

DC Motor Velocity Control

ME 207 - Mechanical Engineering Laboratory III

March 18, 2024

Contents

1	Introduction and Goals	2
2	Modeling	2
3	PID Control	7
4	Velocity Control	8
5	Position Control	9

1 Introduction and Goals

In this experiment you will work on the Qube-Servo 3 by Quanser (Fig. 1). The goal of this experiment is to control both the velocity and the position of the inertial disc module. The successful completion of this experiment requires the execution of the following tasks:

- Obtain the differential equation governing the dynamics of the inertial disc.
- Calculate transfer functions between applied voltage and velocity/position of the disc.
- Design a proportional controller for both velocity and position regulation.
- Design a proportional-derivative controller for position regulation.

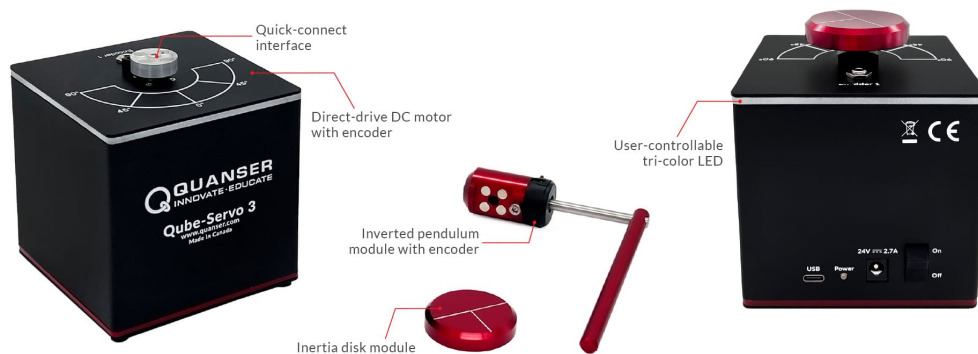


Figure 1: Qube-Servo 3

All the necessary files to carry out the experiment are available in the Google Drive folder shared with you. These files needed to design and test the controllers are in two subfolders:

- Design: These files are intended to be used “off-line” before or between sessions.
- Test: These files are intended to be used to run the experiment (Qube-Servo 3 must be connected to the computer). To run any of these SIMULINK[®] files, navigate to the “Hardware” tab in the top menu bar and select “Monitor and Tune.”

Lab Work 1:

1. Get familiar with Qube-Servo 3 and the provided MATLAB[®] /SIMULINK[®] files.

2 Modeling

The Quanser Qube-Servo 3 is a direct-drive rotary servo system. Its motor armature circuit schematic is shown in Fig. 2 and the electrical and mechanical parameters are given in

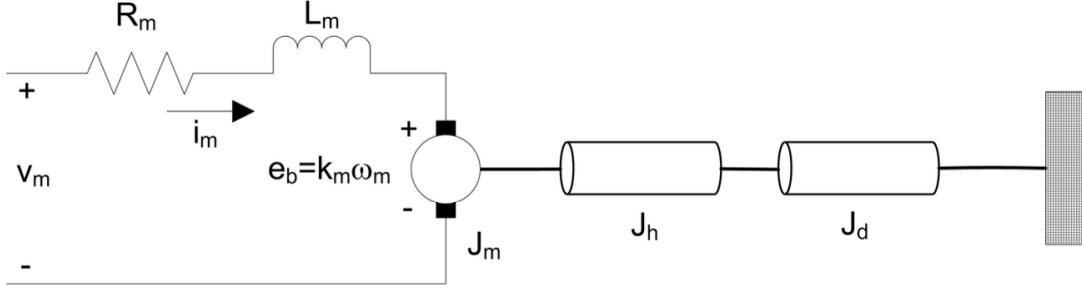


Figure 2: Motor

Table 1. The DC motor shaft is connected to the load hub. The hub is a metal disk used to mount the disk or rotary pendulum and has a moment of inertia of J_h . A disk load is attached to the output shaft with a moment of inertia of J_d .

The back-emf (electromotive) voltage $e_b(t)$ depends on the speed of the motor shaft, ω_m , and the back-emf constant of the motor, k_m . It opposes the current flow. The back-emf voltage is given by:

$$e_b(t) = k_m \omega_m(t) \quad (1)$$

Using Kirchoff's Voltage Law, we can write the following equation:

$$v_m(t) - R_m i_m(t) - L_m \frac{di_m(t)}{dt} - e_b(t) = 0.$$

Since the motor inductance L_m is much less than its resistance, it can be ignored. Then, the equation becomes

$$v_m(t) - R_m i_m(t) - e_b(t) = 0. \quad (2)$$

The motor shaft equation is expressed as

$$J_{eq} \dot{\omega}_m(t) = \tau_m(t), \quad (3)$$

where J_{eq} is total moment of inertia acting on the motor shaft and τ_m is the applied torque from the DC motor. Based on the current applied, the torque is

$$\tau_m = k_t i_m(t). \quad (4)$$

The moment of inertia of a disk about its pivot, with mass m_d and radius r_d , is

$$J = \frac{1}{2} m r^2. \quad (5)$$

Lab Work 2:

1. Formulate the differential equation for ω_m using (1), (2), (3) and (4). Show that this equation can be written as

$$\dot{\omega}_m(t) = \frac{k_t}{J_{eq} R_m} v_m(t) - \frac{k_t k_m}{J_{eq} R_m} \omega_m(t) \quad (6)$$

Symbol	Description	Value/Units
R_m	Motor terminal resistance	8.4 Ω (ohms)
L_m	Rotor inductance	1.16 mH
k_t	Motor torque constant	0.042 Nm/A (newton-meters per amp)
k_m	Motor back-electromotive force constant	0.042 V/(rad/s)
J_m	Rotor moment of inertia	4×10^{-6} kgm ²
m_h	Load hub mass	0.0106 kg
r_h	Load hub radius	0.0111 m
m_d	Disk mass	0.053 kg
r_d	Disk radius	0.0248 m
v_m	Motor terminal voltage	V (Volts)
i_m	Motor current	A (amps)
ω_m	Motor shaft angular velocity	rad/s
τ_m	Torque produced by the motor	Nm
J_{eq}	Equivalent moment of inertia	kgm ²

Table 1: Motor parameters.

- The motor shaft of the Qube-Servo 3 is attached to a load hub and a disk load. Based on the parameters in Table 1, calculate the equivalent moment of inertia $J_{eq} \triangleq J_m + J_h + J_d$ acting on the motor shaft by using (5). Implement this calculation in the MATLAB[®] file `MEM05_Qube_DCMotor_Parameters_Std.m` provided in the “Design” directory.
- The SIMULINK[®] file `MEM05_Qube_DCMotor_OLResponse.slx` provided in the “Design” directory will allow you to simulate the response of both velocity and position to the imposed voltage based on the model (6). The provided file generates the input voltage by combining Signal Generator and Constant blocks but you can use the input you want (explore “Sources” in the Library Browser). Besides showing the simulated inputs and outputs in Scope blocks, SIMULINK[®] saves all the signals in both the workspace and “mat” files. The MATLAB[®] file `MEM05_Qube_DCMotor_Parameters_Std.m` file must be run before running the SIMULINK[®] model to load in the workspace all the variables that are needed by the model. The MATLAB[®] file `MEM05_Qube_DCMotor_Plots.m` in the same directory shows you how to plot all the signals both from the workspace and the “mat” files in both individual and combined plots. Simulate the response of the Qube-Servo 3 to a sinusoidal voltage input with frequency of 1 Hz.

NOTE: The file `MEM05_Qube_DCMotor_Plots.m` shows you how to control font sizes, titles, labels, legends, and lines in a MATLAB[®] plot. All the plots in your report should follow this standard. Fig. 3 shows the type of plot expected in your report. Screen captures of the SIMULINK[®] scopes should not be used in the report.

- Take the Laplace Transform of (6) and find the voltage to speed transfer function, $\Omega_m(s)/V_m(s)$, of the system. Show that the transfer function can be written as

$$\frac{\Omega_m(s)}{V_m(s)} = \frac{k}{\tau s + 1} \quad (7)$$

with $k = 23.8$ and $\tau = 0.1$, where $\Omega_m(s) \triangleq \mathcal{L}\{\omega_m(t)\}$ and $V_m(s) \triangleq \mathcal{L}\{v_m(t)\}$.

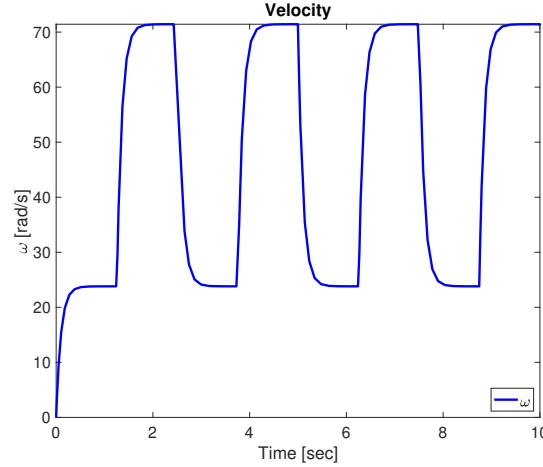


Figure 3: Example plot for the Qube-Servo 3 velocity as simulated by the SIMULINK[®] file MEM05_Qube_DCMotor_OLResponse.slx. This plot has been generated by the MATLAB[®] file MEM05_Qube_DCMotor_Plots.m.

5. Show that the voltage to position transfer function, $\Theta_m(s)/V_m(s)$, can be written as

$$\frac{\Theta_m(s)}{V_m(s)} = \frac{k}{s(\tau s + 1)} \quad (8)$$

where $\Theta_m(s) \triangleq \mathcal{L}\{\theta_m(t)\}$ and $\omega_m(t) = d\theta_m(t)/dt$.

6. The SIMULINK[®] file q_qube3_model_student.slx provided in the “Test” directory will allow you to compare the simulated response by the model with the actual response of the Qube-Servo 3. Before being able to run this file you need to implement model (6) in the “QUBE-Servo Model” subsystem as shown in Fig 4.

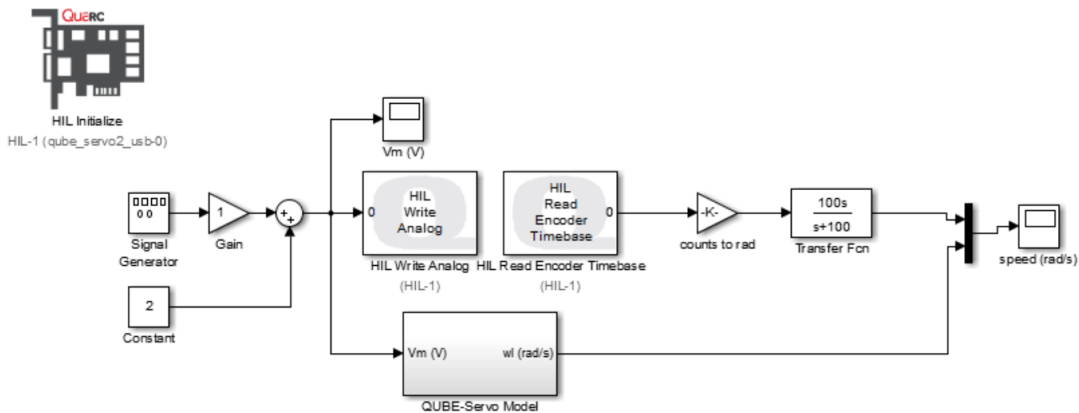


Figure 4: Qube-Servo 3 schematic for model validation. Applies an input voltage and displays measured and simulated Qube-Servo-3 speed. The “QUBE-Servo Model” subsystem must be configured first. The qube_servo3_usb board must be selected in the HIL Initialize block.

This subsystem block in SIMULINK[®] will need Gain blocks, a Subtract block, and an Integrator block (to go from acceleration to speed). Part of the solution is shown in Fig. 5.

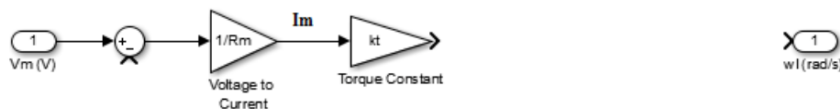


Figure 5: Incomplete Qube-Servo Model subsystem.

- Build and run the QUARC[®] controller with your Qube-Servo 3 model in the edited file `q_qube3_model_student.slx`. Make sure to run the MATLAB[®] file `qube3_param.m` to load all the necessary variables in the workspace before running the experiment. The scope response should be similar to Fig. 6. Does your model represent the Qube-Servo 3 reasonably well? Include plots in your report showing: 1- Input voltage; 2- Simulated and actual velocity output; 3- Simulated and actual position output. The SIMULINK[®] file `q_qube3_model_student.slx` should also be modified so that SIMULINK[®] saves the signals in either the workspace or in “mat” files. Use the provided file `MEM05_Qube_DCMotor_OLResponse.slx` as an example. This will enable you to produce the necessary plots by a modified version of the provided file `MEM05_Qube_DCMotor_Plots.m`.

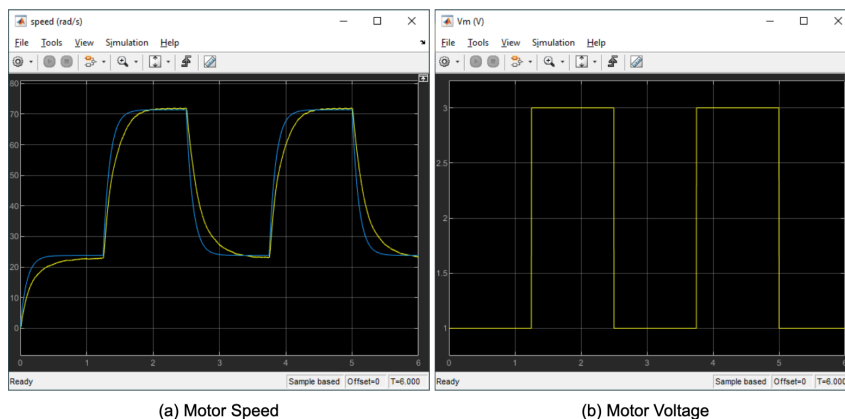


Figure 6: Example response when running block diagram model and with Qube-Servo 3 hardware in parallel.

- You may notice that the the model does not match the measured system exactly. What could cause this difference?
- Stop the QUARC[®] controller.
- Power off the Qube-Servo 3.

3 PID Control

The proportional, integral, and derivative control can be expressed mathematically as follows

$$u(t) = k_p e(t) + k_i \int_0^t e(\tau) d\tau + k_d \frac{de(t)}{dt}. \quad (9)$$

The control action is a sum of three terms referred to as proportional (P), integral (I) and derivative (D) control gain. The input of the controller is the error $e(t) = r(t) - y(t)$, which is the difference between the reference $r(t)$ and the output of the plant $y(t)$. The output of the controller is the input of the plant $u(t)$. By applying the Laplace transform to the two sides of the equation and assuming zero initial conditions, the controller (9) can also be described by the transfer function

$$C(s) = \frac{U(s)}{E(s)} = k_p + \frac{k_i}{s} + k_d s, \quad (10)$$

where $U(s) \triangleq \mathcal{L}\{u(t)\}$, $E(s) \triangleq \mathcal{L}\{e(t)\}$, $R(s) \triangleq \mathcal{L}\{r(t)\}$, and $Y(s) \triangleq \mathcal{L}\{y(t)\}$. The corresponding block diagram is given in Fig. 7. The functionality of the PID controller can be summarized as follows. The proportional term is based on the present error, the integral term depends on past errors, and the derivative term is a prediction of future errors. The design of the PID controller reduces to the selection of the proportional gain k_p , integral gain k_i , and derivative gain k_d so that the output $y(t)$ tracks the reference $r(t)$.

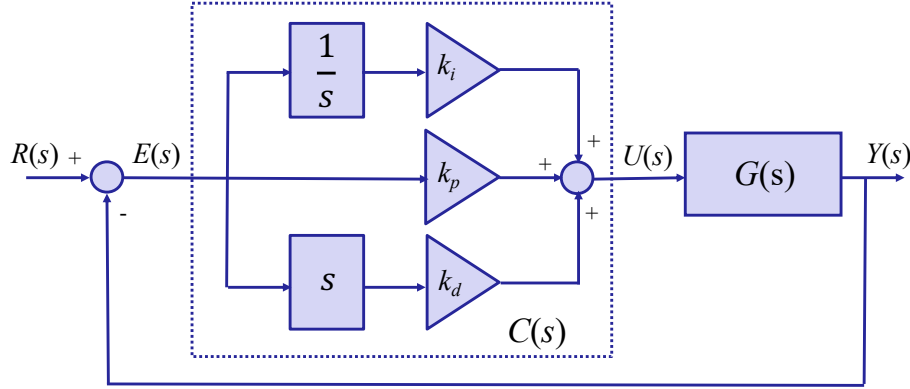


Figure 7: Block diagram of PID control.

The Final Value theorem can be used to determine the steady-state or final value of the tracking error $e(t)$ given its Laplace transform $E(s)$. For a stable system response (i.e. all poles of the system are strictly in the left half of the s -plane or, in other words, the real parts of all the poles are negative), the following holds

$$e_{ss} = r_{ss} - y_{ss} = \lim_{t \rightarrow \infty} e(t) = \lim_{s \rightarrow 0} sE(s), \quad (11)$$

where the subscript (ss) denotes “steady-state.” This is of high interest because e_{ss} is a measure of the tracking capabilities of the controller ($e_{ss} = 0$ implies perfect tracking ($y_{ss} = r_{ss}$)).

Lab Work 3:

1. Show that the transfer function $Y(s)/R(s)$ in terms of the general controller $C(s)$ and the plant $G(s)$ for a closed-loop system as the one shown in Fig. 7 can be written as

$$\frac{Y(s)}{R(s)} = \frac{C(s)G(s)}{1 + C(s)G(s)}. \quad (12)$$

2. Show that the transfer function $E(s)/R(s)$ in terms of the general controller $C(s)$ and the plant $G(s)$ for a closed-loop system as the one shown in Fig. 7 can be written as

$$\frac{E(s)}{R(s)} = \frac{1}{1 + C(s)G(s)}. \quad (13)$$

4 Velocity Control

In this lab, you will examine how a proportional controller can be used to control the QUBE-Servo-3 velocity as shown in Fig. 8.

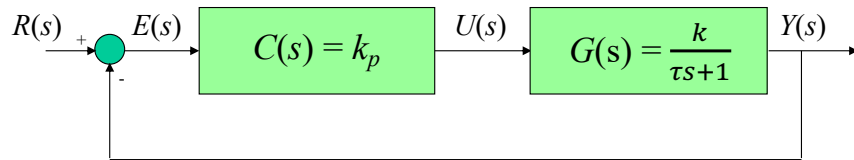


Figure 8: Proportional control of servo velocity.

The proportional (P) control has the following structure

$$u(t) = k_p e(t) = k_p (r(t) - y(t)) \quad (14)$$

where k_p is the proportional gain, $r(t) = \omega_d(t)$ is the reference angular velocity, $y(t) = \omega_m(t)$ is the measured angular velocity, and $u(t) = v_m(t)$ is the control input (applied motor voltage).

Lab Work 4:

1. Show that for our plant $G(s) = k/(\tau s + 1)$ ($k = 23.8$ and $\tau = 0.1$) with a proportional (P) controller the closed-loop transfer functions (12) and (13) take the forms:

$$\frac{Y(s)}{R(s)} = \frac{Kk_p}{\tau s + 1 + Kk_p}, \quad (15)$$

$$\frac{E(s)}{R(s)} = \frac{\tau s + 1}{\tau s + 1 + Kk_p}. \quad (16)$$

What is the order of the closed-loop system? Provide a formula for the closed-loop pole. Is the closed-loop system stable when $k_p > 0$? How the location of the closed-loop pole changes as you increase the proportional gain? What will be the effect of increasing the proportional gain on the time response of the closed-loop system?

2. By using the Final Value theorem show that the steady-state error when tracking a step reference $R(s) = R_0/s$, where R_0 is the desired angular rate step amplitude, is given by

$$e_{ss} = \lim_{t \rightarrow \infty} e(t) = \frac{R_0}{1 + Kk_p}. \quad (17)$$

Is it possible to achieve perfect tracking with a proportional (P) controller? How could you reduce the steady-state tracking error?

3. The voltage that can be applied to the QUBE-Servo 3 is limited to ± 10 V. Determine the maximum proportional gain for a square wave reference signal of ± 0.5 rad/s that does not saturate the QUBE-Servo 3. What will be the minimum possible steady-state error with the maximum proportional gain? Use the SIMULINK[®] file `MEM05_Qube_DCMotor_PControl_Velocity.slx` provided in the “Design” directory to corroborate these calculations. Provide a plot comparing reference and actual angular velocity and confirming the value of the calculated steady-state error associated with the computed maximum proportional gain. Also provide a plot showing the motor voltage. The provided SIMULINK[®] file will need to be modified to allow plotting from MATLAB[®]. Use a ± 0.5 rad/s square wave at a frequency of 0.4 Hz.
4. Open the SIMULINK[®] file `q_qube3_p_speed_cntrl.slx` provided in the “Test” directory. To generate a reference speed command with an amplitude of 15rad/s, set the Amplitude and Offset gain blocks both to 7.5 rad/s and ensure the Smooth Signal Generator block is configured to output a square wave at a frequency of 0.4 Hz.
5. Build and run the QUARC[®] controller.
6. Calculate the steady-state error if the proportional gain is 0.5Vs/rad and the step amplitude is 15rad/s using (17). Measure the actual steady-state error. How does it compare with your value computed above? If they do not match, give one reason why there would be a difference.

Hint: Use the Cursor Measurement tool in the Simulink Scope to take your measurements or modify the SIMULINK[®] file `q_qube3_p_speed_cntrl.slx` as it has been done in the sample file `MEM05_Qube_DCMotor_OLResponse.slx`.

7. Show how the error can be decreased by half its current magnitude. Validate your results with the QUBE-Servo 3 and show your response.
8. Stop the QUARC[®] controller.
9. Power off the QUBE-Servo 3 if no more experiments will be conducted in this session.

5 Position Control

In this lab, you will examine how both a proportional controller and a proportional-derivative controller can be used to control the QUBE-Servo-3 position as shown in Fig. 9.

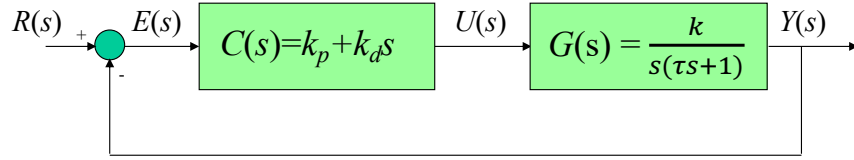


Figure 9: Proportional-derivative control of servo position.

The proportional (P) control has the following structure

$$u(t) = k_p e(t) = k_p (r(t) - y(t)) \quad (18)$$

while the proportional-derivative (PD) control has the following structure

$$u(t) = k_p e(t) + k_d \frac{de(t)}{dt} = k_p (r(t) - y(t)) + k_d \frac{(r(t) - y(t))}{dt} \quad (19)$$

where k_p is the proportional gain, k_d is the derivative gain, $r(t) = \theta_d(t)$ is the setpoint or reference motor/load angle for the position control, $y(t) = \theta_m(t)$ is the measured load shaft angle, and $u(t) = v_m(t)$ is the control input (applied motor voltage).

Lab Work 5:

1. Show that for our plant $G(s) = k/(s(\tau s + 1))$ ($k = 23.8$ and $\tau = 0.1$) with a proportional (P) controller ($k_p \neq 0$ and $k_d = 0$) the closed-loop transfer functions (12) and (13) take the forms:

$$\frac{Y(s)}{R(s)} = \frac{\frac{Kk_p}{\tau}}{s^2 + \frac{1}{\tau}s + \frac{Kk_p}{\tau}}, \quad (20)$$

$$\frac{E(s)}{R(s)} = \frac{s(s + \frac{1}{\tau})}{s^2 + \frac{1}{\tau}s + \frac{Kk_p}{\tau}}. \quad (21)$$

What is the order of the closed-loop system? Provide a formula for the closed-loop poles. Is the closed-loop system stable when $k_p > 0$? Can the locations of the poles be arbitrarily determined by selecting the appropriate k_p ? How the locations of the closed-loop poles change as you increase the proportional gain? What will be the effect of increasing the proportional gain on the time response of the closed-loop system? Remember that if the poles of a second-order system are located at $s = -\sigma \pm j\omega_d$, the damping ratio is given by $\zeta = \sigma/\sqrt{\sigma^2 + \omega_d^2}$ and the natural frequency is given by $\omega_n = \sqrt{\sigma^2 + \omega_d^2}$. Use the SIMULINK[®] file `MEM05_Qube_DCMotor_PControl_Position.slx` provided in the “Design” directory to corroborate your answers.

2. What is the steady-state error when tracking a step reference $R(s) = R_0/s$, where R_0 is the desired angular step amplitude? Why is it possible to achieve perfect tracking with a proportional (P) controller when regulating the position while it is not possible when regulating the velocity (see Lab Work 4)? Provide a plot comparing reference and actual angular position and confirming the value of the calculated steady-state

error when $k_p = 1.5$. Also provide a plot showing the motor voltage. The provided SIMULINK[®] file will need to be modified to allow plotting from MATLAB[®]. Use a ± 0.5 rad square wave at a frequency of 0.4 Hz.

3. Open the SIMULINK[®] file `q_qube3_p_pos_cntrl1.slx` provided in the “Test” directory. Set the Signal Generator block such that the servo command (i.e. reference angle) is a square wave with an amplitude of 0.5 rad and at a frequency of 0.4 Hz.
4. Build and run the QUARC[®] controller.
5. Run the controller with a proportional gain of $k_p = 1.5$. Provide one plot showing the position response in comparison with the reference and another plot showing the motor voltage responses. Measure the percent overshoot M_p and peak time t_p of the response when $k_p = 1.5$. Compare these values with the theoretical values based on the model:

$$t_p = \frac{\pi}{\omega_n \sqrt{1 - \zeta^2}}, M_p[\%] = e^{-\pi \frac{\zeta}{\sqrt{1 - \zeta^2}}} \times 100. \quad (22)$$

6. Is there a steady-state error? If so, evaluate it.
7. Vary k_p between 1 and 5. How does the proportional gain affect the servo position control response?
8. Vary k_p between 0.1 and 1. What happens when k_p is decreased?
9. Stop the QUARC[®] controller.
10. Show now that for our plant $G(s) = k/(s(\tau s + 1))$ ($k = 23.8$ and $\tau = 0.1$) with a proportional-derivative (PD) controller ($k_p \neq 0$ and $k_d \neq 0$) the closed-loop transfer functions (12) and (13) take the forms:

$$\frac{Y(s)}{R(s)} = \frac{\frac{K(k_p + k_d s)}{\tau}}{s^2 + \frac{1 + K k_d}{\tau} s + \frac{K k_p}{\tau}}, \quad (23)$$

$$\frac{E(s)}{R(s)} = \frac{s(s + \frac{1}{\tau})}{s^2 + \frac{1 + K k_d}{\tau} s + \frac{K k_p}{\tau}}. \quad (24)$$

What is the order of the closed-loop system? Provide a formula for the closed-loop poles. Is the closed-loop system stable when $k_p > 0$ and $k_d > 0$? Can the locations of the poles be arbitrarily determined by selecting the appropriate k_p ?

11. By comparing the closed transfer function (23) with the standard second-order transfer function,

$$\frac{Y(s)}{R(s)} = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}, \quad (25)$$

give formulas for both k_p and k_d as functions of ζ and ω_n .

12. Calculate k_p and k_d to make the response of the closed loop system have a peak time of 0.15 s and a percentage overshoot of 2.5%. Use the formulas in (22) to obtain the needed ζ and ω_n . Use the SIMULINK[®] file MEM05_Qube_DCMotor_PDControl_Position.slx provided in the “Design” directory to corroborate your answers.
13. Open the SIMULINK[®] file q_qube3_pd.slx provided in the “Test” directory. Set the Signal Generator block such that the servo command (i.e. reference angle) is a square wave with an amplitude of 0.5 rad and at a frequency of 0.4 Hz.
14. Build and run the QUARC[®] controller.
15. Set $k_p = 2.5$ V/rad and $k_d = 0$ V/(rad/s). Keep the derivative gain at 0 and vary k_p between 1 and 4. How does the proportional gain affect the servo position control response?
16. Set $k_p = 2.5$ V/rad and vary the derivative gain k_d between 0 and 0.15 V/(rad/s). How does the derivative gain affect the servo position control response?
17. Run the controller with the gains you calculated to satisfy the time constraints. Provide a plot comparing reference and actual angular positions and a plot showing the motor voltage.
18. Measure the percent overshoot and peak time of the Qube-Servo-3 response. Do they match the desired percent overshoot and peak time specifications without saturating the motor (going beyond ± 10 V)?
Hint: Use the Cursor Measurements tool in the SIMULINK[®] Scope to take measurements of the response.
19. If your response did not match the above overshoot and peak time specification, try tuning your control gains until your response does satisfy them. Attach the MATLAB[®] figures showing your results and comment on how you modified your controller to arrive at those results.
20. Stop the QUARC[®] controller.
21. Power off the QUBE-Servo 3 if no more experiments will be conducted in this session.