QUANSER
INNOVATE · EDUCATE

Lab Procedure
Proportional Control

Introduction

Ensure the following:

1. You have reviewed the Application Guide – Proportional Control
2. The Qube-Servo 3 has been previously tested, is ON and connected to the PC.
3. Inertia disc load is attached to the Qube-Servo 3.
4. Launch MATLAB and browse to the working directory that includes the Simulink models for this lab.

This experiment consists of three parts that will help you understand how to control your DC motor. First, you'll explore second order systems by calculating and measuring a step response input when trying to control the motor's position. The second and third steps will be using a proportional controller to control both position and speed of the Qube and looking at the differences between the calculations and measurements of the real system.

The **Hardware Interfacing** and **Filtering** labs explained the basic blocks to read and write from the Qube-Servo 3. For simplicity, all labs forward will use a Qube-Servo 3 block that sets up the system beforehand and outputs the available information from the Qube.

Second Order Systems

Calculating the System Response

1. Use the closed-loop equation for the Qube-Servo 3 under unity feedback (equation below) and find the natural frequency (ω_n) and damping ratio (ζ) of the system in terms of K and τ.

$$\frac{\Theta_d(s)}{V_m(s)} = \frac{\frac{K}{\tau}}{s^2 + \frac{1}{\tau}s + \frac{K}{\tau}}$$

2. Using the previous answer, solve for the natural frequency (ω_n) and damping ratio (ζ) using the model parameters K and τ calculated in the modeling labs, or use $K = 24$ and $\tau = 0.1$ as defaults if you have not done any of those labs.
3. Calculate the expected peak time and percent overshoot based on the calculated natural frequency (ω_n) and damping ratio (ζ).

Measuring the System Response

Using the gains found to convert encoder counts into rads from the instrumentation labs, use the qs3_proportional_position.slx file to design a model that applies a step as the desired position of 5 rad at second 1 and implements unity feedback position control using the servo position using the encoder, as shown in Figure 1. The unconnected blocks will be used later.

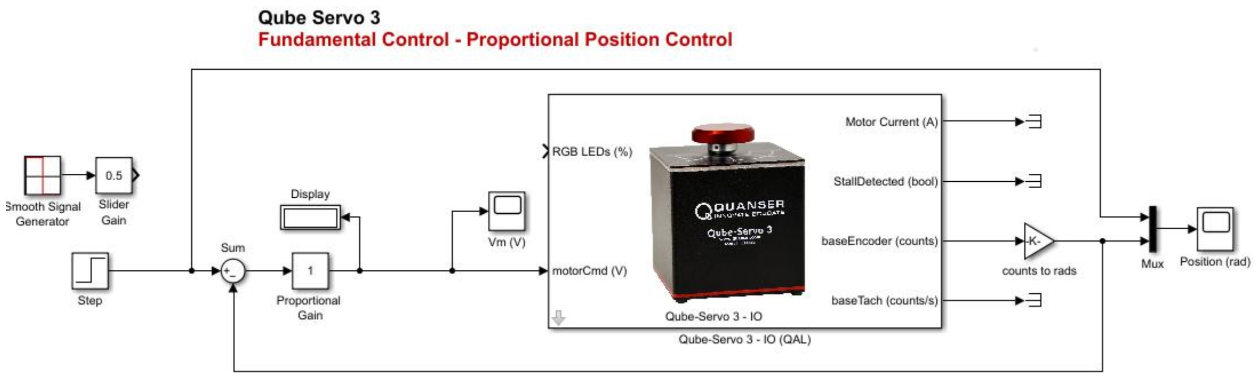
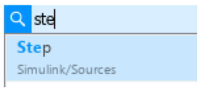


Figure 1: Unity feedback position control for the Qube-Servo 3

4. Add a `Step` and a `Smooth Signal Generator` block into your model, by double clicking on an empty part of your model and typing the block name. Configure the `Step` to be a 5V step that starts at a time of 1 second.

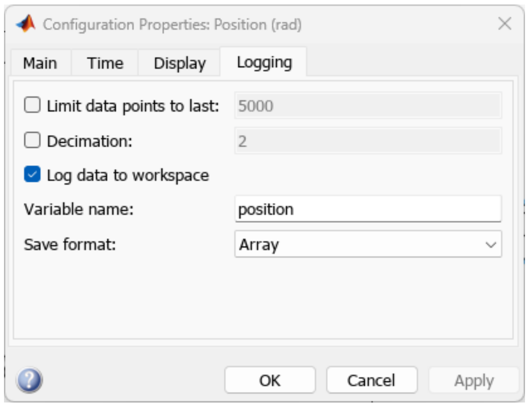


5. Use the gain value found in previous labs to convert the encoder counts to radians on the gain connected to the Encoder output. Connect all the blocks as shown in Figure 1.

6. In your Simulink model, go to the `Hardware` tab, and change the Stop Time to 2.5 seconds.

7. Build and run the QUARC controller using the `Monitor & Tune` button on the `Hardware` or `QUARC` tab. The model will run for 2.5 seconds and then stop.

8. Confirm that the scopes are configured to save the output information on your MATLAB workspace. Open the scopes and click the configuration/gear icon on the top left side of the window, right below `File`. The logging tab in the window should look like this for the position scope. The variable name should be **voltage** for the Voltage scope.


Configuration Properties: Position (rad)

Main | Time | Display | Logging

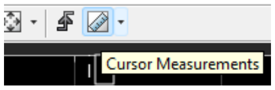
- [ ] Limit data points to last: 5000
- [ ] Decimation: 2
- [x] Log data to workspace

Variable name: position

Save format: Array

OK Cancel Apply

9. Measure the peak time and percent overshoot and record the results. Alternatively, used the saved data in the MATLAB workspace for analyzing later.
   Hint: Use the `Cursor Measurements` tool in the Simulink Scope to take measurements directly from the response plots.



10. Take a screenshot of the response in the scopes. It should look similar to the response below in Figure 2 but it should only have been run for 2.5 seconds.

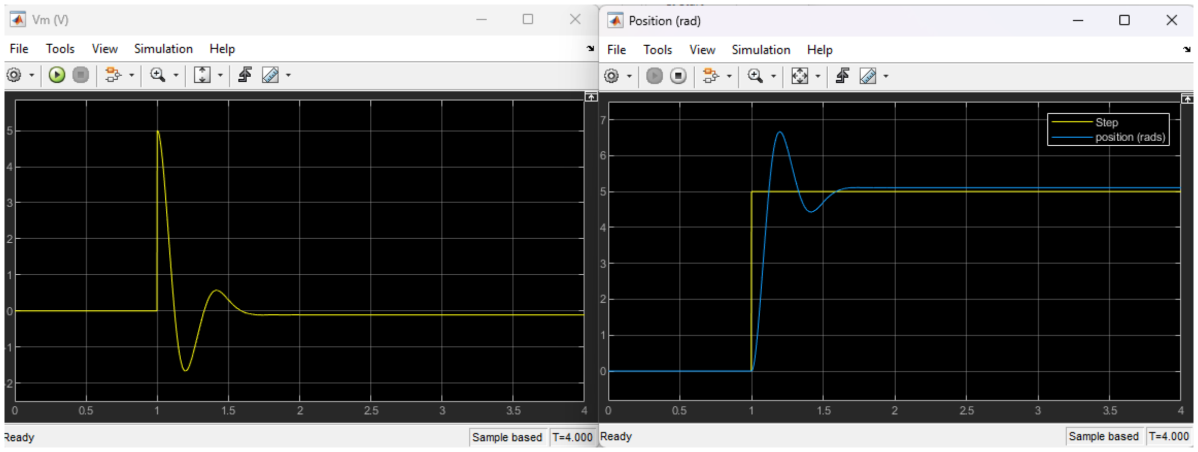


Figure 2: Example unity feedback Qube-Servo 3 step response.

Proportional Position Control

11. Is it possible to get any desired second-order system response (by specifying natural frequency (ω_n) and damping (ζ)) by only using proportional feedback gain?
Hint: Compare the Qube's closed loop transfer function to the standard second order system transfer function.

$$\frac{Y(s)}{R(s)} = \frac{K\,k_p}{\tau s^2 + s + K\,k_p} \qquad \frac{Y(s)}{R(s)} = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}$$

Use the same qs3_proportional_position.slx file from the previous section. Disconnect (shift+click and drag) the `Step` input and replace it with the `smooth signal generator` and the `slider gain`.

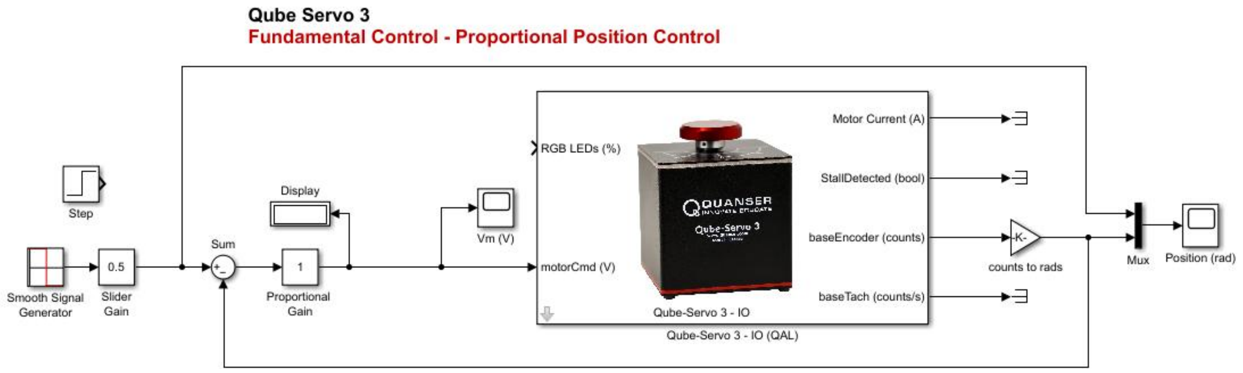


Figure 3: Unity feedback position control for the Qube-Servo 3

12. Add a `Step` block into your model, by double clicking on an empty part of your model and typing the block name. Configure the step to be a 5V step that starts at a time of 1 second.

13. Configure the `Smooth Signal Generator` block to output a square wave with amplitude of 1 and frequency of 0.4 Hz. This will make the servo command or reference angle to be the square wave of 0.5 rads because of the `Slider Gain`.

14. The voltage that can be applied to the Qube – Servo 3 is limited to ±10V. Determine the maximum proportional gain for the square wave reference signal of ±0.5 rad above that does not saturate the Qube – Servo 3.

15. In your Simulink model, go to the `Hardware` tab, and change the Stop Time to `inf`.

16. Build and run the QUARC controller using the `Monitor & Tune` button on the `Hardware` or `QUARC` tab.

17. Run the controller with a **proportional gain of 1.5**. Save the position response and motor voltage scope responses. With a k_p of 1, it should look like Figure 4.

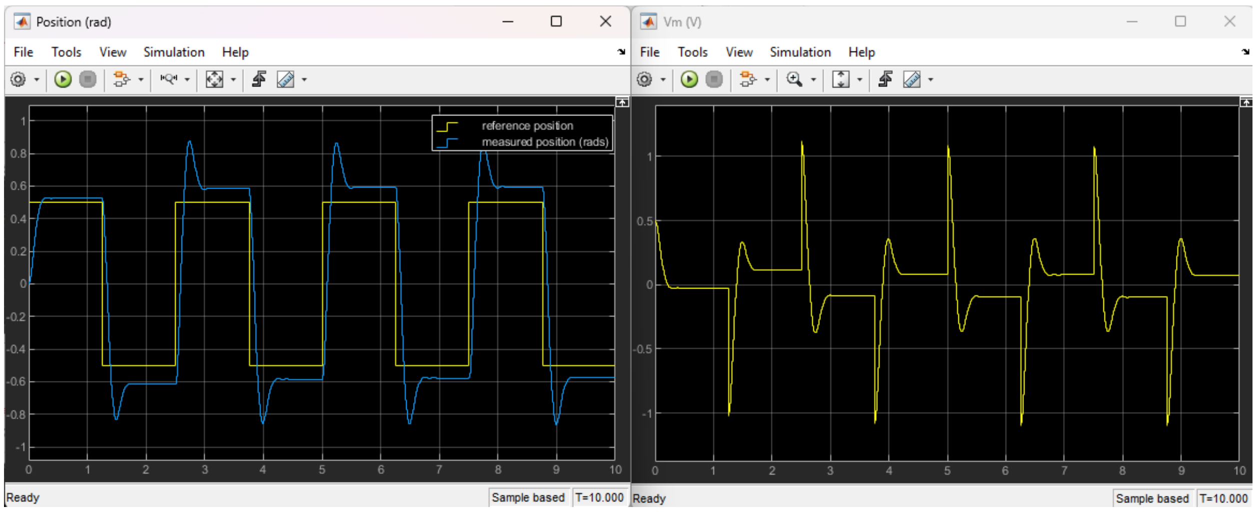


Figure 4: Proportional Position Control response with $k_p = 1$

18. Measure the percent overshoot and peak time of the response when $k_p = 1.5$.

19. Is there a steady-state error? If so, what is it?

20. Vary k_p between 1 and 5. How does the proportional gain affect the servo position control response?

21. Vary k_p between 0.1 and 1. What happens when k_p is decreased?

22. Describe the response for very small gains of k_p (i.e. below 0.5). Does the system respond as expected? Explain.
Hint: If the system's response does not meet your expectations, verify that the desired control signals are applied to the Qube – Servo 3 by looking at the display.

23. Stop and close your model. Ensure you save a copy of the files for review later.

Proportional Speed Control

A proportional speed control for the system is shown in Figure 5.

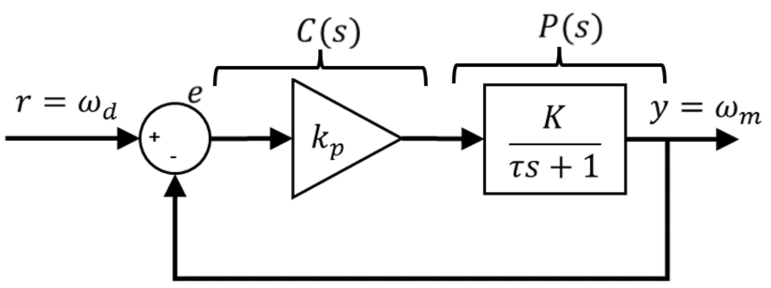


Figure 5: Proportional control of servo speed

24. Find the closed-loop transfer function $E(s)/R(s)$ in that represents the dynamics between a desired speed, $R(s) = \Omega_d(s)$, and the error, $E(s) = R(s) - Y(s) = \Omega_d(s) - \Omega_m(s)$

25. Find the steady-state error of the system, e_{ss}, for the reference step input $R(s) = R_0/s$ where R_0 the desired angular rate step amplitude.
Hint: Use the Final-Value Theorem described in the concept review

26. Evaluate the steady-state error if the proportional gain is 0.5 V*s / rad and the system is given a step amplitude of 15 rad/s. Use the K found in one of the modeling laboratories, e.g. Step Response modeling lab.
NOTE: If no modeling lab has been done, use, $K = 24$.

27. Open the qs3_proportional_speed.slx file, as shown in Figure 6. Update the gains to convert tach counts/s into rads/s from the instrumentation labs.

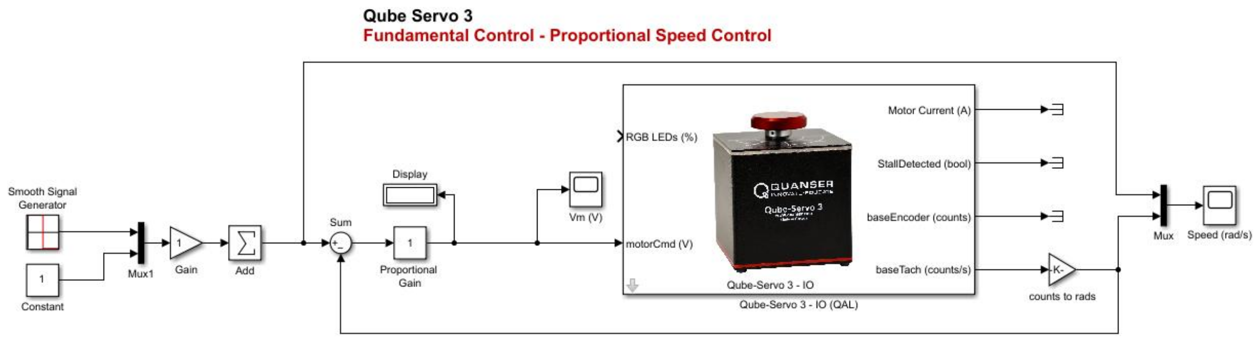


Figure 6: Unity feedback position control for the Qube-Servo 3

28. Add a `Smooth Signal Generator` block and configure it to output a square wave with amplitude of 1 and frequency of 0.2 Hz. Connected as shown in Figure 6. The `Gain` and `Add` blocks make sure that your signal after the `Add` will be a square wave from 0 to 2*`Gain`.

29. Generate a reference speed command with an amplitude of 15 rad/s by setting the Gain to 7.5 rad/s. Make sure the `Smooth Signal Generator` block is configured to a frequency of 0.2 Hz.

30. Build and run the QUARC controller using the `Monitor & Tune` button on the `Hardware` or `QUARC` tab.

31. Run the controller with a proportional gain of 0.5. Save the speed response and motor voltage scope responses. With a k_p of 0.7, it looks like Figure 4.

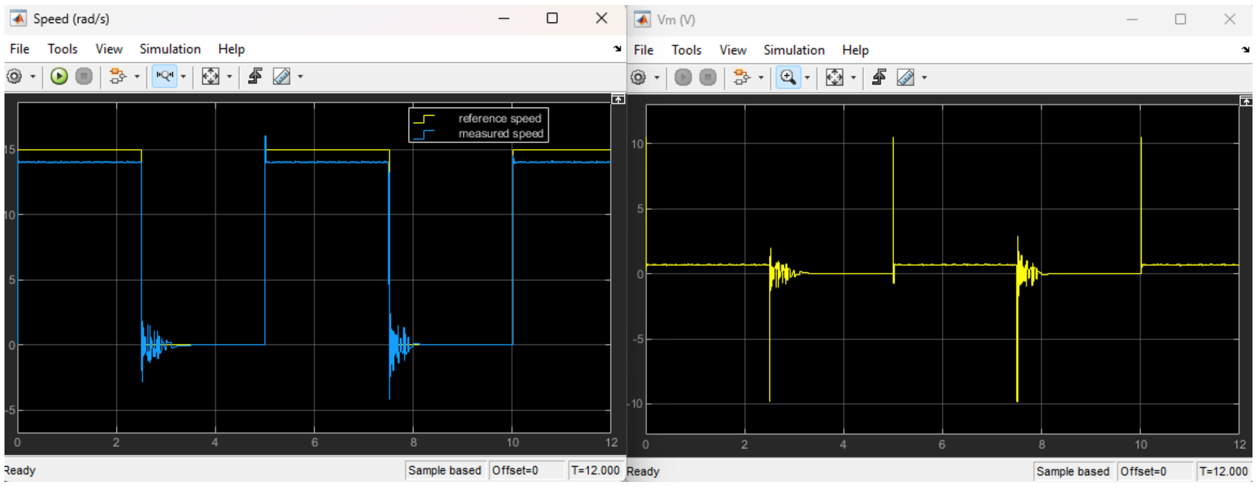

Speed (rad/s)
File Tools View Simulation Help
reference speed
measured speed
15
10
5
0
-5
0 2 4 6 8 10 12
Ready
Sample based Offset=0 T=12.000
Vm (V)
File Tools View Simulation Help
10
5
0
-5
-10
0 2 4 6 8 10 12
Ready
Sample based Offset=0 T=12.000

Figure 7: Proportional Speed Control response with $k_p = 0.7$

32. Measure the steady-state error. How does it compare with your value computed above?
Hint: Use the Cursor Measurement tool in the Simulink Scope to take your measurements.

33. Show how the error can be decreased by half the current magnitude. Validate your results with the Qube - Servo 3 and show your response.
Hint: Recall how the steady state error was defined and identify which parameters you can adjust.

34. Stop and close your model. Ensure you save a copy of the files for review later.

35. Power OFF the Qube-Servo 3.