Rotary Motion Servo Plant: SRV02

Rotary Experiment #02: Position Control

SRV02 Position Control using QuaRC

Student Manual

Table of Contents

1. Introduction

The objective of this experiment is to develop a feedback system that controls the position of the rotary servo load shaft. Using the proportional-integral-derivative (PID) family, a compensator is designed to meet a set of specifications.

The following topics are covered in this laboratory:

- Design a proportional-velocity (PV) compensator that controls the position of the servo load shaft according to certain time-domain requirements.
- Actuator saturation.
- Design a proportional-velocity-integral (PIV) control to track a ramp reference signal.
- Simulate the PV and PIV controllers using the developed model of the plant and ensure the specifications are met without any actuator saturation.
- Implement the controllers on the Quanser SRV02 device and evaluate its performance.

2. Prerequisites

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

- Data acquisition card (e.g. Q8), the power amplifier (e.g. UPM), and the main components of the SRV02 (e.g. actuator, sensors), as described in References [1], [4], and [5], respectively.
- Wiring and operating procedure of the SRV02 plant with the UPM and DAC device, as discussed in Reference [5].
- Transfer function fundamentals, e.g. obtaining a transfer function from a differential equation.
- Laboratory described in Reference [6] in order to be familiar using QuaRC with the SRV02.

3. Overview of Files

Table 1 below lists and describes the various files supplied with the SRV02 Position Control laboratory.

File Name	***Description***
02 – SRV02 Position Control–Student Manual.pdf	This laboratory guide contains pre-lab and in-lab exercises demonstrating how to design and implement a position controller on the Quanser SRV02 rotary plant using QuaRC.
setup_srv02_exp02_pos.m	The main Matlab script that sets the SRV02 motor and sensor parameters as well as its configuration-dependent model parameters. **Run this file only 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 UPM limits VMAX_UPM and IMAX_UPM.
d_model_param.m	Calculates the SRV02 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_srv02_pos.mdl	Simulink file that simulates a closed-loop PIV controller for the SRV02 system.
q_srv02_pos.mdl	Simulink file that implements a closed-loop PIV position controller on the SRV02 system using QuaRC.

File Name	Description

Table 1: Files supplied with the SRV02 Position Control experiment.

4. Pre-Lab Assignments

4.1. Desired Position Control Response

The block diagram shown in Figure 1 is a general unity feedback system with a compensator 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 yield to certain specifications.

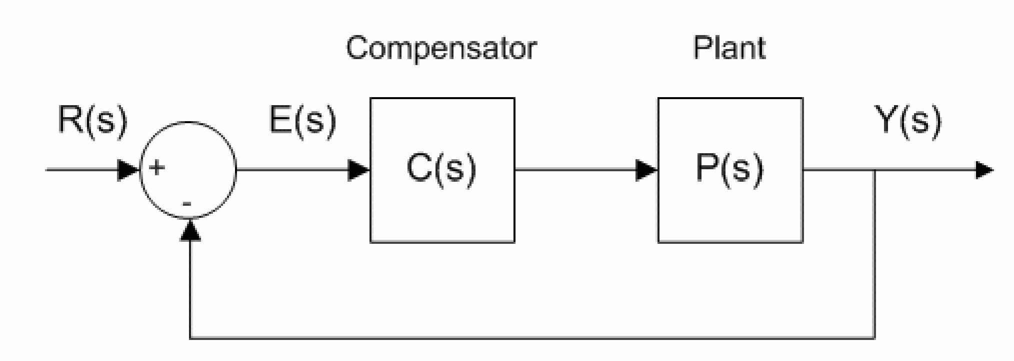

Figure 1: Unity feedback system.

The output of the system shown in Figure 1 is

$$Y(s) = C(s)\,P(s)\,(R(s) - Y(s)) \qquad [1]$$

and by solving for Y(s) the resulting closed-loop transfer function

$$Y(s) = \frac{C(s)\,P(s)\,R(s)}{1 + C(s)\,P(s)} \qquad [2]$$

is obtained. Recall in Reference [7], the SRV02 voltage-to-speed transfer function was derived. Placing that model in series with an integrator gives the open-loop voltage-to-load gear position transfer function

$$P(s) = \frac{K}{(\tau\, s + 1)\, s}\,. \qquad [3]$$

Using a proportional compensator with a second-order plant, such as in [3], the closed-loop transfer function in [2] has the form

$$\frac{Y(s)}{R(s)} = \frac{\omega_n^2}{s^2 + 2\,\zeta\,\omega_n\, s + \omega_n^2} \qquad [4]$$

where ω_n is the natural undamped frequency and ζ is the damping ratio. This is called the *standard second-order* transfer function and the properties of its response depend on the values of ω_n and ζ parameters.

4.1.1. Peak Time and Overshoot

Consider when a second-order system as shown in Equation [4] is subjected to a reference step

$$R(s) = \frac{R_0}{s} \qquad [5]$$

with an step amplitude of $R_0 = 1.5$. The obtained response is shown in Figure 2, below, where the red trace is the output response, y(t), and the blue trace is the reference step, r(t).

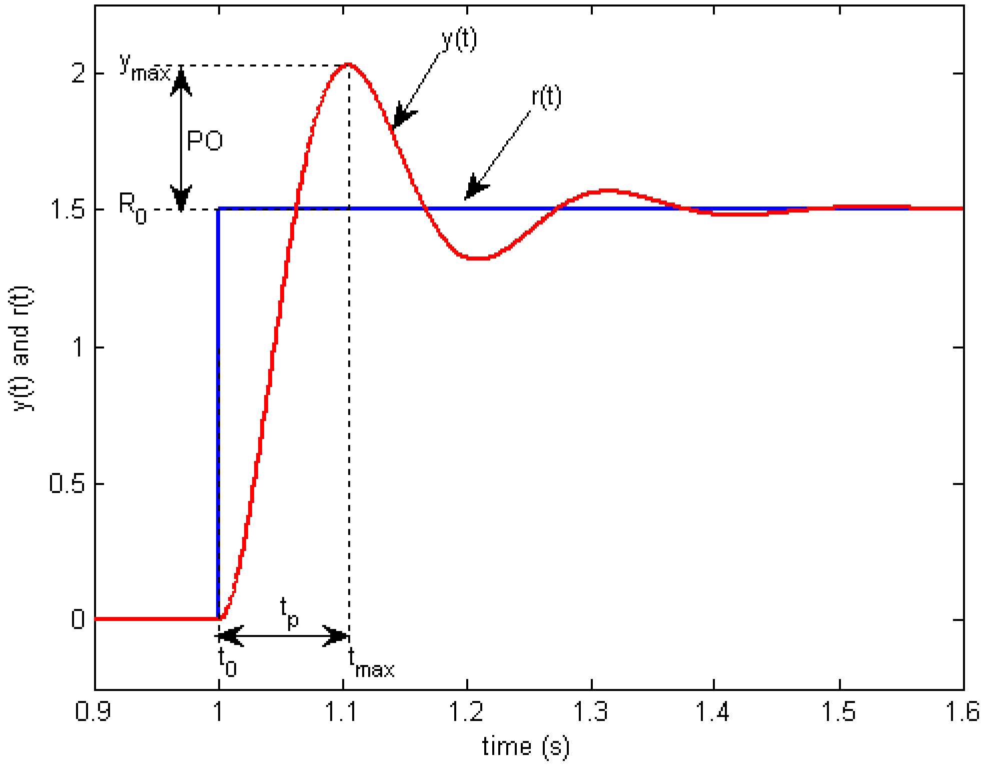

Figure 2: Standard second-order step response.

The maximum value of the response is denoted by the variable y_{max} and it occurs at a time t_{max}. For a response similar to Figure 2, the *percentage overshoot* is found using the equation

$$PO = \frac{100\,(y_{max} - R_0)}{R_0} \quad [6]$$

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

$$t_p = t_{max} - t_0 \quad [7]$$

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

$$PO = 100\,\mathbf{e}^{\left(-\frac{\pi\,\zeta}{\sqrt{1-\zeta^2}}\right)} \quad [8]$$

The peak time depends on both the damping ratio and natural frequency of the system and it can be derived that the relationship between them is

$$t_p = \frac{\pi}{\omega_n \sqrt{1 - \zeta^2}} \quad [9]$$

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

4.1.2. Steady-state Error

Steady-state error is illustrated in the ramp response given in Figure 3 and is denoted by the variable e_{ss}. It is the difference between the reference 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

$$e_{ss} = r_{ss}(t) - y_{ss}(t) \quad [10]$$

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

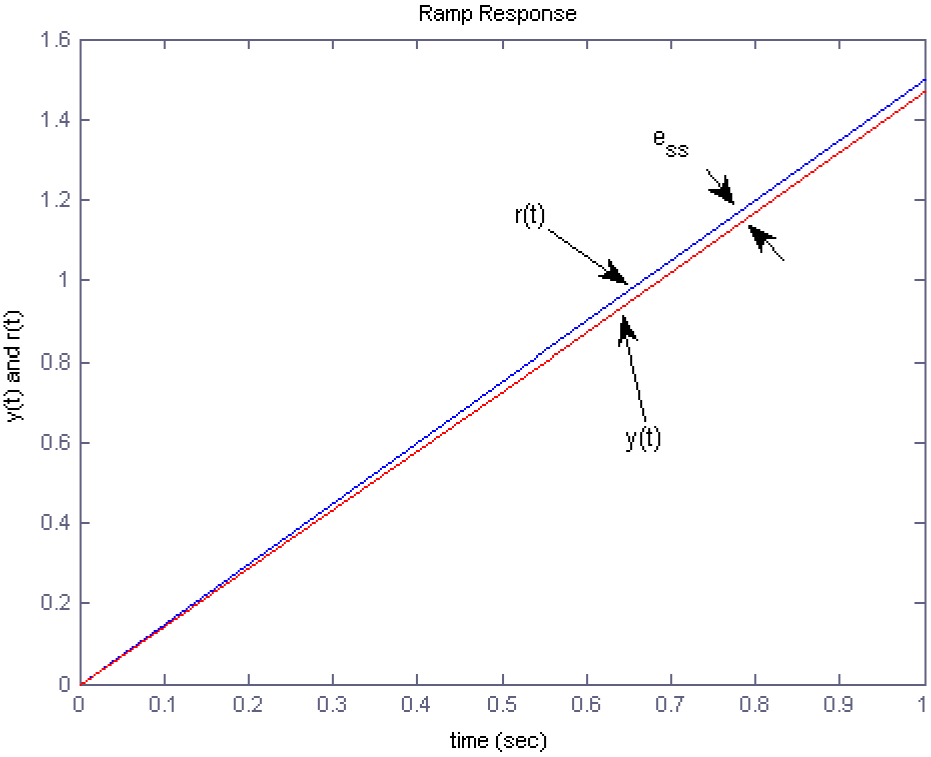

Figure 3: Steady-state error in ramp response.

1. Find the error transfer function E(s) in Figure 1 in terms of the reference R(s), the plant P(s),

and the compensator C(s).

0	1	2

2. Recall the final-value theorem

$$e_{ss} = \lim_{s \to 0} s \, \mathrm{E}(s) \quad . \qquad [11]$$

Given that the compensator equals

$$\mathrm{C}(s) = 1 \quad , \qquad [12]$$

the reference is the step defined in [5], and the process model is the transfer function in [3], find the steady-state error of the system.

0	1	2

3. Based on the steady-state error result from a step response with a unity proportional gain, what *Type* of system is the SRV02 when performing position control (0, 1, 2, or 3)? Given the *Type* of system, what would be the steady-state angle if a ramp input was given instead?

0	1	2

4.1.3. SRV02 Position Control Specifications

The time-domain specifications for controlling the position of the SRV02 load shaft are:

$$e_{ss} = 0 \quad , \tag{13}$$

$$t_p = 0.20\ [s] \quad \text{, and} \tag{14}$$

$$PO = 5.0\ [\text{"\%"}] \quad . \tag{15}$$

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.

Calculate the maximum overshoot of the response (in radians) given a step setpoint of 45 degrees, or

$$\theta_d(t) = \frac{\pi}{4} \quad . \tag{16}$$

0 1 2

4.2. PV Control Design

4.2.1. Closed-loop Transfer Function

The proportional-velocity (PV) compensator used to control the position of the SRV02 has the structure

$$V_m(t) = k_p(\theta_d(t) - \theta_l(t)) - k_v\left(\frac{d}{dt}\theta_l(t)\right) \qquad [17]$$

where k_p is the proportional control gain, k_v is the velocity control gain, $\theta_d(t)$ is the setpoint or reference load angle, and $\theta_l(t)$ is the measured load shaft angle, and $V_m(t)$ is the SRV02 input voltage. The block diagram of the PV control is illustrated in Figure 4.

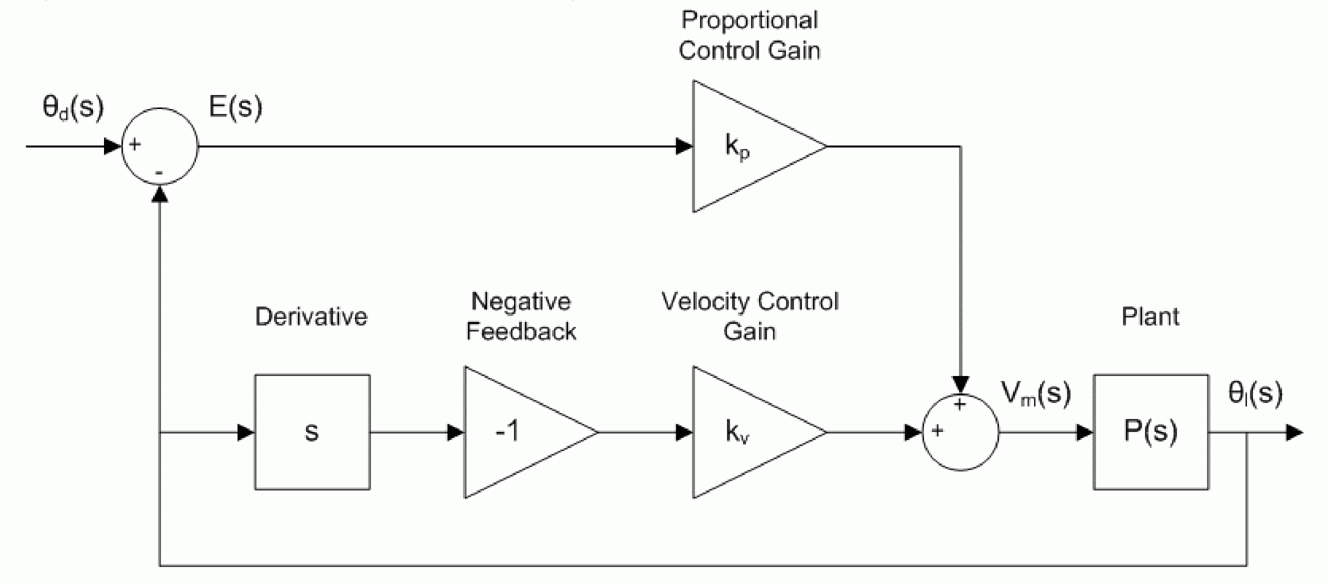

Figure 4: Block diagram of SRV02 PV position control.

1. Find the closed-loop SRV02 position control transfer function, $\Theta_l(s)/\Theta_d(s)$, using the time-domain PV control in Equation [17], the block diagram in Figure 4, and the process model in [3]. Assume the zero initial conditions, thus $\theta_l(0^-) = 0$.

0	1	2

4.2.2. Finding Gains to Satisfy Specifications

1. The resulting SRV02 closed-loop transfer function has the same structure as the *standard second-order system* given in [4]. The denominator of [4] is called the *standard characteristic equation,*

$$s^2 + 2\zeta\omega_n s + \omega_n^2 . \qquad [17]$$

Find the control gains k_p and k_v that map the characteristic equation of the SRV02 closed-loop system to the standard characteristic equation given above. With these two equations, the control gains can be designed based on a specified natural frequency, ω_n, and damping ratio, ζ.

0	1	2

2. Using the equations described in Section 4.1.1, express the natural frequency and the damping ratio in terms of percentage overshoot and peak time specifications.

0	1	2

3. Calculate the minimum damping ratio and natural frequency required to meet the specifications given in Section 4.1.3.

0	1	2

4. Based on the nominal SRV02 model parameters, K and τ, found in Experiment #1: SRV02 Modeling (Reference [7]), calculate the control gains needed to satisfy the time-domain response requirements.

0	1	2

4.2.3. Control Gain Limits

In control design, another factor to be considered is actuator saturation. This is a nonlinear element and is represented by a saturation block, as shown in Figure 5. Consider an SRV02 type setup where the PC computes a control voltage. The control voltage is outputted by a digital-to-analog channel of the data-acquisition device, $V_{dac}(t)$, and then amplified by a power amplifier by a factor of K_a. The amplified voltage, $V_{amp}(t)$, is saturated by either the maximum output voltage of the amplifier or the limits of the motor (whichever is smaller). The input SRV02 voltage, $V_m(t)$, is the effective voltage being applied to the motor.

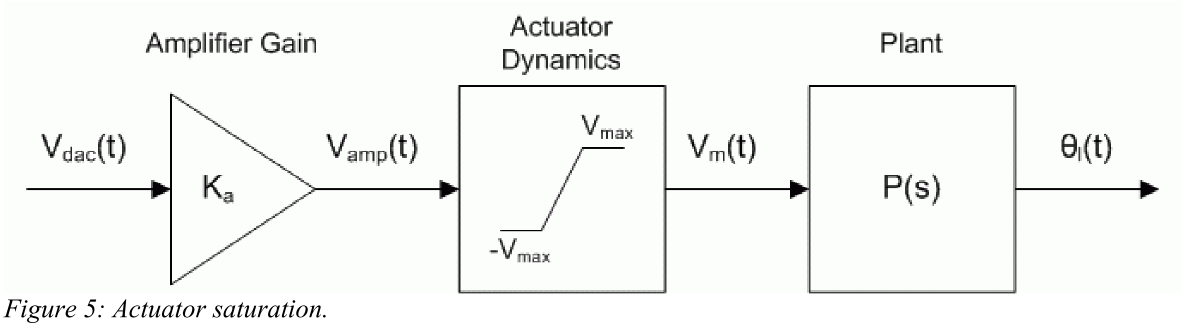

Figure 5: Actuator saturation.

1. The limitations of the actuator must be taken into account when designing a controller. For instance, the voltage entering the SRV02 motor should never exceed

$$V_{max} = 10.0\ [V] \quad . \qquad [18]$$

For the reference step given in [16] (i.e. 45 degree step), calculate the maximum proportional gain. Ignore the velocity control. Is the k_p designed above, which satisfies the specifications, below the maximum proportional gain?

0	**1**	**2**

4.3. Steady-state Error due to a Ramp Reference

From the previous steady-state analysis, it was found that the closed-loop SRV02 system was a certain type of system. This section investigates the steady-state error due to a ramp position setpoint when using PV and PIV controllers.

4.3.1. Ramp Steady-State Error using PV

The steady-state error when the SRV02 tracks a ramp setpoint using the PV controller is evaluated in this section.

1. Given the ramp setpoint

$$\mathrm{R}(s) = \frac{R_0}{s^2} \qquad [19]$$

find the error transfer function of the closed-loop SRV02 position PV control.

0	1	2

2. Find the steady-state error and evaluate it numerically given a ramp with a slope of

$$R_0 = 3.36 \left[\frac{rad}{s} \right], \qquad [20]$$

the control gains found in Section 4.2.2, and the nominal model parameters derived in Laboratory #1: SRV02 Modeling (see Reference [7]).

0	1	2

4.3.2. PIV Control

Adding an integral control can help to eliminate any steady-state error. The proportional-integral-velocity (PIV) algorithm that will be used to control the position of the SRV02 (in addition to PV) is of the form

$$V_m(t) = k_p(\theta_d(t) - \theta_l(t)) + k_i \int \theta_d(t) - \theta_l(t)\, dt - k_v \left(\frac{d}{dt} \theta_l(t) \right), \qquad [21]$$

where k_i is the integral gain. The PIV position control block diagram is shown in Figure 6.

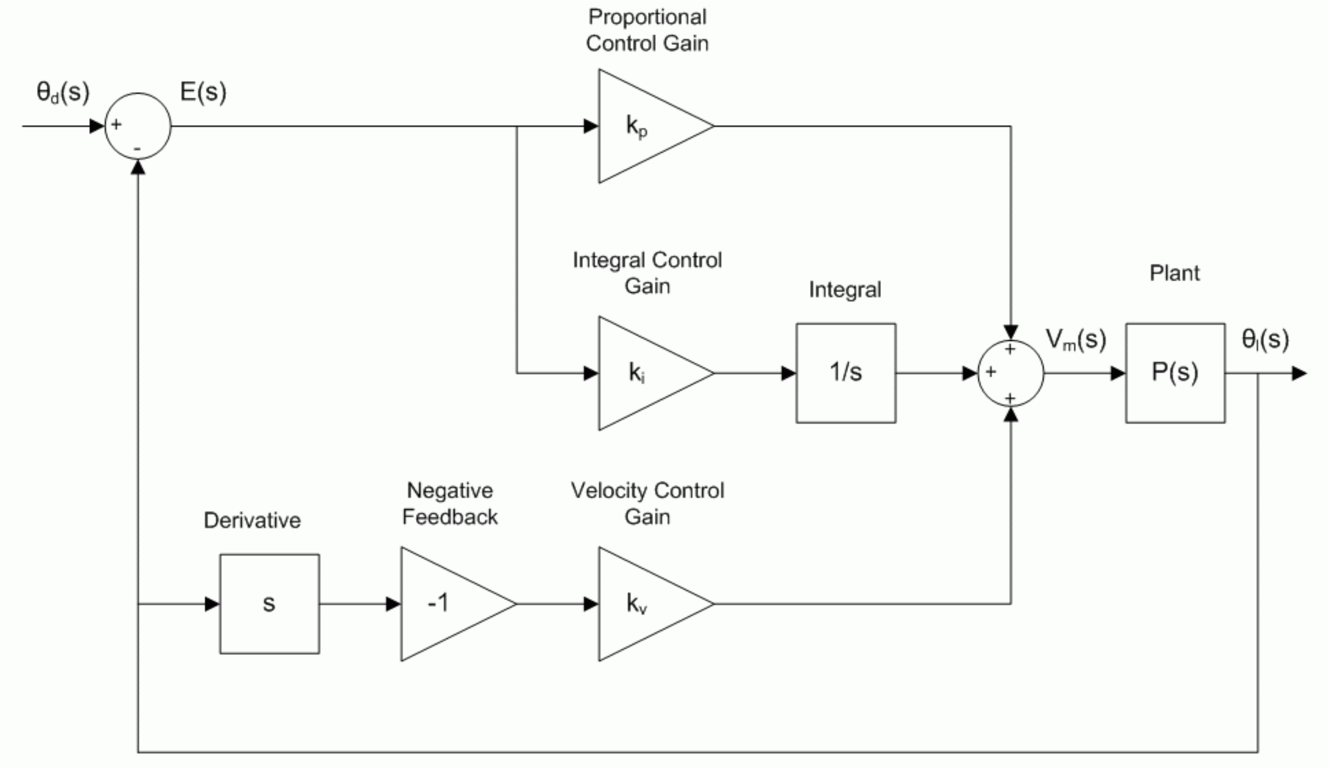

Figure 6: Block diagram of PIV SRV02 position control.

5. Find the closed-loop SRV02 position control transfer function, $\Theta_l(s)/\Theta_d(s)$, using the time-domain PIV control in Equation [21], the block diagram in Figure 5, and the process model in [3]. Assume the zero initial conditions, thus $\theta_l(0^-) = 0$.

0	1	2

4.3.3. Ramp Steady-State Error using PIV

In this section, the steady-state error when using the PIV control is evaluated.

1. Find the error transfer function of the closed-loop SRV02 position PIV control.

0	1	2

2. Evaluate the steady-state error.

0	1	2

4.3.4. 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 this section, the integral gain is calculated based on the steady-state error that is obtained with the PV control and the amount of voltage there is to spare.

1. In steady-state, the ramp response error is constant and does not vary. Therefore, to design an integral gain the velocity compensation can be neglected. Thus the PI control

   $$V_m(t) = k_p\,(\theta_d(t) - \theta_l(t)) + k_i \int \theta_d(t) - \theta_l(t)\,dt \qquad [22]$$

   is being considered. When in steady-state, the expression can be simplified to

   $$V_m(t) = k_p\,e_{ss} + k_i \int_0^{t_i} e_{ss}\,dt \qquad [23]$$

   .

   The variable t_i is the integration time. Find the integral gain when taking the error over two seconds

   $$t_i = 1.0\ [s] \qquad [24]$$

   and based on the proportional gain found in Section 4.2.2, the maximum SRV02 voltage [18], the steady-state error yielded with the PV control calculated in Section 4.3.1.

0	1	2

5. In-Lab Procedures

Students are asked to simulate the closed-loop PV and PIV responses. Then, the PV and PIV controllers are implemented on the actual SRV02.

5.1. Position Control Simulation

The *s_srv02_pos* Simulink diagram shown in Figure 7 is used to simulate the closed-loop position response of the SRV02 when using either the PV or PIV controls. The *SRV02 Model* uses a Transfer Fcn block from the Simulink library to simulate the SRV02 system. The *PIV Control* subsystem contains the PIV control detailed in Section 4.3.2. When the integral gain is set to zero, this is essentially a PV controller. Before going through the step response exercises in Section 5.1.2 and the ramp response questions in Section 5.1.3, go through Section 5.1.1 to configure the lab files properly.

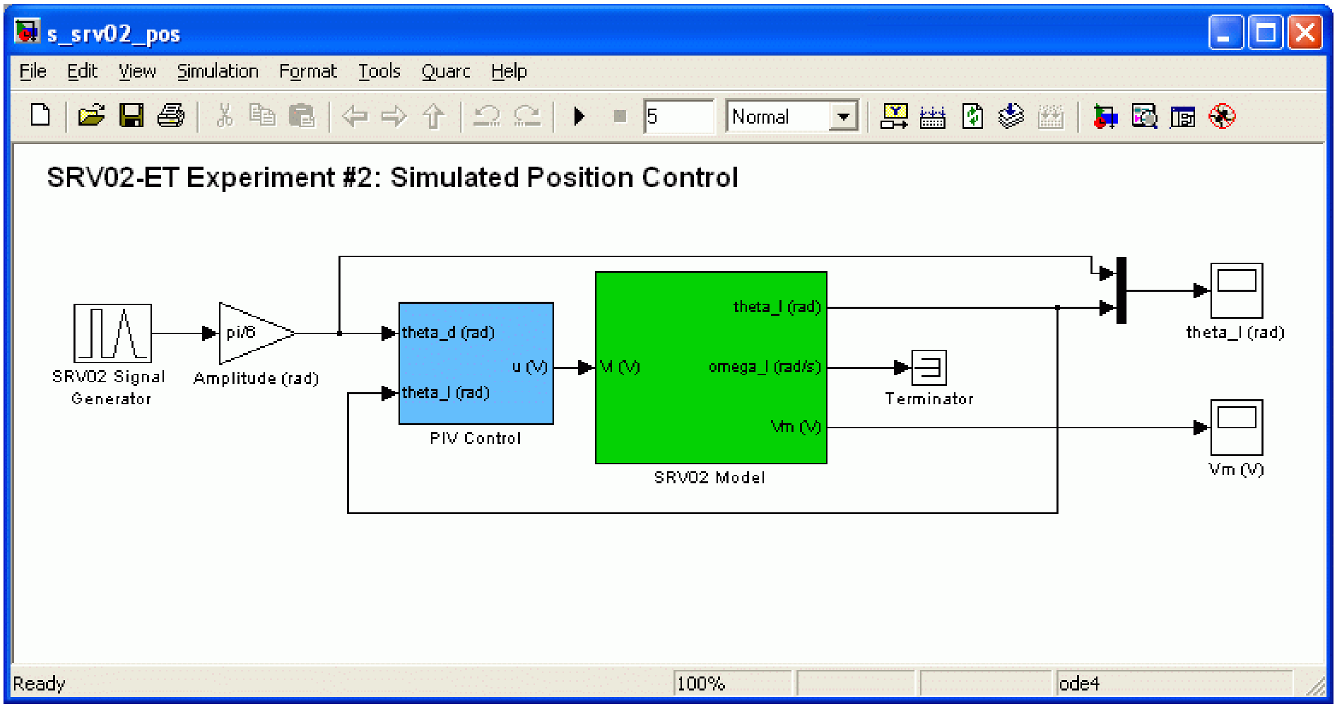

Figure 7: Simulink model used to simulate the SRV02 closed-loop position response.

5.1.1. Setup for Position Control Simulation

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 SRV02 position controller files, e.g. *q_srv02_pos.mdl*.
3. Double-click on the s*_srv02_pos.mdl* file to open the Simulink diagram shown in Figure 7.
4. Double-click on the *setup_srv02_exp02_spd.m* file to open the setup script for the position control Simulink models.
5. **Configure setup script**: The controllers will be ran on an SRV02 in the high-gear configuration with the disc load, as in Reference [7]. In order to simulate the SRV02 properly, make sure the script is setup to match this configuration, i.e. the EXT_GEAR_CONFIG should be set to 'HIGH' and the LOAD_TYPE should be set to 'DISC'. Also, ensure the ENCODER_TYPE, TACH_OPTION, K_CABLE, UPM_TYPE, and VMAX_DAC parameters are set according to the SRV02 system that is to be used in the laboratory. Next, set CONTROL_TYPE to 'MANUAL'.
6. 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 in Text 1, below, should be generated in the Matlab

Command Window. The model parameters and specifications are loaded but the PIV gains are all set to zero – they need to be changed.

```
SRV02 model parameters:
   K = 1.53 rad/s/V
   tau = 0.0254 s
Specifications:
   tp = 0.2 s
   PO = 5 %
Calculated PV control gains:
   kp = 0 V/rad
   kv = 0 V.s/rad
Integral control gain for triangle tracking:
   ki = 0 V/rad/s
```

Text 1: Display message shown in Matlab Command Window after running setup_srv02_exp02_pos.m.

5.1.2. Simulating Step Response using PV

The closed-loop step position response of the SRV02 will be simulated using the PV control to confirm that the specifications in an ideal situation are satisfied and verify that the motor is not saturated. In addition, the affect of using a high-pass filter instead of a direct derivative is examined.

Follow these steps to simulate the SRV02 PV position response:

1. Enter the proportional and velocity control gains found in Section 4.2.2 in Matlab.
2. Select *square* in the *Signal Type* field of the SRV02 Signal Generator in order to generate a step reference.
3. Set the *Amplitude (rad)* gain block to π/8 to generate a step with an amplitude of 45 degrees.
4. Inside the *PIV Control* subsystem, set the *Manual Switch* is to the upward position so the *Derivative* block is used.
5. Open the load shaft position scope, *theta_l (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 8 and 9. 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 *SRV02 Model* block). This simulation is the called the *Ideal PV* response as it uses the PV compensator with the derivative block.

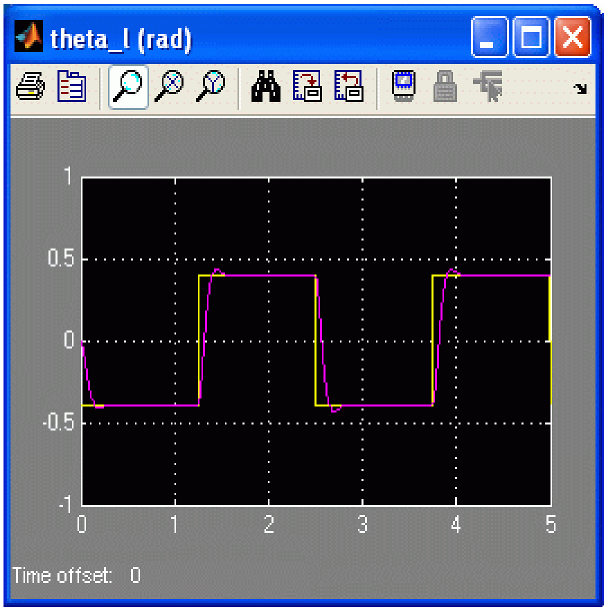

Figure 8: Ideal PV position response.

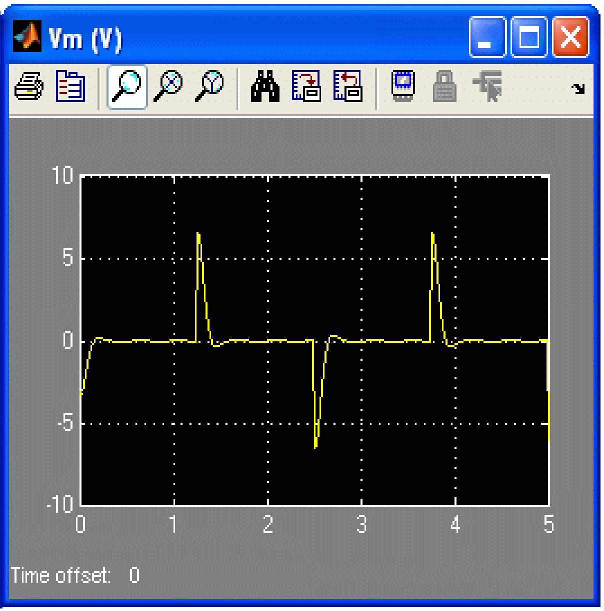

Figure 9: Ideal PV motor input voltage.

7. Generate a Matlab figure showing the *Ideal PV* position response and the ideal input voltage and attach it to your report. After each simulation run, each scope automatically saves their response to a variable in the Matlab workspace. That is, the *theta_l (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.

0	**1**	**2**

8. Measure the steady-state error, the percentage overshoot, and the peak time of the simulated response. Does the response satisfy the specifications given in Section 4.1.3?

0	1	2

9. When implementing a controller on actual hardware it is generally not advised to take the direct derivative of a measured signal. For instance, when taking the derivative of the SRV02 encoder or potentiometer signal any noise or spikes in the signal become amplified. This then gets multiplied by a gain and fed into the motor which may lead to damage. To remove any high-frequency 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 be tuned properly and does have some adverse affects. Go in the *PIV Control* block and set the Manual Switch block to the downward position in order to enable the high-pass filter.
10. Start the simulation. The response in the scopes should still be similar to figures 8 and 9. This simulation is the called the *Filtered PV* response as it uses the PV compensator with the high-pass filter block.
11. Generate a Matlab figure showing the *Filtered PV* position and input voltage responses and attach it to your report.

0	1	2

12. Measure the steady-state error, peak time, and percentage overshoot. Are the specifications still satisfied without saturating the actuator? Recall that the peak time and percentage overshoot should not exceed the values given in Section 4.1.3. Discuss the changes that were noticed from the ideal response.

0	**1**	**2**

13. If the specifications in the *ideal* case are satisfied without overloading the servo motor, proceed to the next section to simulate the ramp response.

5.1.3. Simulating Ramp Response using PV and PIV

The closed-loop ramp position response of the SRV02 will be simulated using the PV and PIV control systems. It will be verified that the zero steady-state error specifications in Section 4.1.3 is met without the motor becoming saturated.

Follow these steps to simulated the SRV02 PV position response:

1. Enter the proportional and velocity control gains found in Section 4.2.2.
2. Select *triangle* in the *Signal Type* field of the SRV02 Signal Generator in order to generate a ramp or triangular reference.
3. Set the *Amplitude (rad)* gain block to π/3.
4. Ensure the *Frequency (Hz)* field in the *Triangle Waveform* block is set to 0.8 Hz. This will generate a alternative increasing and decreasing ramp signal with the same slope used in the pre-lab exercises, given in Equation [20]. The slope is calculated from the *Triangular Waveform* amplitude, *Amp*, and frequency, *f*, using the expression

$$R_0 = 4\, Amp\, f \quad . \qquad [25]$$

5. Inside the *PIV Control* subsystem, set the *Manual Switch* is to the downward position so the *High-Pass Filter* block is used.

6. Open the load shaft position scope, *theta_l (rad)*, and the motor input voltage scope, *Vm (V)*.
7. Start the simulation. The scopes should be displaying responses similar to figures 10 and 11.

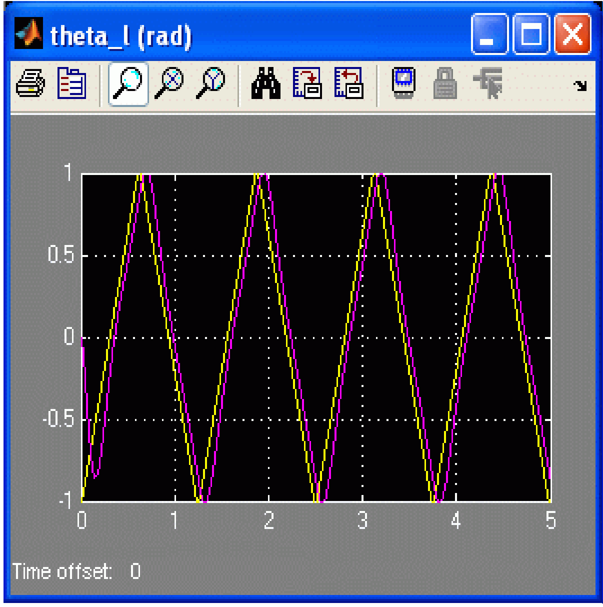

Figure 10: Ramp response using PV.

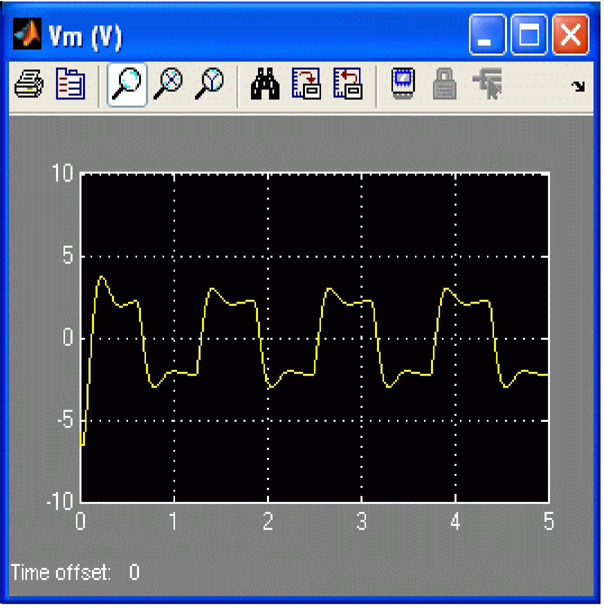

Figure 11: Input voltage of ramp tracking using PV.

8. Generate a Matlab figure showing the *Ramp PV* position response and its corresponding input voltage trace. Attach it to your report.

9. Measure the steady-state error. Compare the simulation measurement with the steady-state error calculated in Pre-Lab Section 4.3.1.

0	**1**	**2**

0	1	2

10. Enter the integral gain designed in Section 4.3.4 in Matlab. The PIV controller will now be simulated.
11. Start the simulation. The *Ramp PIV* response in the scopes should be similar to figures 12 and 13.

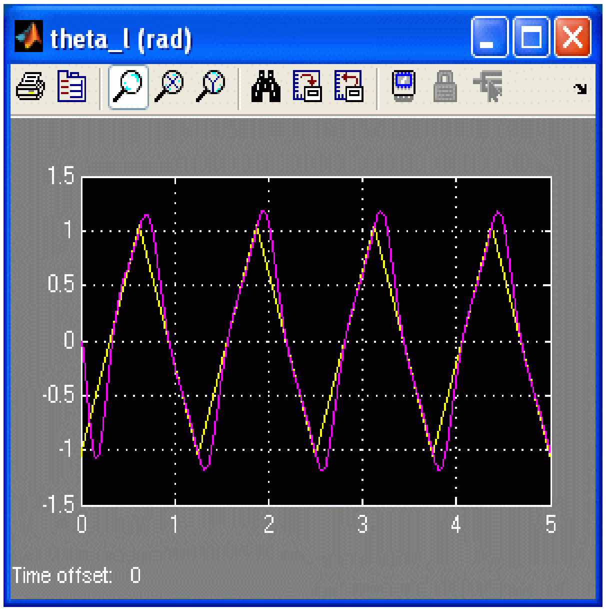

Figure 12: PIV ramp response.

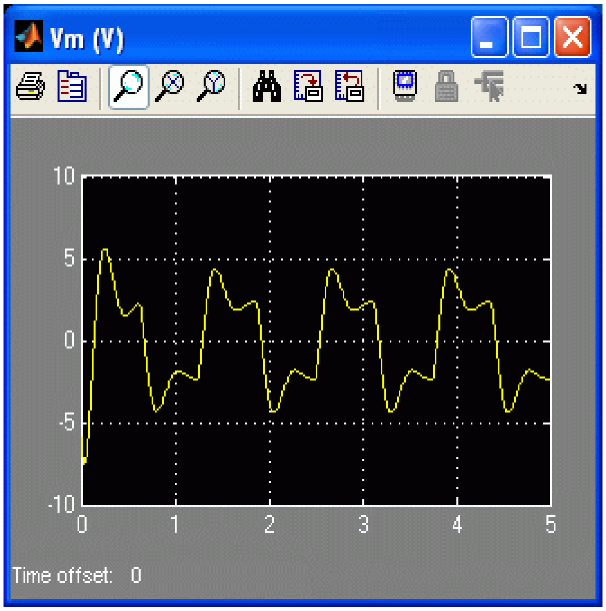

Figure 13: Input voltage using PIV control.

12. Generate a Matlab figure showing the *Ramp PIV* position response and its corresponding input voltage. Attach it to your report.

0	**1**	**2**

13. Measure the steady-state error. Is the steady-state specification given in Section 4.1.3 satisfied without saturating the actuator?

0	1	2

14. If the specifications are satisfied without saturating the actuator, proceed to the next section to begin implementing the controllers on the SRV02.

5.2. Position Control Implementation

The *q_srv02_pos* Simulink diagram shown in Figure 14 is used to perform the position control exercises in this laboratory. The *SRV02-ET* subsystem contains QuaRC blocks that interface with the DC motor and sensors of the SRV02 system, as discussed in Reference [6]. The *PIV Control* subsystem implements the PIV control detailed in Section 4.3.2, except a high-pass filter is used to obtain the velocity signal (as opposed to taking the direct derivative).

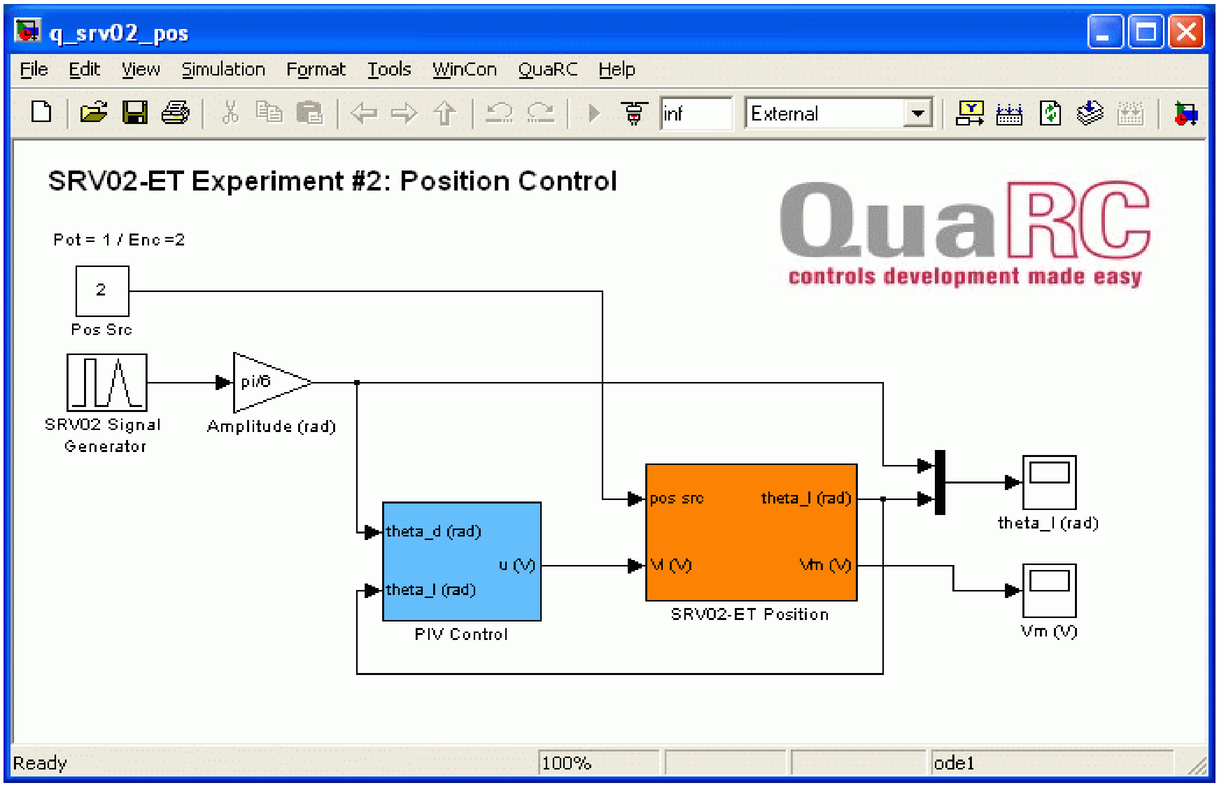

Figure 14: Simulink model used with QuaRC to run the PV and PIV position controllers on the SRV02.

5.2.1. Setup for Position Control Implementation

Before beginning the in-lab exercises on the SRV02 device, the q_srv02_pos Simulink diagram and the setup_srv02_exp02_pos.m script must be configured.

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

1. Setup the SRV02 in the high-gear configuration and with the disc load as described in Reference [5].
2. Load the Matlab software.
3. Browse through the *Current Directory* window in Matlab and find the folder that contains the SRV02 position control files, e.g. *q_srv02_pos.mdl*.
4. Double-click on the *q_srv02_pos.mdl* file to open the Position Control Simulink diagram shown in Figure 14.
5. **Configure DAQ**: Double-click on the HIL Initialize block in the *SRV02-ET* subsystem (which is located inside the *SRV02-ET Position* subsystem) and ensure it is configured for the DAQ device that is installed in your system. For instance, the block shown in Figure 14 is setup for the Quanser Q8 hardware-in-the-loop board. See Reference [6] for more information on configuring the HIL Initialize block.
6. **Configure Sensor**: The position of the load shaft can be measured using various sensors. Set the *Pos Src* Source block in q_srv02_pos, as shown in Figure 14, as follows:
   - 1 to use the potentiometer

- 2 to use to the encoder

Note that when using the potentiometer, there will be a discontinuity.

7. **Configure setup script**: Set the parameters in the setup_srv02_exp02_pos.m script according to your system setup. See Section 5.1.1 for more details.

5.2.2. Implementing Step Response using PV

In this lab, the angular position of the SRV02 load shaft, i.e. the disc load, will be controlled using the developed PV control. Measurements will then be taken to ensure that the specifications are satisfied.

Follow the steps below:

1. Run the setup_srv02_exp02_pos.m script.
2. Enter the proportional and velocity control gains found in Section 4.2.2.
3. Set *Signal Type* in the SRV02 Signal Generator to *square* to generate a step reference.
4. Set the *Amplitude (rad)* gain block to π/8 to generate a step with an amplitude of 45 degrees.
5. Open the load shaft position scope, *theta_l (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 be displaying responses similar to figures 15 and 16. Note that in the *theta_l (rad)* scope, the yellow trace is the setpoint position while the purple trace is the measured position.

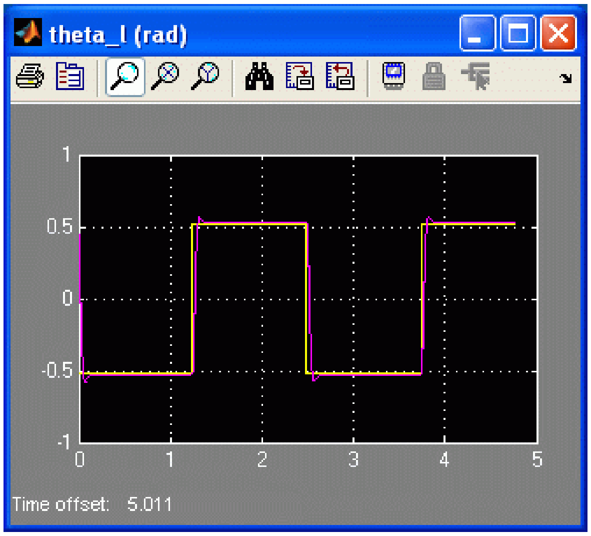

Figure 15: Measured PV step response.

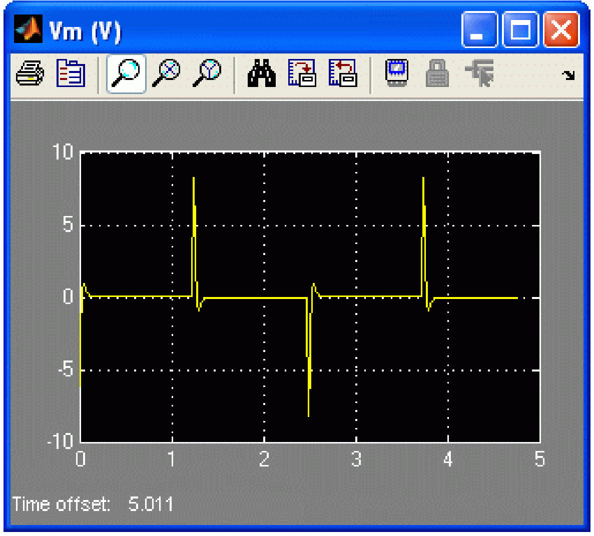

Figure 16: PV control input voltage.

8. When a suitable response is obtained, click on the *Stop* button in the Simulink diagram tool bar (or select QuaRC | Stop from the menu) to stop running the code. Generate a Matlab figure showing the PV position response and its input voltage. Attach it to your report. As in the s_srv02_pos 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.

0	1	2

9. Measure the steady-state error, the percentage overshoot, and the peak time of the SRV02 load gear. Does the response satisfy the specifications given in Section 4.1.3?

0	1	2

10. Make sure QuaRC is stopped.
11. Shut off the power of the UPM if no more experiments will be performed on the SRV02 in this session.

5.2.3. Implementing Ramp Response using PV and PIV

Follow these steps to make the SRV02 track a triangular response using the PV and PIV controls:

1. Run the setup_srv02_exp02_pos.m script.
2. Enter the proportional and velocity control gains found in Section 4.2.2.
3. Set *Signal Type* in the SRV02 Signal Generator to *triangle* in order to generate a ramp or triangular reference.
4. Set the *Amplitude (rad)* gain block to π/3.
5. Ensure the *Frequency (Hz)* field in the *Triangle Waveform* block is set to 0.8 Hz.
6. Open the load shaft position scope, *theta_l (rad)*, and the motor input voltage scope, *Vm (V)*.
7. Click on QuaRC | Build to compile the Simulink diagram.
8. Select QuaRC | Start to begin running the controller. The scopes should be displaying responses similar to figures 18 and 17.

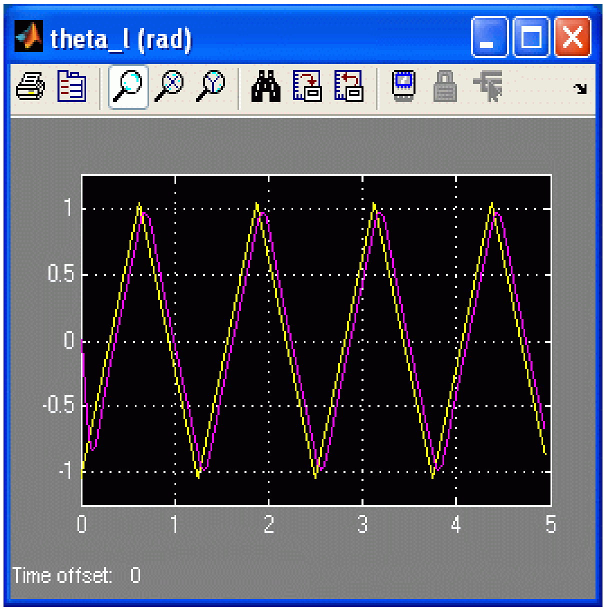

Figure 17: Measured SRV02 PV ramp response.

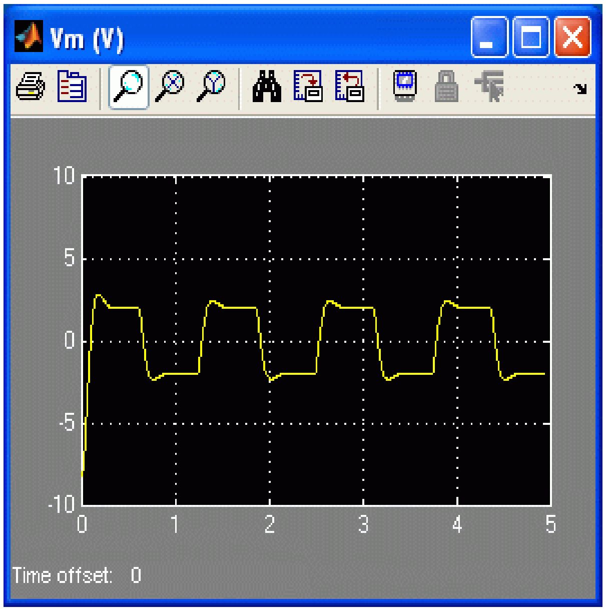

Figure 18: Input voltage of PV ramp response.

9. Generate a Matlab figure showing the *Ramp PV* position response and its corresponding input voltage trace. Attach it to your report.

10. Measure the steady-state error and compare it with the steady-state error calculated in Pre-Lab Section 4.3.1.

0	**1**	**2**

0	1	2

11. Enter the integral gain designed in Section 4.3.4 in Matlab in order to run the PIV controller on the SRV02.
12. Start the QuaRC controller. The *Ramp PIV* response in the scopes should be similar to figures 19 and 20.

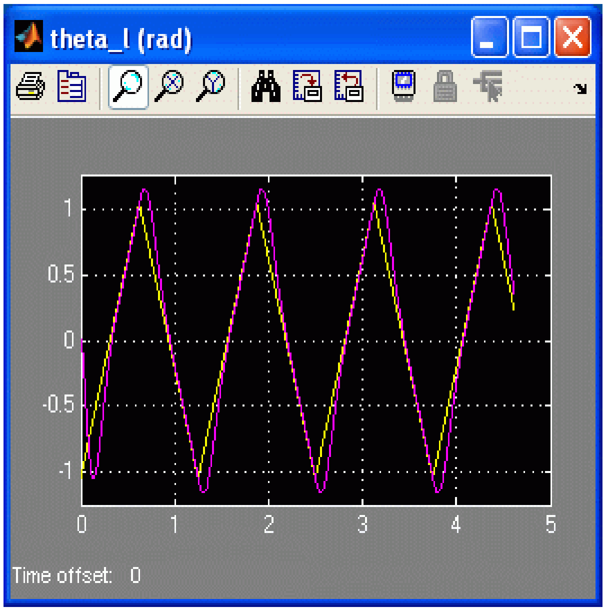

Figure 19: Measured SRV02 PIV ramp response.

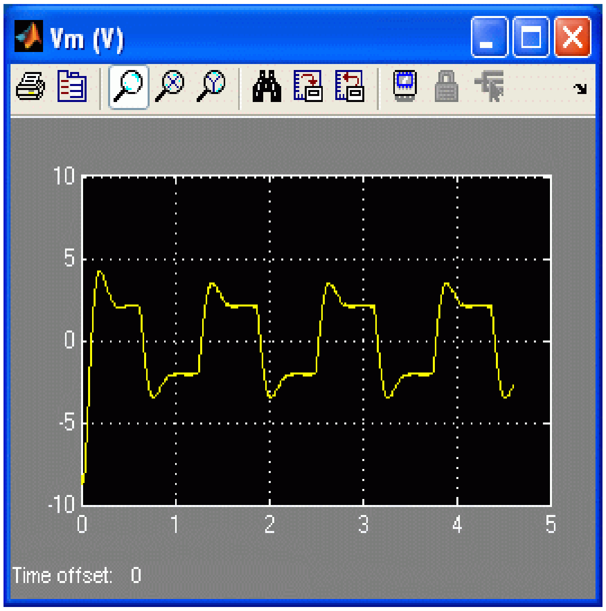

Figure 20: Input voltage from PIV ramp control.

13. Generate a Matlab figure showing the measured SRV02 *Ramp PIV* position response and its corresponding input voltage. Attach it to your report.

14. Measure the steady-state error. Is the steady-state specification given in Section 4.1.3 satisfied without saturating the actuator?

0	**1**	**2**

0	1	2

15. Make sure QuaRC is stopped.
16. Shut off the power of the UPM if no more experiments will be performed on the SRV02 in this session.

6. Results Summary

Fill out Table 2, below, with the pre-laboratory and in-laboratory results obtained such as the designed PIV gains along with the measured peak time, percentage overshoot, and steady-state error obtained from the simulated and implemented step and ramp responses.

Section	*Description*	*Symbol*	*Value*	*Unit*
4.2.2	***Pre-Lab: Model Parameters***			
4.	Open-Loop Steady-State Gain	K		rad/(V.s)
4.	Open-Loop Time Constant	τ		s
4.2.2	***Pre-Lab: PV Gain Design***			
4.	Proportional gain	k_p		V/rad
4.	Velocity gain	k_v		V.s/rad
4.2.3	***Pre-Lab: Control Gain Limits***			
1.	Maximum proportional gain	$k_{p,max}$		V/rad
4.3.1	***Pre-Lab: Ramp Steady-State Error***			
2.	Steady-state error using PV	e_{ss}		rad
4.3.3	***Pre-Lab: Ramp Steady-State Error***			
2.	Steady-state error using PIV	e_{ss}		rad
4.3.4	***Pre-Lab: Integral Gain Design***			
1.	Integral gain	k_i		V/(rad.s)
5.1.2	***In-Lab Simulation: Ideal PV Step Response***			
8.	Peak time	t_p		s
8.	Percentage overshoot	PO		%
8.	Steady-state error	e_{ss}		rad
5.1.2	***In-Lab Simulation: Filtered PV Step Response***			
12.	Peak time	t_p		s
12.	Percentage overshoot	PO		%
12.	Steady-state error	e_{ss}		rad
5.1.3	***In-Lab Simulation: Ramp Response***			

9.	PV Steady-state error	e_{ss}		rad
13.	PIV Steady-state error	e_{ss}		rad
5.2.2	***In-Lab Implementation: PV Step Response***			
9.	Peak time	t_p		s
9.	Percentage overshoot	PO		%
9.	Steady-state error	e_{ss}		rad
5.2.3	***In-Lab Implementation: Ramp Response***			
9.	PV Steady-state error	e_{ss}		rad
13.	PIV Steady-state error	e_{ss}		rad

Table 2: SRV02 Experiment #2: Position control results summary.

7. References

[1] Quanser. *DAQ User Manual*.
[2] Quanser. *QuaRC User Manual* (type `doc quarc` in Matlab to access).
[3] Quanser. *QuaRC Installation Manual*.
[4] Quanser. *UPM User Manual*.
[5] Quanser. *SRV02 User Manual*.
[6] Quanser. *SRV02 QuaRC Integration – Instructor Manual*.
[7] Quanser. *Rotary Experiment #1: SRV02 Modeling*.