Signal Generator* block parameters to the following:
   - Signal type = *square*
   - Amplitude = 1
   - Frequency = 0.4 Hz
3. In the Speed Control SIMULINK® model, set the *Amplitude (rad/s)* gain block to 1.5 rad/s and the *Offset (rad/s)* constant block to 5.0 rad/s.
4. To engage the lead control, set the *Manual Switch* to the downward position.
5. Open the load shaft position scope, *Speed (rad)*, and the motor input voltage scope, *Vm (V)*.
6. Start the simulation. By default, the simulation runs for 5 seconds. The scopes should be displaying responses similar to Figures Figure 3.2a and Figure 3.2b.
7. Verify if the time-domain specifications in Section 1.1.1 are satisfied and that the motor is not being saturated. To calculate the steady-state error, peak time, and percent overshoot, use the simulated response data stored in the *data_spd* variable.
8. If the specifications are not satisfied, go back in the lead compensator design. You may have to, for example, add more maximum phase in order to increase the phase margin. If the specifications are met, move on to the next step.
9. Generate a MATLAB® figure showing the *Simulated Lead* speed response and its input voltage.

3.2.2 Implementing LEAD Speed Control

In this section the speed of the Rotary Servo Base Unit is controlled using the lead compensator. Measurements will be taken to see if the specifications are satisfied.

1. Run the setup_servo_spd_cntrl.m script.
2. Enter the *Kc*, *a*, and *T*, lead parameters found in Section 1.3.2 in MATLAB®.
3. Set the *Signal Generator* block parameters to the following:
   - Signal type = *square*
   - Amplitude = 1
   - Frequency = 0.4 Hz
4. In the Speed Control SIMULINK® model, set the *Amplitude (rad/s)* gain block to 1.5 rad/s and the *Offset (rad/s)* constant block to 5.0 rad/s.

5. To engage the lead compensator, set the *Manual Switch* in the Speed Control SIMULINK® diagram to the downward position.
6. Open the load shaft speed scope, *Speed (rad/s)*, and the motor input voltage scope, *Vm (V)*.
7. Click on QUARC | Build to compile the SIMULINK® diagram.
8. Select QUARC | Start to run the controller. The scopes should be displaying responses similar to Figure 3.4a and Figure 3.4b.
9. When a suitable response is obtained, click on the *Stop* button in the Simulink diagram toolbar (or select QUARC | Stop from the menu) to stop running the code. Generate a MATLAB® figure showing the lead speed response and its input voltage.
10. Measure the steady-state error, the percent overshoot, and the peak time of the Rotary Servo Base Unit load gear. For the steady-state error, it may be beneficial to give a constant reference and take its average as done in Section 3.1.2. Does the response satisfy the specifications given in Section 1.1.1?
11. Using both your simulation and implementation results, comment on any differences between the PI and lead controls.
12. Click the Stop button on the SIMULINK® diagram toolbar (or select QUARC | Stop from the menu) to stop the experiment.
13. Turn off the power to the amplifier if no more experiments will be performed on the SRV02 in this session.

3.3 Results

Fill out Table 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 2	**Pre-Lab: PI Gains**			
	Proportional Gain	k_p		V/(rad.s)
	Integral Gain	k_i		V/(rad.s)
	Open-Loop Time Constant	τ		s
	Open-Loop Steady-state Gain	K		rad/(V.s)
Question 4	**Pre-Lab: DC Gain Estimate**			
	DC Gain Estimate of $P_i(s)$	$\vert P_i(1)\vert$		dB
Question 5	**Pre-Lab: Gain Crossover Frequency**			
	Gain crossover frequency	ω_g		rad/s
Section 3.1.1	**In-Lab: PI Step Response Simulation**			
	Peak time	t_p		s
	Percent overshoot	PO		%
	Steady-state error	e_{ss}		rad/s
Section 3.1.2	**In-Lab: PI Speed Control Implementation**			
	Peak time	t_p		s
	Percent overshoot	PO		%
	Steady-state error	e_{ss}		rad/s
Section 3.2.1	**In-Lab: Step Response Simulation with Lead Control**			
	Peak time	t_p		s
	Percent overshoot	PO		%
	Steady-state error	e_{ss}		rad/s
Section 3.2.2	**In-Lab: Lead Speed Control Implementation**			
	Peak time	t_p		s
	Percentage overshoot	PO		%
	Steady-state error	e_{ss}		rad/s

Table 3.1: Summary of results for the Speed Control laboratory.

4 File Description and Configuration

4.1 Overview of Files

File Name	Description
Rotary Servo Base Unit Speed Control Workbook (Student)pdf	This laboratory guide contains pre-lab questions and lab experiments demonstrating how to design and implement a speed controller on the Quanser Rotary Servo Base Unit rotary plant using QUARC®.
setup_servo_spd_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 an 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_speed_cntrl	Simulink file that simulates the closed-loop Rotary Servo Base Unit speed control using either the PI control or the lead compensator.
q_servo_speed_cntrl	Simulink file that runs the PI or Lead speed control on the actual Rotary Servo Base Unit system using QUARC®.

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

4.2 Setup for Speed Control Simulation

Follow these steps to configure the MATLAB® setup script and the SIMULINK®diagram for the Speed Control simulation laboratory:

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 speed controller files, e.g. s_servo_speed_cntrl.
3. Double-click on the s_servo_speed_cntrl file to open the Rotary Servo Base Unit Speed Control Simulation Simulink diagram shown in Figure Figure 3.1.
4. Double-click on the setup_servo_spd_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, e.g. the EXT_GEAR_CONFIG should be set to 'HIGH' and the LOAD_TYPE should be set to 'DISC'.
6. Set the CONTROL_TYPE to 'MANUAL' to find the control gains yourself.
7. Run the script by selecting the Debug | Run item from the menu bar or clicking on the Run button in the tool bar. The messages shown below, should be generated in the MATLAB® Command Window. The correct model

parameters are loaded but the control gains and related parameters loaded are default values that need to be changed. That is, the PI control gains are all set to zero, the lead compensator parameters *a* and *T* are both set to 1, and the compensator proportional gain *Kc* is set to zero.

```
Servo model parameters:
   K = 1.53 rad/s/V
   tau = 0.0217 s
PI control gains:
   kp = 0 V/rad
   ki = 0 V/rad/s
Lead compensator parameters:
   Kc = 0 V/rad/s
   1/(a*T) = 1 rad/s
   1/T = 1 rad/s
```

4.3 Setup for Speed Control Implementation

Before beginning the in-lab exercises on the SRV02 device, the q_srv02_spd SIMULINK®diagram and the setup_servo_spd_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 SRV02 speed control files, e.g. q_servo_speed_cntrl.
4. Double-click on the q_servo_speed_cntrl file to open the Speed Control Simulink diagram shown in Figure 3.3.
5. **Configure DAQ:** Double-click on the HIL Initialize block in the Rotary Servo Interface - Speed subsystem and ensure it is configured for the DAQ device that is installed in your system. See QUARC® docuementation for more information on configuring the HIL Initialize block.
6. **Configure setup script:** Set the parameters in the setup_servo_spd_cntrl.m script according to your system setup. See Section 4.2 for more details.

5 Lab Report

This laboratory contains two experiments, namely,

1. step response with PI control, and
2. step response with lead control.

When you are writing your report, follow the outline corresponding to the experiment you conducted to build the *content* of your report. Also, in Section 5.3 you can find some basic tips for the *format* of your report.

5.1 Template for Content (PI Control Experiments)

I. PROCEDURE

I.1. Step Response with PI Control

1. *Simulation*
   - Briefly describe the main goal of this simulation.
   - Briefly describe the procedure (Section 3.1.1)
2. *Implementation*
   - 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 7 in Section 3.1.1, *Step response simulation with PI Control*
2. Response plot from step 9 in Section 3.1.2, *Step response implementation with PI Control*
3. Provide 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:

III.1. Step Response with PI Control

1. Step 8 in Section 3.1.1, *Step response simulation with PI Control*
2. Step 10 in Section 3.1.2, *Step response implementation with PI Control*

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

1. Step 8 in Section 3.1.1, *Step response simulation with PI Control*
2. Step 10 in Section 3.1.2, *Step response implementation with PI Control*

5.2 Template for Content (Lead Control Experiments)

I. PROCEDURE

I.1. Step Reponse with Lead Control

1. *Simulation*
   - Briefly describe the main goal of this simulation.
   - Briefly describe the procedure (Section 3.2.1)
2. *Implementation*
   - 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 9 in Section 3.2.1, *Step response simulation with Lead Control*
2. Response plot from step 9 in Section 3.2.2, *Step response implementation with Lead Control*
3. Provide 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:

III.1. Step Response with Lead Control

1. Step 7 in Section 3.2.1, *Step response simulation with Lead Control*
2. Step 10 in Section 3.2.2, *Step response implementation with Lead Control*

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

1. Step 7 in Section 3.2.1, *Step response simulation with Lead Control*
2. Step 10 in Section 3.2.2, *Step response implementation with Lead Control*
3. Step 11 in Section 3.2.2, *Step response implementation with Lead Control*

5.3 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）

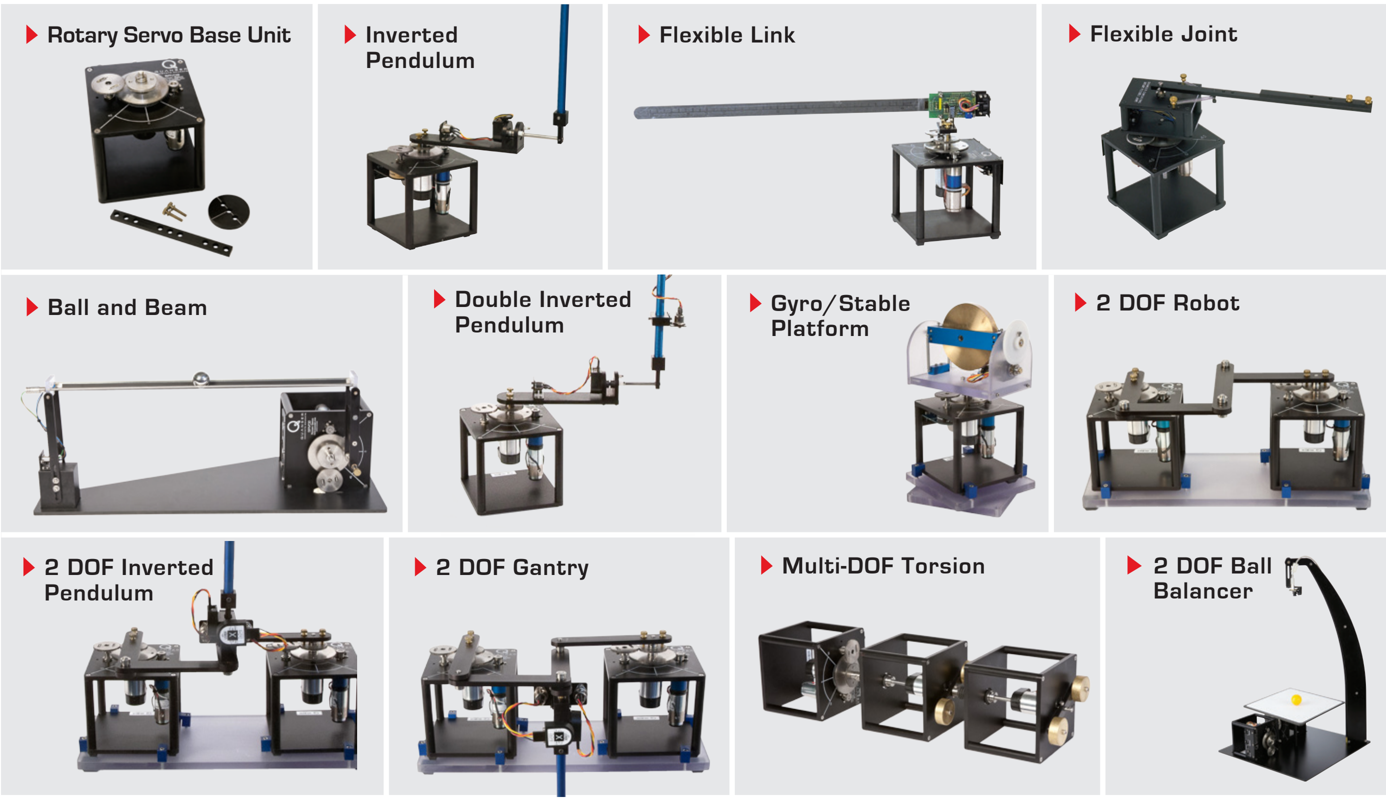
中国客户 National Instruments 符合中国电子信息产品中限制使用某些有害物质命令 （RoHS）。关于National Instruments 中国 RoHS合规性信息，请登录 ni.com/environment/rohs_china (For information about China RoHS compliance, go to ni.com/environment/rohs_china)

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