



STUDENT WORKBOOK

1 DOF Torsion Experiment for MATLAB®/Simulink® Users

Standardized for ABET* Evaluation Criteria

Developed by:
Jacob Apkarian, Ph.D., Quanser
Hervé Lacheray, M.A.Sc., Quanser
Michel Lévis, M.A.Sc., Quanser

Quanser educational solutions
are powered by:



Course material
complies with:



CAPTIVATE. MOTIVATE. GRADUATE.

*ABET Inc., is the recognized accreditor for college and university programs in applied science, computing, engineering, and technology; and has provided leadership and quality assurance in higher education for over 75 years.

© 2012 Quanser Inc., All rights reserved.

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>

This document and the software described in it are provided subject to a license agreement. Neither the software nor this document may be used or copied except as specified under the terms of that license agreement. All rights are reserved and no part may be reproduced, stored in a retrieval system or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of Quanser Inc.

ACKNOWLEDGEMENTS

Quanser, Inc. would like to thank Dr. Hakan Gurocak, Washington State University Vancouver, USA, for his help to include embedded outcomes assessment.

CONTENTS

1	Introduction	4
2	Modeling	5
2.1	Background	5
2.2	Pre-Lab Exercises	9
2.3	Lab Experiments	10
2.4	Results	14
3	Control Design	15
3.1	Specifications	15
3.2	Background	15
3.3	Pre-Lab Questions	17
3.4	Lab Experiments	18
3.5	Results	22
4	System Requirements	23
4.1	Overview of Files	24
4.2	Setup for Finding Stiffness	24
4.3	Setup for Model Validation	25
4.4	Setup for 1 DOF Torsion Control Simulation	25
4.5	Setup for Rotary Torsion Control Implementation	26
5	Lab Report	27
5.1	Template for Content (Modeling)	27
5.2	Template for Content (Control)	28
5.3	Tips for Report Format	29

1 INTRODUCTION

The challenge of this experiment is to design a feedback controller to position the output of a rotational system with one torsional flexible coupling as fast as possible with minimum vibration. Such a system emulates torsional compliance and joint flexibility, which are common characteristics in mechanical systems such as high-gear-ratio harmonic drives or lightweight transmission shafts.

The laboratory objective is to design a state-feedback controller for the rotary torsion module which allows you to command a desired load angle position via a flexible coupling. The controller should eliminate the load's vibrations while maintaining a fast response. Full-state and partial-state feedback control strategies are compared. The Linear Quadratic Regulator (LQR) tuning algorithm is used. Frequency tests are carried out to examine the system's resonance.

Topics Covered

- Model the rotary 1 DOF torsional system from first-principles.
- Identify the stiffness and the damping of the system.
- Using a nominal model parameters, design a state-feedback control using Linear Quadratic Regulator (LQR).
- Tune the LQR control in simulation.
- Implement the state-feedback control on the actual 1 DOF Torsion device using **QUARC®**.
- Evaluate the different between partial-state feedback and full-state feedback.

Prerequisites

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

- Transfer function fundamentals, e.g. obtaining a transfer function from a differential equation.
- Basics of **Simulink®**.
- QUARC Integration Laboratory in [6] in order to be familiar using **QUARC®** with the SRV02.

2 MODELING

2.1 Background

2.1.1 Equations of Motion

The Rotary 1 DOF Torsion schematic is depicted in Figure 2.1. Notice that the positive rotation is in counter-clockwise direction. Thus when a positive voltage is applied to the motor, a positive torque is generated and both the servo and the torsion load move counter-clockwise.

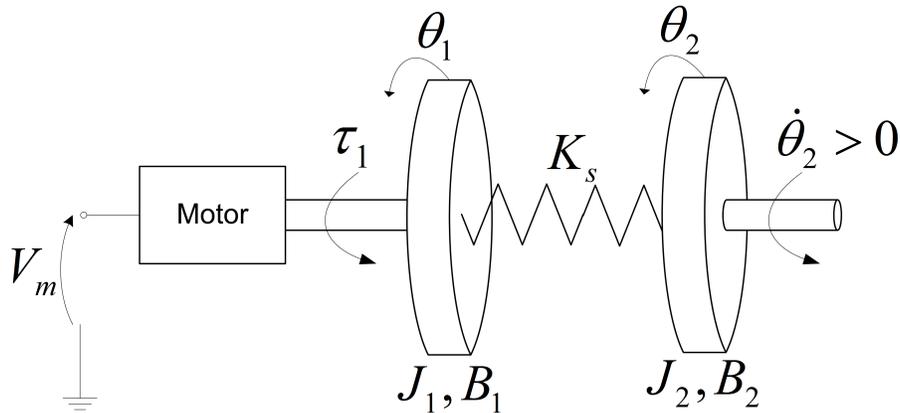


Figure 2.1: Rotary 1 DOF Torsion System

The symbols used for the schematic and the modeling are listed in Table 2.1.

Symbol	Description
V_m	Servo dc motor input voltage.
τ_1	Torque generated by servo.
θ_1	Angular position of servo load shaft.
J_1	Equivalent moment of inertia of servo.
B_1	Equivalent viscous damping acting at servo load shaft.
J_2	Equivalent moment of inertia of torsion load shaft.
B_2	Equivalent viscous damping acting at torsion load shaft.
θ_2	Angular position of torsion load shaft.
K_s	Stiffness of flexible coupling.

Table 2.1: 1 DOF Torsion Model Parameters

The equations of motion (EOMs) can be represented by a set of linear equations of the form

$$J\ddot{q}(t) + B\dot{q}(t) + Kq(t) = \tau \quad (2.1)$$

where J is the system inertia, B is the damping, K is the stiffness, τ is the applied torque, and q are generalized coordinates. For multi-axis systems, the terms J , B , and K would be matrices and q a vector. For the 1 DOF Torsion we have two equations in the form:

$$\ddot{\theta}_1 = g_1(\theta_1, \theta_2, \dot{\theta}_1, \tau_1) \quad (2.2)$$

and

$$\ddot{\theta}_2 = g_2(\theta_1, \theta_2, \dot{\theta}_2) \quad (2.3)$$

where g_1 and g_2 are functions that represent the angular acceleration of the SRV02 load shaft and the 1 DOF Torsion load shaft, respectively. The acceleration of the SRV02, Equation 2.2, depends on the torque being applied, τ_1 , the

angle of the servo, θ_1 , the angle of the torsion module, θ_2 , and the velocity of the SRV02 gear (due to damping). The torsion acceleration depends on the angle of the servo, θ_1 , as well as the angle of the torsion module, θ_2 , and its velocity, $\dot{\theta}_2$.

The torque applied at the base of the rotary arm (i.e., at the load gear) is generated by the servo motor as described by the equation

$$\tau = \frac{\eta_g K_g \eta_m k_i (V_m - K_g k_m \dot{\theta}_1)}{R_m}. \quad (2.4)$$

See the SRV02 User Manual [4] for a description of the corresponding SRV02 parameters (e.g., such as the back-emf constant, k_m).

2.1.2 Linear State-Space Model

The linear state-space equations are

$$\dot{x} = Ax + Bu \quad (2.5)$$

and

$$y = Cx + Du \quad (2.6)$$

where x is the state, u is the control input, A , B , C , and D are state-space matrices. For the 1 DOF Torsion system, the state and output are defined

$$x^T = [\theta_1 \quad \theta_2 \quad \dot{\theta}_1 \quad \dot{\theta}_2] \quad (2.7)$$

and

$$y^T = [x_1 \quad x_2]. \quad (2.8)$$

In the output equation, only the position of the servo and torsion angles are being measured. Based on this, the C and D matrices in the output equation are

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \quad (2.9)$$

and

$$D = \begin{bmatrix} 0 \\ 0 \end{bmatrix}. \quad (2.10)$$

The velocities of the servo and torsion angles can be computed in the digital controller, e.g., by taking the derivative and filtering the result through a high-pass filter.

2.1.3 Free-Oscillation of a Second Order System

The free-oscillatory equation of motion of a second-order system described by

$$J\ddot{x} + B\dot{x} + Kx = 0 \quad (2.11)$$

is shown in Figure 2.2. Assuming the initial conditions $x(0^-) = x_0$ and $\dot{x}(0^-) = 0$, the Laplace transform of Equation 2.11 is

$$X(s) = \frac{\frac{x_0}{J}}{s^2 + \frac{B}{J}s + \frac{K}{J}} \quad (2.12)$$

The prototype second-order equation is defined

$$s^2 + 2\zeta\omega_n s + \omega_n^2,$$

where ζ is the damping ratio and ω_n is the natural frequency. Equating the characteristic equation in 2.12 to this gives

$$\omega_n^2 = \frac{K}{J}$$

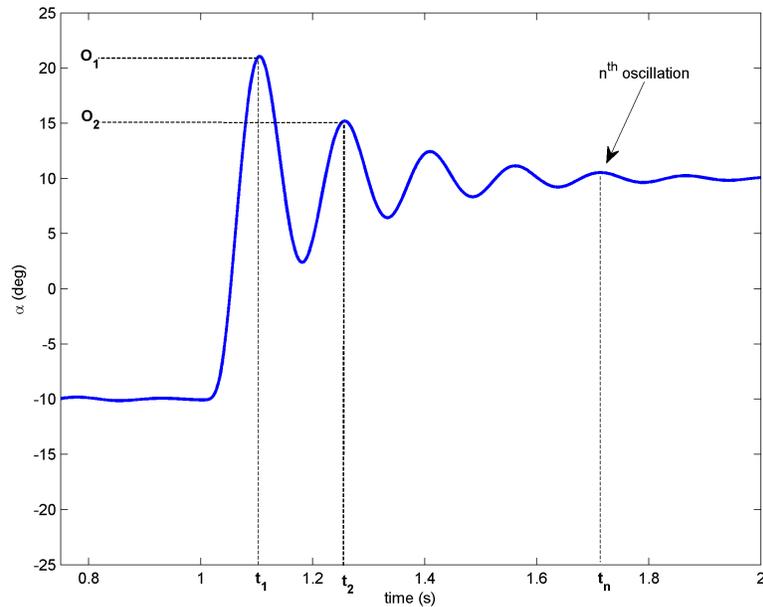


Figure 2.2: Free Oscillation Response

and

$$2\zeta\omega_n = \frac{B}{J}$$

Based on the measured damping ratio and natural frequency, the friction (or stiffness) of the system is

$$K = J\omega_n^2 \quad (2.13)$$

and the viscous damping is

$$B = 2\zeta\omega_n J. \quad (2.14)$$

Finding the Natural Frequency

The period of the oscillations in a system response can be found using the equation

$$T_{osc} = \frac{t_n - t_1}{n - 1} \quad (2.15)$$

where t_n is the time of the n^{th} oscillation, t_1 is the time of the first peak, and n is the number of oscillations considered. From this, the damped natural frequency (in radians per second) is

$$\omega_d = \frac{2\pi}{T_{osc}} \quad (2.16)$$

and the undamped natural frequency is

$$\omega_n = \frac{\omega_d}{\sqrt{1 - \zeta^2}}. \quad (2.17)$$

Finding the Damping Ratio

The damping ratio of a second-order system can be found from its response. For a typical second-order underdamped system, the subsidence ratio (i.e., decrement ratio) is defined as

$$\delta = \frac{1}{n} \ln \frac{O_1}{O_n} \quad (2.18)$$

where O_1 is the peak of the first oscillation and O_n is the peak of the n^{th} oscillation. Note that $O_1 > O_n$, as this is a decaying response.

The damping ratio is defined

$$\zeta = \frac{1}{\sqrt{1 + \frac{2\pi^2}{\delta}}} \quad (2.19)$$

2.2 Pre-Lab Exercises

1. Find the equation of motion of the servo by completing Equation 2.2 and Equation 2.3.
2. Given state x defined in Equation 2.7, find the linear state-space matrices A and B .
3. Find the the natural frequency of the response shown in Figure 2.2 if the peak values for the first and fifth oscillation are: $t_1 = 1.12$ and $t_5 = 1.71$ seconds. Because the damping is low, assume the damped and undamped natural frequency are equivalent.

2.3 Lab Experiments

In the first part of this laboratory, the stiffness of the torsion is determined by measuring its natural frequency. In the second part, the state-space model is finalized and validated against actual measurements.

2.3.1 Finding Stiffness

In Section 2.1.3 we found an equation describing the free-oscillation response of a second-order system. This can also be used to describe the response of the torsional unit when initially perturbed and left to decay.

Physical Parameters for the Lab

In order to do some of the laboratory exercises, you will need these values:

$$\begin{aligned} B_1 &= 0.015 \text{ N-m/(rad/s)} \\ J_1 &= 2.18 \times 10^{-3} \text{ kg-m}^2 \\ J_2 &= 5.45 \times 10^{-4} \text{ kg-m}^2 \end{aligned}$$

Note: The equivalent viscous damping, B_1 , and moment of inertia, J_1 , parameters are for the SRV02 when there is no load (i.e., the parameter found in the SRV02 Modeling Laboratory was for servo with the disc load).

Experimental Setup

The `q_torsion_id` Simulink diagram shown in Figure 2.3 is used to find the natural frequency of the torsion unit. For more information about QUARC, see Reference [2]. This model sends a short step voltage to the Rotary Servo and outputs the measured torsion module angle.

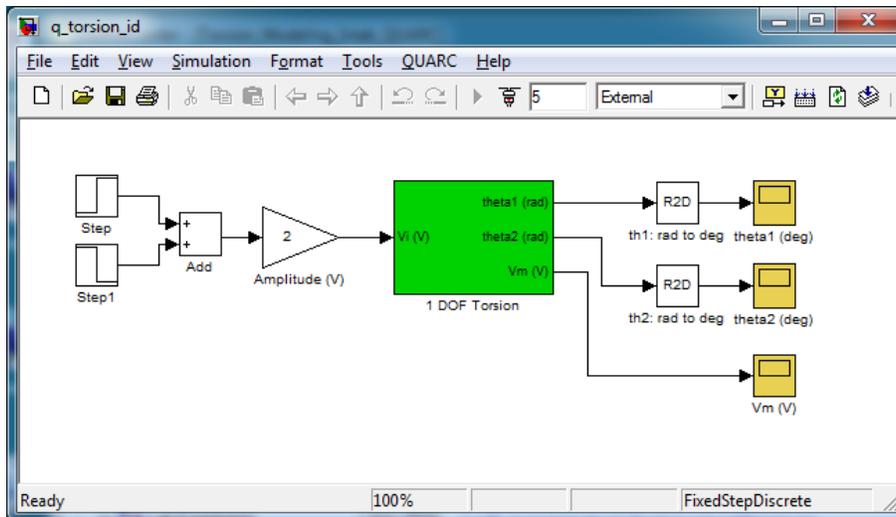


Figure 2.3: Simulink diagram used to find stiffness

IMPORTANT: Before you can conduct this experiment, you need to make sure that the lab files are configured according to your system setup. If they have not been configured already, then you need to go to Section 4.2 to configure the lab files first.

1. In the `q_torsion_id` Simulink diagram, go to QUARC | Build to build the QUARC controller.
2. Go to QUARC | Start to run the controller. The servo should begin tracking the reference angle. The θ_1 (deg) and θ_2 (deg) scopes should be reading a response similarly as shown in Figure 2.4. Note that the controller is set to run for 5 seconds.

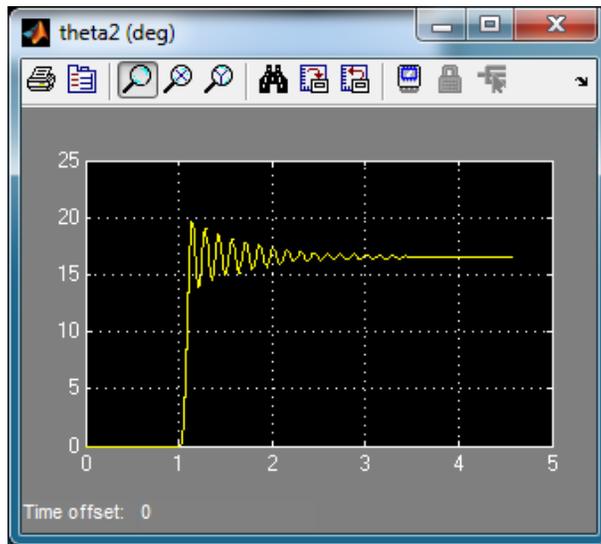


Figure 2.4: Typical Torsion Free Oscillation Response

3. After the controller stops (i.e., after 5 sec), the data is automatically saved in the Matlab workspace to the variable `data_theta2`. The time is stored in `data_theta2(:,1)` vector and the torsion angle is stored the `data_theta(:,2)` vector. Plot the response in a Matlab figure.
4. Find the natural frequency of the flexible coupling. As in Pre-Lab Question 3, the damping is low. Therefore assume the damped natural frequency (which is being measured) is equivalent to the undamped natural frequency.
5. Calculate the stiffness of the flexible coupling, K_s .

2.3.2 State-Space Model

In this section, the the state-space model representing the Quanser 1 DOF Torsion system is setup.

1. Run the `setup_torsion.m`.
2. Enter the stiffness you found in Section 2.3.1 as the Matlab variable `Ks`. Otherwise, it will use a default value of the stiffness.
3. Depending on your stiffness, the Matlab prompt should generate a gain similarly as shown below (this gain is generated for a K_s of 0.5):

```
K =
    39.7946   -38.3804    1.4705   -0.2055
```

This means the script ran correctly.

4. In Matlab, open the M-File called `Torsion_ABCD_eqns_student.m`. The script has the following state-space matrices entered:

```
A = [0 0 1 0;
      0 0 0 1;
      -400 400 0 0;
      2250 -2250 0 0];
```

```
B = [0; 0; 3500; 0];
```

```
C = zeros(2,4);
```

```
D = zeros(2,1);
```

5. Enter the state-space matrices you found in Section 2.2 for A , B , C , and D . In Matlab, the SRV02 moment of inertia and damping are denoted $J1$ and $B1$, the TORSION moment of inertia and damping are defined as $J2$ and $B2$, and the stiffness is Ks .
6. Run the `Torsion_ABCD_eqns_student.m` script to load the state-space matrices in the Matlab workspace. Show the numerical matrices that are displayed in the Matlab prompt.
7. The input of the state-space model you found in Section 2.2 is the torque acting at the servo load gear. However, *we do not control torque directly - we control the servo input voltage*. Recall the voltage-torque relationship given in Equation 2.4 in Section 2.1.1. In the `System Model` section of the `setup_torsion.m` script, the actuator dynamics are added to your state-space matrices with the code:

```
% Add actuator dynamics  
B = Kg * eta_g * kt * eta_m * B / Rm;  
A(3,3) = A(3,3) - Kg*km*B(3);  
A(4,3) = A(4,3) - Kg*km*B(4);
```

8. Run the `setup_torsion.m` script again so the Torsion state-space model is based on dc motor voltage.
9. In Matlab, find the open-loop poles (i.e., eigenvalues) of the system using the state-space matrix A that is loaded.
Note: These will be required for a pre-lab question in Section 3.3.

2.3.3 Model Validation

By running a simulation and the actual device in parallel, we can verify whether the dynamic model (which drives the simulation) accurately represents our system.

Experimental Setup

The `q_torsion_val` Simulink diagram shown in Figure 2.5 applies a step input to both the Quanser Rotary 1 DOF Torsion hardware and its dynamic model and reads the measured and simulated servo and torsion angles, θ_1 and θ_2 . The *1 DOF Torsion* subsystem contains the QUARC blocks that interface to the actual hardware. The Simulink *State-Space* block reads the A , B , C , and D state-space matrices that are loaded in the Matlab workspace. This model outputs the servo and torsion angles.

IMPORTANT: Before you can conduct this experiment, you need to make sure that the lab files are configured according to your system setup. If they have not been configured already, then go to Section 4.3 to configure the lab files first.

1. Make sure the state-space model has been loaded, as dictated in Section 2.3.2.
2. Open the `q_torsion_val` Simulink diagram.
3. In the Simulink diagram, go to QUARC | Build to build the QUARC controller.
4. Go to QUARC | Start to run the `q_torsion_val` controller. Typical scope responses are shown in Figure 2.6. The *theta1 (deg)* scope displays the simulated servo angle in yellow and the measured angle in purple. Similarly, the *theta2 (deg)* scope shows the simulated torsion angle in yellow and the measured angle in purple.
5. If your simulation and measured response match, go to the next step. If they do not, then there is an issue with your model. Here are some issues to investigate:

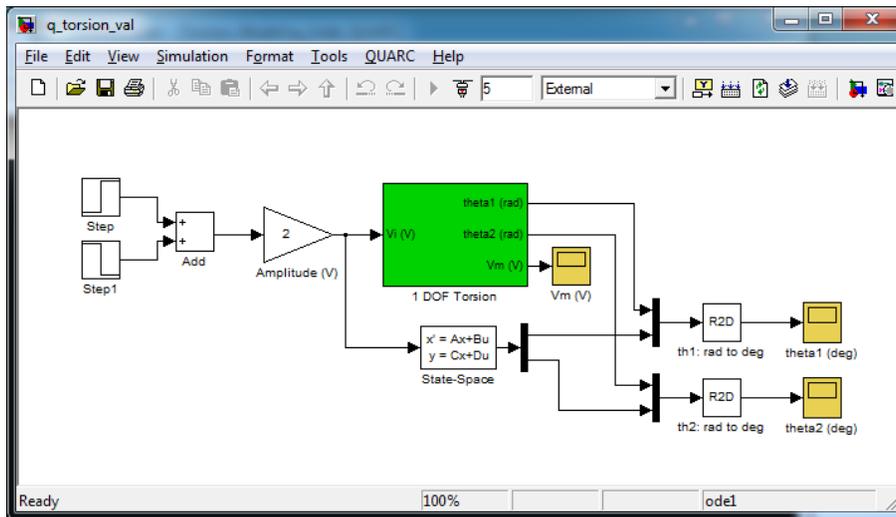


Figure 2.5: q_torsion_val Simulink diagram used validate the model

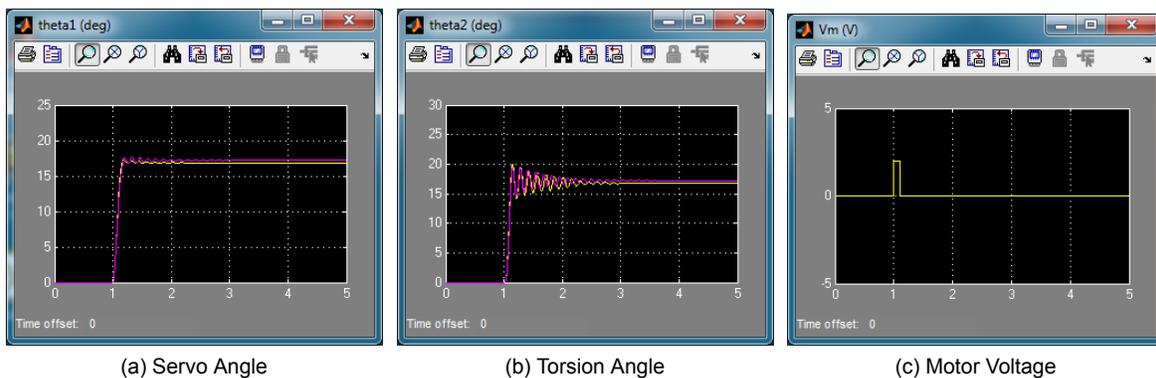


Figure 2.6: Typical Model Validation Response

- Ensure the state-space model was entered properly in the script.
 - The stiffness, K_s , found in Section 2.3.1 is not correct. Review your calculations or redo the experiment.
 - Review your model derivation in Section 2.2, e.g., might be a mistake in solving the EOMs.
6. The $theta1(deg)$ and $theta2(deg)$ scopes save their data to the variables $data_theta1$ and $data_theta2$. For $data_theta1$, the time is in $data_theta1(:,1)$, the simulated servo angle is in $data_theta1(:,2)$, and the measured SRV02 angle is in $data_theta1(:,3)$. The same structure applies to the torsion response in $data_theta2$. Plot the response in a Matlab figure.
 7. How well does your model represent the actual system? We want a model that is fairly representative but, having said that, keep in mind that no model is perfect. This is just a quick test to see how well your model represents the actual device. As shown in 2.6, the simulation does not match the measured response perfectly.
 8. Give at least one reason why the model does not represent the system accurately.

2.4 Results

Fill out Table 2.2 with your answers from your modeling lab results - both simulation and implementation.

Description	Symbol	Value	Unit
Finding Stiffness			
Natural frequency	ω_n		rad/s
Stiffness	K_s		N m/rad
Model Validation			
State-Space Matrix	A		
State-Space Matrix	B		
State-Space Matrix	C		
State-Space Matrix	D		
Open-loop poles	OL		

Table 2.2: Results

3 CONTROL DESIGN

3.1 Specifications

The time-domain requirements are:

Specification 1: Torsion angle 2% settling time: $t_s \leq 0.4$ s.

Specification 2: Torsion angle percentage overshoot: $PO \leq 5$ %.

Specification 3: Torsion angle steady-state error: $e_{ss} \leq 0.5$ deg.

Specification 4: Maximum control effort / voltage: $|V_m| \leq 10$ V.

These specifications are to be satisfied when the servo is tracking a ± 10 degree angle square wave.

3.2 Background

In Section 2.2, we found a linear state-state space model that represents the 1 DOF Torsion system. This model is used to investigate the stability properties of the 1 DOF Torsion system in Section 3.2.1. In Section 3.2.2, the notion of controllability is introduced. Using the Linear Quadratic Regular algorithm, or LQR, is a common way to find the control gain and is discussed in Section 3.2.3. Lastly, Section 3.2.4 describes the state-feedback control used to control the torsion load position while minimizing the effect from the flexible coupling.

3.2.1 Stability

The stability of a system can be determined from its poles ([8]):

- Stable systems have poles only in the left-hand plane.
- Unstable systems have at least one pole in the right-hand plane and/or poles of multiplicity greater than 1 on the imaginary axis.
- Marginally stable systems have one pole on the imaginary axis and the other poles in the left-hand plane.

The poles are the roots of the system's characteristic equation. From the state-space, the characteristic equation of the system can be found using

$$\det(sI - A) = 0 \quad (3.1)$$

where $\det()$ is the determinant function, s is the Laplace operator, and I the identity matrix. These are the *eigenvalues* of the state-space matrix A .

3.2.2 Controllability

If the control input, u , of a system can take each state variable, x_i where $i = 1 \dots n$, from an initial state to a final state then the system is controllable, otherwise it is uncontrollable ([8]).

Rank Test The system is controllable if the rank of its controllability matrix

$$T = [B \ AB \ A^2B \ \dots \ A^nB] \quad (3.2)$$

equals the number of states in the system,

$$\text{rank}(T) = n. \quad (3.3)$$

3.2.3 Linear Quadratic Regular (LQR)

If (A,B) are controllable, then the Linear Quadratic Regular optimization method can be used to find a feedback control gain. Given the plant model in Equation 2.5, find a control input u that minimizes the cost function

$$J = \int_0^{\infty} x(t)'Qx(t) + u(t)'Ru(t) dt, \quad (3.4)$$

where Q and R are the weighting matrices. The weighting matrices affect how LQR minimizes the function and are, essentially, tuning variables.

Given the control law $u = -Kx$, the state-space in Equation 2.5 becomes

$$\begin{aligned} \dot{x} &= Ax + B(-Kx) \\ &= (A - BK)x \end{aligned}$$

3.2.4 Feedback Control

The feedback control loop that in Figure 3.1 is designed to stabilize the servo and torsion to a desired angular position, θ_d , while minimizing the vibration introduced by the flexible coupling.

The reference state is defined

$$x_d = [\theta_d \quad \dot{\theta}_d \quad 0 \quad 0]$$

and the controller is

$$u = K(x_d - x).$$

Note that if $x_d = 0$ then $u = -Kx$, which is the control used in the LQR algorithm.

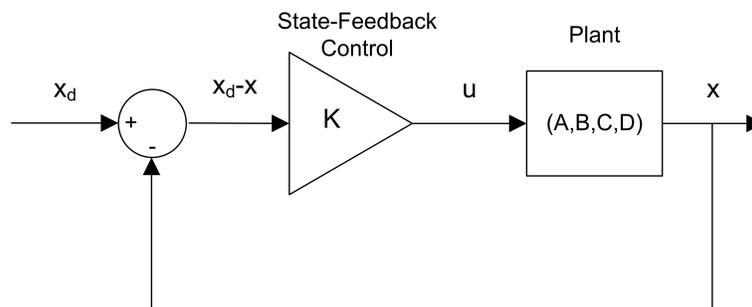


Figure 3.1: State-feedback control loop

3.3 Pre-Lab Questions

1. Based on your analysis of the system in the Modeling Laboratory (Section 2.3), is the system stable, marginally stable, or unstable? From your experience in Section 2.3, does the stability you determined analytically match how the actual system behaves?
2. Designing a controller with the Linear Quadratic Regular (LQR) technique is an iterative process. In software, you have to tune the Q and R matrices, generate the gain K using LQR, and either simulate the system or implement the control to see if you have the desired response. The relationship between changing Q and R and the closed-loop response is not evident. However, we can have a better idea on how changing the different elements in Q and R will effect the response. We will only be changing the diagonal elements in Q , thus let

$$Q = \begin{bmatrix} q_1 & 0 & 0 & 0 \\ 0 & q_2 & 0 & 0 \\ 0 & 0 & q_3 & 0 \\ 0 & 0 & 0 & q_4 \end{bmatrix}. \quad (3.5)$$

Since we are dealing with a single-input system, R is a scalar value. Using the Q and R defined, expand the cost function given in Equation 3.4.

3. For the feedback control $u = -Kx$, the Linear-Quadratic Regular algorithm finds a gain K that minimizes the cost function J . Matrix Q sets the weight on the states and determines how u will minimize J (and hence how it generates gain K). From your solution in Question 2, explain how increasing the diagonal elements, q_i , effects the generated gain $K = [k_1 \ k_2 \ k_3 \ k_4]$.
4. Explain the effect of increasing R has on the generated gain, K .


```

if strcmp ( CONTROL_TYPE , 'STUDENT' )
    Torsion_1D_ABCD_eqns;
elseif strcmp ( CONTROL_TYPE , 'INSTRUCTOR' )
    Torsion_1D_ABCD_eqns_student;
end

```

However, you may opt to use the default value of $K_s = 0.5$.

- Go down to the *Control Gain* section shown here:

```

%% Calculate Control Gain
if strcmp ( CONTROL_TYPE , 'STUDENT' )
    % Set Q and R matrices to get desired response.
    Q = diag([1 1 1 1]);
    R = 1;
    [K,S,E] = lqr(A,B,Q,R);

```

The Q and R are initially set to the default values of:

$$Q = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \text{ and } R = 1.$$

- These will not give you the desired response, but run the script to generate the default gain K .

$K =$

39.8762 -38.4620 1.2126 -0.1970

- Run `s_torsion` to simulated the closed-loop response with this gain. See Figure 3.3 for the typical response.

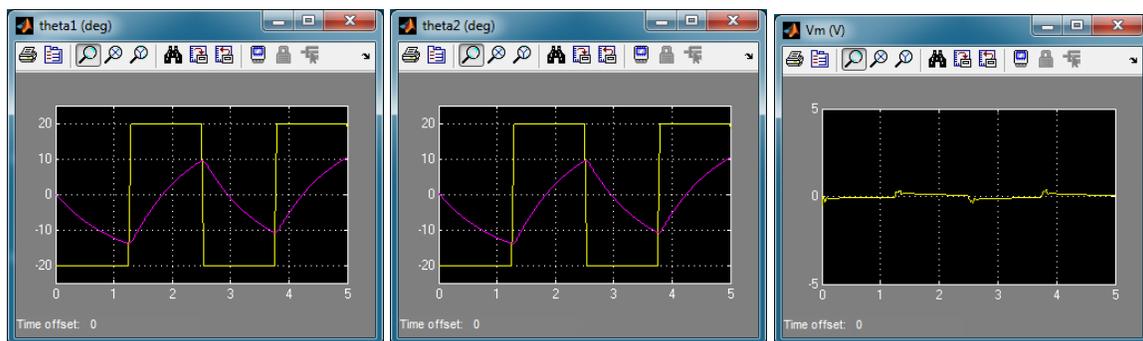


Figure 3.3: Default Simulated Closed-Loop Response

- If $Q = \text{diag}[q_1, q_2, q_3, q_4]$, vary each q_i independently and examine its effect on the gain and the closed-loop response. For example, when increasing q_3 , what happens to θ and α ? Vary each q_i by the same order of magnitude and compare how the new gain K changes compared to the original gain. Keep $R = 1$ throughout your testing. Summarize your results.

Note: Recall your analysis in pre-lab Question 3 where the effect of adjusting Q on the generated K was assessed generally by inspecting the cost function equation. You may find some discrepancies in this exercise and the pre-lab questions.

- Find a Q and R that will satisfy the specifications given in Section 3.1. When doing this, don't forget to keep the dc motor voltage within ± 10 V. This control will later be implemented on actual hardware. Therefore, make sure the actuator is not being saturated. Enter the weighting matrices, Q and R , used and the resulting gain, K .
- Plot the responses from the θ (deg), α (deg), and V_m (V) scopes in a Matlab figure.

Saved response data: When the QUARC controller is stopped, these scopes automatically save the last 5 seconds of their response data to the variables `data_theta1`, `data_theta2`, and `data_vm`. For `data_theta1`, the time is in `data_theta1(:,1)`, the setpoint (i.e., desired SRV02 angle) is in `data_theta1(:,2)`, and the simulated SRV02 angle is in `data_theta1(:,3)`. The same applies for the torsion data saved in `data_theta2`. In the `data_vm` variable, the `(:,1)` holds the time vector and the `(:,2)` holds the input voltage.

- Measure the settling time, percent overshoot, and steady-state error of the simulated torsion response. Does the response satisfy the specifications given in Section 3.1?
- Briefly explain the procedure you used to find Q and R .

3.4.2 Control Implementation

In this experiment, the servo position is controlled while minimizing the vibration of the torsion load using the LQR-based control found in Section 3.4.1. Measurements will then be taken to ensure that the specifications are satisfied.

Experiment Setup

The `q_torsion` Simulink diagram shown in Figure 3.4 is used to run the state-feedback control on the Quanser Rotary 1 DOF Torsion system. The `1 DOF Torsion` subsystem contains QUARC blocks that interface with the DC motor and sensors of the system. The feedback developed in Section 3.4.1 is implemented using a Simulink `Gain` block.

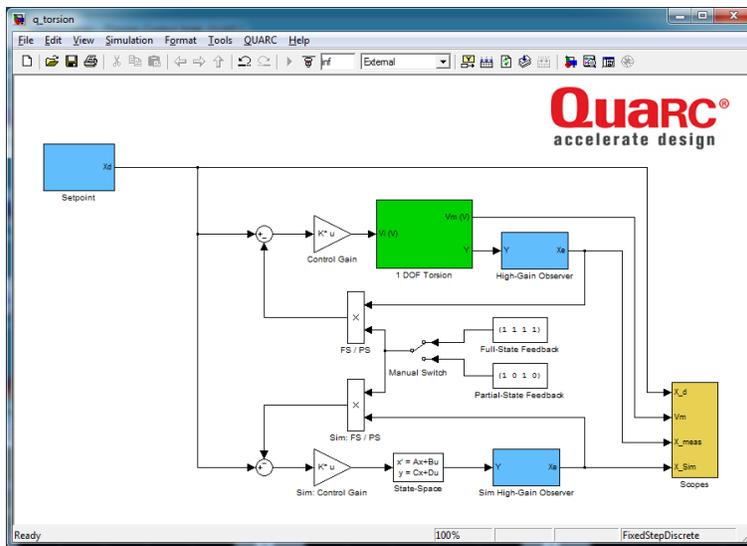


Figure 3.4: Simulink diagram that runs LQR control on 1 DOF Torsion system

IMPORTANT: Before you can conduct this experiment, you need to make sure that the lab files are configured according to your system setup. If they have not been configured already, then go to Section 4.5 to configure the lab files first.

- Run the `setup_torsion.m`.
- Ensure the LQR controller you found in Section 3.4.1 is loaded, i.e., gain K .

3. Open the `q_torsion` Simulink diagram. Make sure the *Manual Switch* is set to the *Full-State Feedback* (upward) position.
4. Go to QUARC | Build to build the controller.
5. Go to QUARC | Start to run the controller. The torsion shaft should start going between ± 10 degree signal.
6. Stop the controller once you have obtained a representative response.
7. Plot the responses from the *theta (deg)*, *alpha (deg)*, and *Vm (V)* scopes in a Matlab figure. Similarly as described in Section 3.4.1, the response data is saved in variables *data_theta*, *data_alpha*, and *data_vm*.
8. Measure the settling time, percent overshoot, and steady-state error of the measured torsion response. Does the response satisfy the specifications given in Section 3.1?

3.4.3 Implementing Partial-State Feedback Control

In this section, the partial-state feedback response of the system is assessed and compared with the full-state feedback control in Section 3.4.2.

1. Run the *setup_torsion.m*.
2. Ensure the LQR control gain you settled on in Section 3.4.2 is loaded, i.e., gain *K*.
3. Open the `q_torsion` Simulink diagram. Make sure the *Manual Switch* is set to the *Partial-State Feedback* (downward) position.
4. Go to QUARC | Start to run the controller. The torsion angle should be tracking the default ± 10 degree signal.
5. Stop the controller once you have obtained a representative response.
6. As in Section 3.4.2, attach a Matlab figure representing the servo and torsion angle response, as well as the input voltage.
7. Examine the difference between the partial-state feedback (PSF) response and the full-state feedback (FSF) response. Explain why PSF control behaves this way by looking at the `q_torsion` Simulink diagram.

3.5 Results

Fill out Table 3.1 with your answers from your control lab results - both simulation and implementation.

Description	Symbol	Value	Unit
LQR Simulation			
LQR Weighting Matrices	Q		
	R		
LQR Gain	K		
Settling time	t_s		s
Percentage overshoot	PO		%
Steady-state error	e_{ss}		deg
LQR Implementation			
LQR Weighting Matrices	Q		
	R		
LQR Gain	K		
Settling time	t_s		s
Percentage overshoot	PO		%
Steady-state error	e_{ss}		deg

Table 3.1: Results

4 SYSTEM REQUIREMENTS

Required Software

- Microsoft Visual Studio (MS VS)
- Matlab[®] with Simulink[®], Real-Time Workshop, and the Control System Toolbox
- QUARC[®]

See the QUARC[®] software compatibility chart in [5] to see what versions of MS VS and Matlab are compatible with your version of QUARC and for what OS.

Required Hardware

- Data-acquisition (DAQ) card that is compatible with QUARC. This includes Quanser Hardware-in-the-loop (HIL) boards such as:
 - Q2-USB
 - Q8-USB
 - QPID
 - QPIDeand some National Instruments DAQ devices (e.g., NI USB-6251, NI PCIe-6259). For a full listing of compliant DAQ cards, see Reference [1].
- Quanser SRV02-ET rotary servo.
- Quanser Rotary 1 DOF Torsion (attached to SRV02).
- Quanser VoltPAQ-X1 power amplifier, or equivalent.

Before Starting Lab

Before you begin this laboratory make sure:

- QUARC[®] is installed on your PC, as described in [3].
- DAQ device has been successfully tested (e.g., using the test software in the Quick Start Guide or the *QUARC Analog Loopback Demo*).
- Rotary 1 DOF Torsion and amplifier are connected to your DAQ board as described Reference [7].

4.1 Overview of Files

File Name	Description
1 DOF Torsion User Manual.pdf	This manual describes the hardware of the Quanser Rotary 1 DOF Torsion system and explains how to setup and wire the system for the experiments.
1 DOF Torsion Workbook (Student).pdf	This laboratory guide contains pre-lab questions and lab experiments demonstrating how to design and implement a position controller on the Quanser Rotary 1 DOF Torsion plant using QUARC® .
setup_torsion.m	The main Matlab script that sets the SRV02 motor and sensor parameters, the SRV02 configuration-dependent model parameters, and the 1 DOF Torsion sensor parameters. Run this file only to setup the laboratory.
config_srv02.m	Returns the configuration-based SRV02 model specifications R_m , k_t , k_m , K_g , η_{a_g} , B_{e_q} , J_{e_q} , and η_{a_m} , the sensor calibration constants K_{POT} , K_{ENC} , and K_{TACH} , and the amplifier limits V_{MAX_AMP} and I_{MAX_AMP} .
config_torsion.m	Returns the 1 DOF Torsion model inertia, J_l , and viscous damping, B_l .
TORSION_1D_ABCD_eqns_student.m	Contains the incomplete state-space A, B, C, and D matrices. These are used to represent the 1 DOF Torsion system.
calc_conversion_constants.m	Returns various conversions factors.
s_torsion.mdl	Simulink file that simulates the 1 DOF Torsion system when using a full or partial state-feedback control.
q_torsion_id.mdl	When ran with QUARC® , this Simulink model feed a Sine Sweep signal to the servo and measures the corresponding 1 DOF Torsion angle. The measured response can then be used to find the natural frequency of the torsion module.
q_torsion_val.mdl	This Simulink model is used with QUARC® to compare the 1 DOF Torsion state-space model with the measured response from the actual system.
q_torsion.mdl	Simulink file that implements a closed-loop state-feedback controller on the actual torsion system using QUARC® .

Table 4.1: Files supplied with the Quanser Rotary 1 DOF Torsion.

4.2 Setup for Finding Stiffness

Before beginning in-lab procedure outlined in Section 2.3.1, the `q_torsion_id` Simulink diagram must be properly configured.

Follow these steps:

1. Setup the SRV02 with the 1 DOF Torsion module as detailed in the 1 DOF Torsion User Manual ([7]).
2. Load **Matlab®**.
3. Browse through the *Current Directory* window in Matlab and find the folder that contains the file `q_torsion_id.mdl`.
4. Open the `q_torsion_id.mdl` Simulink diagram, shown in Figure 2.3.

5. **Configure DAQ:** Ensure the HIL Initialize block subsystem is configured for the DAQ device that is installed in your system. By default, the block is setup for the Quanser Q8 hardware-in-the-loop board. See Reference [1] for more information on configuring the HIL Initialize block.

4.3 Setup for Model Validation

Before performing the in-lab exercises in Section 2.3.3, the `q_torsion_val` Simulink diagram and the `setup_torsion.m` script must be configured.

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

1. Setup the SRV02 with the 1 DOF Torsion module as detailed in [7].
2. Load **Matlab®**.
3. Browse through the *Current Directory* window in Matlab and find the folder that contains the QUARC torsion file `q_torsion_val.mdl`.
4. Open the `q_torsion_val.mdl` Simulink diagram, shown in Figure 2.5.
5. **Configure DAQ:** Ensure the HIL Initialize block in the *SRV02 1 DOF Torsion* subsystem is configured for the DAQ device that is installed in your system. By default, the block is setup for the Quanser Q8 hardware-in-the-loop board. See Reference [1] for more information on configuring the HIL Initialize block.
6. Open the `setup_torsion.m` file. This is the setup script used for the torsion Simulink models.
7. **Configure setup script:** When used with the 1 DOF Torsion, the SRV02 has no load (i.e., no disc or bar) and has to be in the high-gear configuration. Make sure the script is setup to match this setup:
 - EXT_GEAR_CONFIG to 'HIGH'
 - LOAD_TYPE to 'NONE'
 - Ensure ENCODER_TYPE, TACH_OPTION, K_CABLE, AMP_TYPE, and VMAX_DAC parameters are set according to the SRV02 system that is to be used in the laboratory.
 - CONTROL_TYPE to 'STUDENT'.

4.4 Setup for 1 DOF Torsion Control Simulation

Before going through the control simulation in Section 3.4.1, the `s_torsion` Simulink diagram and the `setup_torsion.m` script must be configured.

Follow these steps to configure the lab properly:

1. Load **Matlab®**.
2. **IMPORTANT:** Make sure the model you found in Section 2.3.3 is entered in `TORSION_1D_ABCD_eqns_student.m`.
3. Browse through the *Current Directory* window in Matlab and find the folder that contains the `s_torsion.mdl` file.
4. Open `s_torsion.mdl` Simulink diagram shown in Figure 3.2.
5. Configure the `setup_torsion.m` script according to your hardware. See Section 4.3 for more information.
6. Enter the stiffness (K_s) you found in Section 2.3.1. If you did not find the stiffness, then you can use the default value of $K_s = 1$ but the results may not be optimal.
7. Run the `setup_torsion.m` script.

4.5 Setup for Rotary Torsion Control Implementation

Before beginning the in-lab exercises given in Section 3.4.2 (or Section 3.4.3), the `q_torsion` Simulink diagram and the `setup_torsion.m` script must be setup.

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

1. Before performing the control on the actual system, go through the simulation laboratory described in Section 3.4.1 to find the control gain K .
2. Setup the SRV02 with the 1 DOF Torsion module as detailed in 1 DOF Torsion User Manual [7] .
3. Load **Matlab®**.
4. Browse through the *Current Directory* window in Matlab and find the folder that contains the `q_torsion.mdl` file.
5. Open the `q_torsion.mdl` Simulink diagram shown in Figure 3.4.
6. **Configure DAQ:** Ensure the HIL Initialize block in the *SRV02 1 DOF Torsion* subsystem is configured for the DAQ device that is installed in your system. By default, the block is setup for the Quanser Q8 hardware-in-the-loop board. See Reference [1] for more information on configuring the HIL Initialize block.
7. **Configure setup script:** Set the parameters in the `setup_torsion.m` script according to your system setup. See Section 4.3 for more details.
8. Run the `setup_torsion.m` script. Make sure the gain found in Section 3.4.1 has been loaded.

5 LAB REPORT

This laboratory contains two groups of experiments, namely,

1. Modeling the Quanser 1 DOF Torsion, and
2. State-feedback control.

For each experiment, follow the outline corresponding to that experiment 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 (Modeling)

I. PROCEDURE

1. *Finding Stiffness*

- Briefly describe the main goal of the experiment.
- Briefly describe the experiment procedure in Step 3 in Section 2.3.1.

2. *Model Validation*

- Briefly describe the main goal of the experiment.
- Briefly describe loading the model in Step 6 in Section 2.3.3.
- Briefly describe the model validation experiment in Step 6 in Section 2.3.3.

II. RESULTS

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

1. Step response plot from step 3 in Section 2.3.1.
2. Open-loop poles found in step 9 in Section 2.3.2.
3. Model validation plot from step 6 in Section 2.3.3.
4. Provide applicable data collected in this laboratory (from Table 2.2).

III. ANALYSIS

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

1. Measured torsion module stiffness in step 5 in Section 2.3.1.
2. Model discrepancies given in step 8 in Section 2.3.3.

IV. CONCLUSIONS

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

1. How does the model compare with the actual system in step 7 of Section 2.3.3, *State-space model validation*.

5.2 Template for Content (Control)

I. PROCEDURE

1. *Simulation*

- Briefly describe the main goal of the simulation.
- Briefly describe the procedure in step 6 of Section 3.4.1 to examine the effect of variables on the gain and closed-loop response.
- Briefly describe the simulation procedure in step 8 of Section 3.4.1.
- Briefly explain the procedure used to find Q and R in step 10 of Section 3.4.1.

2. *Full-State Feedback Implementation*

- Briefly describe the main goal of this experiment.
- Briefly describe the experimental procedure in step 7 of Section 3.4.2).

3. *Partial-State Feedback Implementation*

- Briefly describe the main goal of this experiment.
- Briefly describe the experimental procedure in step 6 of Section 3.4.3.

II. RESULTS

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

1. Response plot from step 8 in Section 3.4.1, *Full-state feedback LQR controller simulation*.
2. Response plot from step 7 in Section 3.4.2, for *Full-state feedback LQR controller implementation*.
3. Response plot from step 6 in Section 3.4.3, for *Partial-state feedback LQR controller implementation*.
4. Provide applicable data collected in this laboratory (from Table 3.1).

III. ANALYSIS

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

1. Settling time and percent overshoot in step 9 in Section 3.4.1, *Full-state feedback LQR controller simulation*.
2. Settling time and percent overshoot in step 8 in Section 3.4.2, for *Full-state feedback LQR controller implementation*.
3. Comparison between partial-state and full-state feedback in step 7 in Section 3.4.3.

IV. CONCLUSIONS

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

1. Whether the controller meets the specifications in step 9 in Section 3.4.1, *Full-state feedback LQR controller simulation*.
2. Whether the controller meets the specifications in step 8 in Section 3.4.2, for *Full-state feedback LQR controller implementation*.

5.3 Tips for Report Format

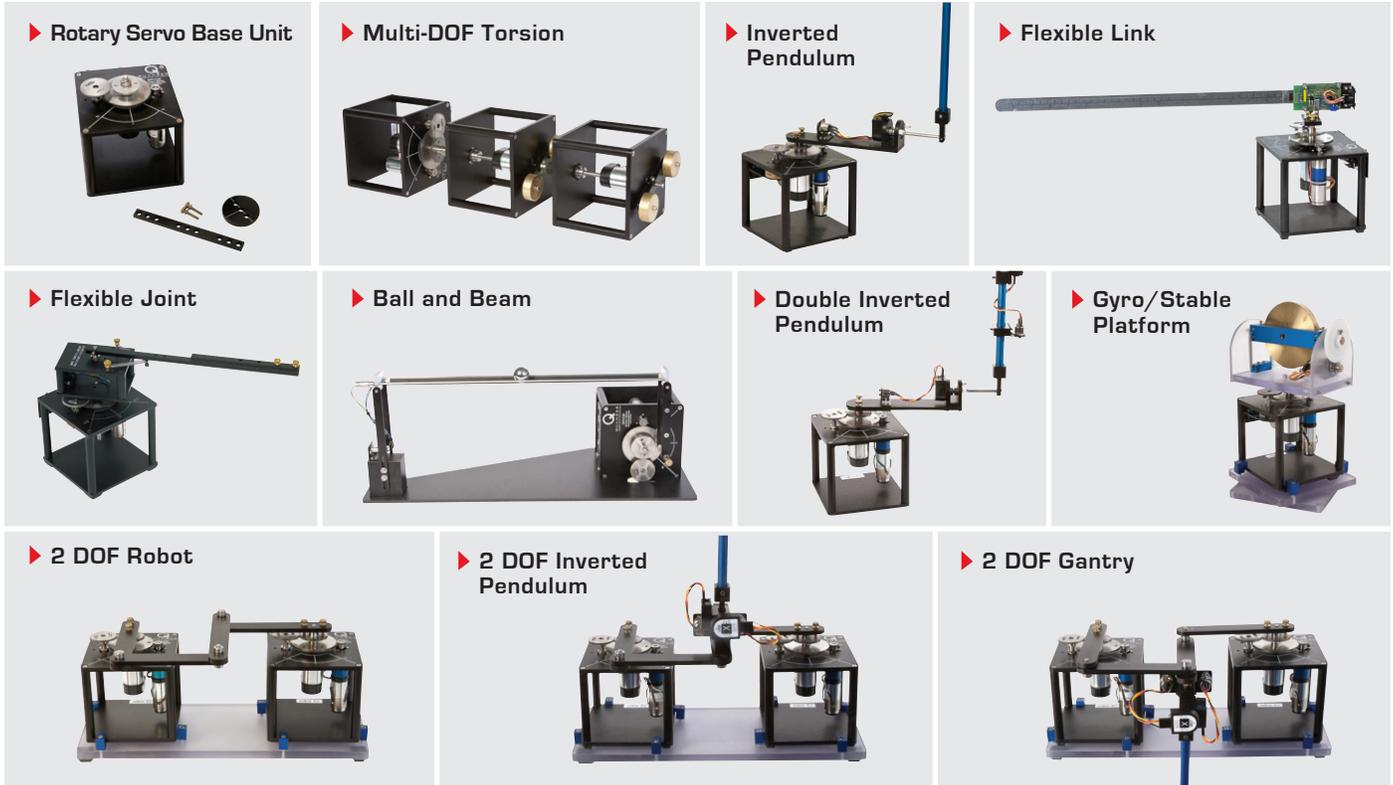
PROFESSIONAL APPEARANCE

- 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.

REFERENCES

- [1] Quanser Inc. *QUARC User Manual*.
- [2] Quanser Inc. *SRV02 QUARC Integration*, 2008.
- [3] Quanser Inc. *QUARC Installation Guide*, 2009.
- [4] Quanser Inc. *SRV02 User Manual*, 2009.
- [5] Quanser Inc. *QUARC Compatibility Table*, 2010.
- [6] Quanser Inc. *SRV02 lab manual*. 2011.
- [7] Quanser Inc. *1 DOF Torsion User Manual*, 2012.
- [8] Norman S. Nise. *Control Systems Engineering*. John Wiley & Sons, Inc., 2008.

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

©2013 Quanser Inc. All rights reserved.



QUANSER
INNOVATE. EDUCATE.

INFO@QUANSER.COM

+1-905-940-3575

QUANSER.COM

Solutions for teaching and research. Made in Canada.