MTHE 430: Modern Control Theory
Lab Manual

May 13th, 2015
Last Edited: June 14th, 2017

Prepared for

Department of Mathematics & Statistics
Queen's University

by Drew Steeves

Under the Supervision of

Abdol-Reza Mansouri and Bahman Gharesifard

INSTRUCTOR'S MANUAL - DO NOT DISTRIBUTE

Contents

I Preface . . . 4
II Lab Safety . . . 5
III Lab 1: Modelling a Rotary Flexible Beam . . . 6
3.1 Introduction . . . 6
3.2 Prelab: Finding the Equations of Motion - 2 DOF Model . . . 6
3.2.1 The Euler-Lagrange Equation . . . 7
3.2.2 Prelab Questions: . . . 8
3.3 Working with Simulink & the Physical System . . . 9
3.3.1 Setting up the Rotary Flexible Beam . . . 9
3.3.2 First Simulink Model . . . 10
3.3.3 Finding the spring torsion of the Flexible Beam Experimentally 12
3.3.4 Model Validation . . . 14
3.4 Higher-Order Model for the Flexible Beam . . . 19
3.5 Deliverables . . . 24
IV Lab 2: Controllability & Observability of a Rotary Pendulum . . . 25
4.1 Introduction . . . 25
4.2 Prelab: Finding the Equations of Motion & State-Space Representation 25
4.2.1 Prelab Questions: . . . 26
4.3 Linearization . . . 27
4.4 Validity of Linearization . . . 29
4.4.1 Setting up the Rotary Pendulum . . . 31
4.4.2 Simulink Model . . . 32
4.5 Controllability & Observability . . . 33
4.6 Deliverables . . . 36
V Lab 3: Stability, Feedback Control & Luenberger Observer for Balancing an Inverted Rotary Pendulum . . . 37
5.1 Introduction . . . 37
5.2 Prelab: Linearization, State-Space Representation & Stability Analysis 37
5.3 Pole Placement & Full State Feedback . . . 41

5.4 Linear Observer 46
5.4.1 Designing & Validating an Observer 48
5.4.2 Testing your Observer on the Physical System 50
5.5 Deliverables 52
VI Lab 4: LQR Optimization for Feedback Control and Observer Design for the Rotary Flexible Beam 53
6.1 Introduction 53
6.2 Prelab: Open-Loop Stability Analysis 53
6.3 Designing Feedback using the Linear Quadratic Regulator 54
6.4 Effects of Partial State Feedback on LQR 62
6.5 Designing an Observer & Using Observer Estimate as Feedback for LQ Problem 64
6.6 Deliverables 67

I Preface

This lab manual contains four labs designed to accompany an undergraduate modern control theory course. These four labs are based on those provided by Quanser for the rotary flexible beam [1] and rotary pendulum [2] systems, yet they differ for numerous reasons. The first difference is the addition of free-body diagrams for the two systems being studied, giving a comprehensive overview of the engineering and mathematics involved in modelling systems. A second difference is the addition of topics including higher-order modelling, minimal realizations and state observers. Much of the content in this lab is original, and all treatments borrowed from Quanser's lab are modified and rearranged to better suit the MTHE 430 course offered at Queen's University in Kingston, Canada.

How to work with this manual: As much as possible, the mathematical theory used in these labs is included, thus this manual is more or less self-contained. Theory sections are differentiated from practical sections via *italic* characters, and questions are differentiated by indentations. There is a Deliverables section at the end of each lab outlining what students should include in their lab reports; note that not all questions asked throughout a lab need to be answered in a student's lab report. The questions that do not need to be answered in a student's lab report are meant to guide the engineering thought process (i.e., these questions usually ask why a certain mathematical theory should hold or fail for a given application).

The development of this lab manual was supported by the Department of Mathematics & Statistics at Queen's University in Kingston, Ontario.

II Lab Safety

Safety and security is paramount when carrying out a lab. Please conduct yourself in a responsible manner at all times during these controls lab. In addition, please follow these safety instructions:

(i) Follow all written and verbal instructions carefully. If you do not understand a part of a procedure, ask your TA for further instructions;

(ii) be alert and proceed with caution at all times in the laboratory;

(iii) these labs may involve rotating disks/beams/pendulums, and as such, participants must clear the area around the Quanser devices before executing any programming;

(iv) as these labs involve electrical equipment (e.g., power amplifiers, etc.), participants should not stick any metal objects into electrical receptacles, and liquids should be kept out of the lab;

(v) notify your TA immediately of any unsafe conditions you observe;

(vi) report any accident (e.g., breakage of equipment or cables, etc.) or injury (cut, etc.) to the TA;

(vii) equipment instructions must be read carefully before use;

(viii) know where the fire alarm and the exits are located;

(ix) work areas should be kept clean and tidy at all times;

(x) do not eat food or drink beverages in the laboratory.

III Lab 1: Modelling a Rotary Flexible Beam

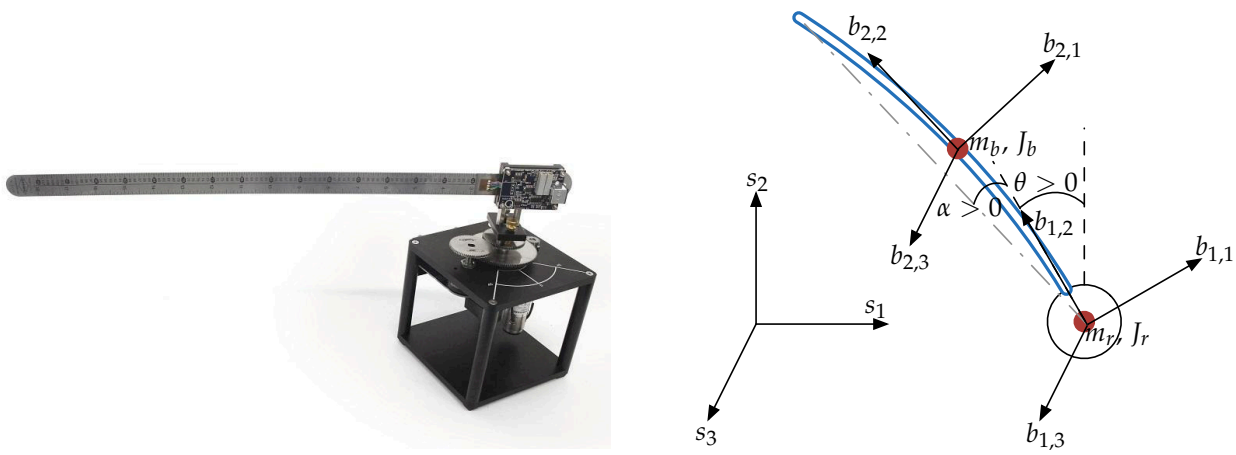
3.1 Introduction

The purpose of this lab is to introduce you to a physical system for which you will develop numerous controllers throughout the semester. This lab will focus on modelling a rotary flexible beam and validating your model by comparing your model's and the physical system's responses to certain inputs.

3.2 Prelab: Finding the Equations of Motion - 2 DOF Model

From previous mechanics courses, you should be familiar with modelling physical systems using first principles (i.e., Newton's Laws). For simple systems, this first principles approach to modelling a physical system is relatively straightforward and works very well; however, for more complicated systems, things can get ugly very quickly. For modelling our rotary flexible beam, it is easier to use a more modern approach to modelling that employs the *Euler-Lagrange equation*.

For a flexible beam, things are further complicated due to the fact that in reality, this system is *infinite-dimensional* as the deflection of the beam changes as a function of the length along the beam. For now, we will study the simplest model of the rotary flexible beam - the 2-degree-of-freedom model. We will deal with a higher order model of the flexible beam in Section 3.4 and compare the accuracies of both of our models.



(a) (b)

Figure 1: (a) Physical rotary flexible beam system [1]; (b) top view of a 2-DOF model of the physical system with one flexible body and one rigid base, where the flexible body and the rotor base both rotate in the horizontal plane. The reference coordinate axes are shown by $\{s_1, s_2, s_3\}$, the rotor base coordinate axes are shown by $\{b_{1,1}, b_{1,2}, b_{1,3}\}$, and the flexible body coordinate axes are shown by $\{b_{2,1}, b_{2,2}, b_{2,3}\}$. The centres of mass for both bodies are shown in red, and the rotation angle, θ, and deflection angle of the beam's tip, α, are shown.

3.2.1 The Euler-Lagrange Equation

Some mathematical theory:

The Euler-Lagrange equation is a second-order partial differential equation that will help you find the equations of motion for our physical system. Before presenting the Euler-Lagrange equation, we must first introduce some mathematical objects and notation that have strong physical interpretations.

To describe the configuration of the physical system relative to some reference configuration, we use coordinates (not to be confused with the coordinate axes shown in Figure 1 - we're talking about variables θ and α here) which we call the *generalized coordinates* of the system, and we denote the ith generalized coordinate by q_i. We define the kinetic and potential energies of the system by functions of the generalized coordinates and their time derivatives, and denote them by T and V, respectively. To summarize the dynamics of our physical system, we use the function L (called the *Lagrangian* of the dynamical system) which is defined by

$$L = T - V.$$

In modelling a physical system, we must also consider external forces applied to the system. We do this by defining a *generalized force* for each generalized coordinate q_i which takes into account the effects of externally applied forces (or torques if q_i is an angle) that are excluded from V (e.g., the force of gravity is a potential force, not a generalized force). We denote the generalized force for q_i by Q_{q_i}. The Euler-Lagrange equation is then given as follows:

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_i}\right) - \frac{\partial L}{\partial q_i} = Q_{q_i}. \tag{1}$$

To find the equation of motion for a particular coordinate q_i, all one must do is find L, V and then evaluate (1).

Let us jump into modelling our rotary flexible beam:

We can define the generalized coordinate vector $q(t)^T = [q_1(t)\ q_2(t)]$ using the rotor base angle, θ, and the flexible beam's tip deflection angle, α, as follows:

$$q(t)^T = [\theta(t)\ \alpha(t)].$$

Next, we can find the generalized forces (here, they are torques) by simple inspection: since the rotor base is actuated and the flexible beam is not, then one can infer that $Q_\theta = \tau$ and $Q_\alpha = 0$, where τ is the torque applied by the servo motor. To account for friction, we add viscous friction torques to the generalized forces, giving $Q_\theta = \tau - B_r\dot{\theta}$ and $Q_\alpha = -B_b\dot{\alpha}$, where B_r and B_b are the viscous damping coefficients for the equivalent rotary system and the flexible beam, respectively.

We now find the kinetic and potential energy functions for our rotary flexible beam. To do this, one could consult the model in Figure 1, calculate the moments of inertia of the flexible beam as well as derive the (radial) velocities of the components of the system at their centres of masses using kinematics. The way we are choosing to model the flexible beam is as a *rigid rod connected to a torsional spring*, which flexes by angle α; this spring energy

can be accounted for in the potential energy term, V. Finding T and V isn't related to the scope of this course, so they are given to you:

$$\begin{cases} T = \frac{1}{2}J_r\dot{\theta}^2 + \frac{1}{2}J_b\left(\dot{\theta}+\dot{\alpha}\right)^2 \\ V = \frac{1}{2}K_b\alpha^2 \end{cases}$$

where K_b is the spring torsion coefficient of the flexible beam, J_r is the equivalent moment of inertia of the rotary system, and J_b is the moment of inertia of the flexible beam about its end. One may be able to deduce T and V by thinking about the model shown in Figure 2. The final steps are left for you to complete ...

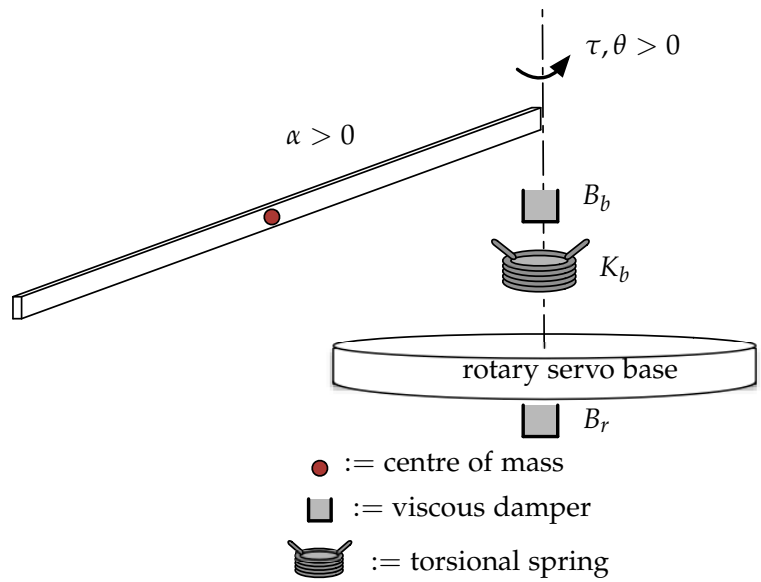


Figure 2: The 2-DOF model used to describe the motion of the rotary flexible beam.

3.2.2 Prelab Questions:

(i) What is the equation of motion for generalized coordinate q_θ?
Answer: it is

$$\left(J_r + J_b\right)\ddot{\theta} + J_b\ddot{\alpha} = \tau - B_r\dot{\theta}.$$

This is obtained via the Euler-Lagrange equation. One calculates:

$$\begin{cases} \frac{\partial L}{\partial \theta} = 0 \\ \frac{\partial L}{\partial \dot{\theta}} = J_r\dot{\theta} + J_b\left(\dot{\theta}+\dot{\alpha}\right) \\ \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta}}\right) = J_r\ddot{\theta} + J_b\left(\ddot{\theta}+\ddot{\alpha}\right) \end{cases}$$

(ii) What is the equation of motion for generalized coordinate q_α?
Answer: it is

$$J_b\left(\ddot{\theta}+\ddot{\alpha}\right) + K_b\alpha = -B_b\dot{\alpha}.$$

This is obtained via the Euler-Lagrange equation. One calculates:

$$\begin{cases} \frac{\partial L}{\partial \alpha} = K_b \alpha \\ \frac{\partial L}{\partial \dot{\theta}} = J_b \left(\dot{\theta} + \dot{\alpha} \right) \\ \frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\theta}} \right) = J_b \left(\ddot{\theta} + \ddot{\alpha} \right) \end{cases}$$

(iii) What is the state-space representation for this system? That is, find A and B and write the EOMs in state-space form.
Hint: Let your state be

$$x(t) = \begin{bmatrix} \theta(t) \\ \alpha(t) \\ \dot{\theta}(t) \\ \dot{\alpha}(t) \end{bmatrix}.$$

Isolate $\ddot{\theta}$ by subtracting your EOM for θ by your EOM for α. You can let $B_b = 0$ as the viscous damping of the beam is negligible.

Important Note: For your (and your team's) benefit, you may want to complete questions Q7 and Q8 before the lab period, as these questions can be answered more easily by using Mathematica. The ILC controls lab computers do not have Mathematica installed at this time.
Answer: the state-space representation is

$$\begin{bmatrix} \dot{\theta}(t) \\ \dot{\alpha}(t) \\ \ddot{\theta}(t) \\ \ddot{\alpha}(t) \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & \frac{K_b}{J_r} & -\frac{B_r}{J_r} & 0 \\ 0 & -\frac{K_b(J_b+J_r)}{J_b J_r} & \frac{B_r}{J_r} & 0 \end{bmatrix} \begin{bmatrix} \theta(t) \\ \alpha(t) \\ \dot{\theta}(t) \\ \dot{\alpha}(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \frac{1}{J_r} \\ -\frac{1}{J_r} \end{bmatrix} \tau$$

3.3 Working with Simulink & the Physical System

Now that you have a state-space representation for the physical system, you can start constructing a Simulink model that will interact with the Quanser SRV02 servo motor in preparation for model validation. Please **save all of your MATLAB and Simulink files** as you may need to consult them in future labs.

3.3.1 Setting up the Rotary Flexible Beam

First, you must assemble and wire the physical system correctly. Examine the close-up assembly shown in Figure 3 and replicate this at your workstation. Note that the high-gear configuration is used here: a large gear is connected to a small pinion and a slip gear, rather than using the smaller gear. You will need to build a two-tiered gear train as shown in Figure 3 to make the large gear fit correctly. Use the wiring diagram shown in Figure 4 to connect the rotary pendulum to a power source and the data acquisition board.

Figure 3: A close-up of the assembly of the rotary flexible beam module and the Quanser SRV02 plant [1]. The high-gear gear train is indicated by a red box.

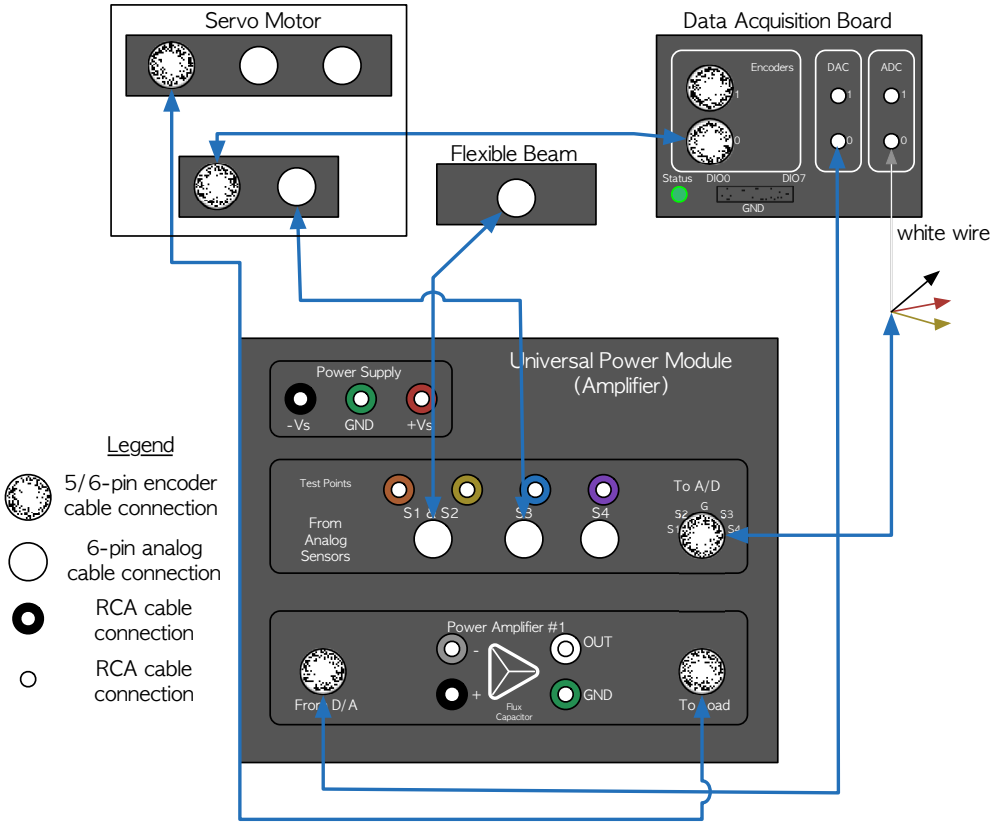


Figure 4: A wiring diagram for the Quanser SRV02 and rotary flexible beam module.

3.3.2 First Simulink Model

You will begin using Simulink by constructing a model including the basic elements required to record the deflection angle of the flexible beam's tip. Here is a step-by-step guide to

accomplish this:

(i) Open MATLAB (**not the 32-bit version**), type and execute the command "simulink" in the Command Window. The Simulink Library Browser should appear. Create a new model.

(ii) Navigate through the Simulink Library Browser and locate the HIL Initialize block (it is located in QUARC Targets — Data Acquisition — Generic — Configuration). Add this block to your new model, and save the model.

(iii) Repeat the same steps to add the HIL Read Analog block (it is located in QUARC Targets — Data Acquisition — Generic — Immediate I/O).

(iv) Now, add a Gain block (located in Simulink — Commonly Used Blocks), a R2D block (located in Simulink Extras — Transformations), a To Workspace block (located in Simulink — Sinks), and a Scope block (located in Simulink — Sinks). From the timebase dropdown menu (next to the time box which is used to input a runtime, and is initially set to *inf*), make sure that External is selected. Also, change *inf* to 9.9.

(v) Construct the model as shown in Figure 5. The HIL Initialize block allows communication between the Quanser SRV02 servo motor, its attached sensors, its added modules and the Simulink model. Double-click on the HIL Initialize block to access its Source Block Parameters. In the Main tab, select "q2_usb" as *Board type* from the options list (USB is the interface we use to connect the Quanser Data Acquisition board to our computer). In the Analog Outputs tab, only check the checkboxes 4, 5 and 6 out of the seven (I'm numbering them in increasing order from the top). In the Digital Outputs tab, enter [0:7] in the *Digital output channels* text box. In the last three text boxes for *Initial digital outputs*, *Final digital outputs* and *Digital outputs watchdog expiry*, enter [0].

(vi) The HIL Read Analog block is used to record the flexible beam's deflection angle, α, via an analog strain gauge sensor located at the end of the beam. Double-click on the HIL Read Analog block to access its Source Block Parameters. Set your *Board name* to HIL-1 to assign this source block to the HIL Initialize block that you previously configured. Next, set your *Channels* to [0], as this is the channel that your strain gauge sensor is connected to.

(vii) The Gain block is used to calibrate strain gauge. Double-click on the Gain block to access its Function Block Parameters. Under the Main tab, set your *Gain* to $\frac{1}{16.5}$. Under the Signal Attributes tab, check the *Saturate on integer flow* check box. Under the Parameter Attributes tab, select "Inherit: Same as Input" in the list of options for *Parameter data type*.

(viii) The Scope block is used to plot the readings of α with respect to time. Rename the block to "alpha (degrees)".

(ix) The To Workspace block is used to send the flexible beam's tip deflection angle to the workspace so that it can be plotted/manipulated in MATLAB. Double-click on the

To Workspace block and under the *Save Format* options list, choose Structure With Time. Rename the block appropriately.

(x) **Before running any Simulink file containing Quanser blocks, you must type in *mex -setup* into the MATLAB command window. You will be prompted to answer three questions in sequence; the answers are: *y*, *1*, and *y*.**

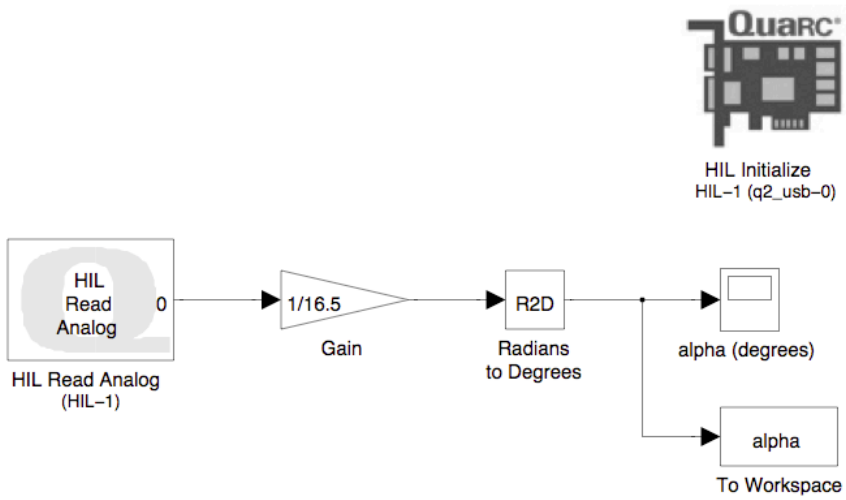


Figure 5: The first Simulink model required for this lab.

Note: The *power amplifier must be turned on* before you can experiment with the physical system. The power switch is located at the back of the amplifier (good luck finding it). Make sure to turn off the power amplifier at your station before you leave the lab.

You won't have to construct all of the Simulink models for each lab, but you will have to add some blocks to modify existing and provided models to make them do exactly what you want. So it will benefit you to have an understanding of how a Simulink model and a physical system interact.

3.3.3 Finding the spring torsion of the Flexible Beam Experimentally

Let us review some vibrations theory:

In the 2-DOF model, we use a spring torsion coefficient, K_b, to characterize the potential energy of the system due to the beam's deflections. In order to find the value of K_b, you can perturb the physical flexible beam while holding the rotor base stationary and study its deflection response. Consider a system response as shown in Figure 6; you can find the average period of the deflection oscillations, T_{osc}, by using

$$T_{\text{osc}} = \frac{t_{n+1} - t_1}{n}, \tag{2}$$

where n is the nth oscillation and t_i is the occurrence time of the ith oscillation, for $i \in \{1, \ldots, n+1\}$. To find T_{osc} using (2), you should always measure oscillation occurrence

times from the same datum. Using the average period, you are able to find the damped natural frequency, ω_d, of the system with

$$\omega_d = \frac{2\pi}{T_{\text{osc}}},$$

and the natural frequency, ω_n is related to ω_d by

$$\omega_n = \frac{w_d}{\sqrt{1-\zeta^2}}, \tag{3}$$

where ζ is the damping ratio of the system. For second-order systems, one can find the damping ratio using the logarithmic decrement equation, but for our flexible beam system $\zeta << 1$ (underdamped case) and thus $\omega_d \approx \omega_n$ by (3). Finally, you can calculate the spring torsion coefficient of the flexible beam by using the relationship

$$K_b = J_b \omega_n^2,$$

which is valid for second-order systems. Recall that the moment of inertia of a thin rod about its end is given by $J_b = \frac{1}{3} m_b L_b^2$.
Back to your lab:

Using the Simulink file that you created in Section 3.3, you can now find K_b given the following constants:

$$\begin{cases} m_b = 0.065\ kg \\ L_b = 0.419\ m \end{cases}$$

You will have to build your Simulink model. To do this, access the QUARC drop-down menu, and select *Build* (or simply type *Ctrl + B*).

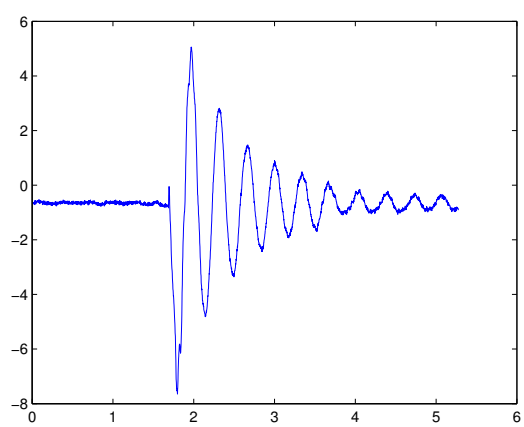


Figure 6: An example of a response of the flexible beam's deflection angle α to a perturbation while rotor base is held stationary.

Q1: What is the natural frequency of the physical flexible beam? Include a plot of the flexible beam response to your perturbation.
Answer: You should get close to

$$T_{\text{osc}} = \frac{3.8 - 2.1}{5} = 0.34s$$

$$\omega_n = \frac{2\pi}{0.34} = 18.5\frac{rad}{s}$$

and the plot should look very similar to Figure 6.

Q2: What is the spring torsion coefficient of the flexible beam, K_b?
Answer: You should get close to

$$K_b = \frac{m_b L_b^2}{3}\omega_n^2 = 1.3\frac{N \cdot m}{rad}$$

3.3.4 Model Validation

You are now in the position to test how accurate your 2-DOF state-space model of the physical system is. To do this, you will compare the response of your state-space model and that of the physical system to an input.

You will start off by modifying an existing Simulink model by adding your state-space model in parallel to the existing physical system model. Open the file **model_validation.mdl** and examine the existing model, shown in Figure 7. This Simulink model can instruct the physical system's servo motor to track a step function; while tracking the input, the servo motor encoder and strain gauge readings will measure and output the rotor angle and beam tip's deflection angle with respect to time.

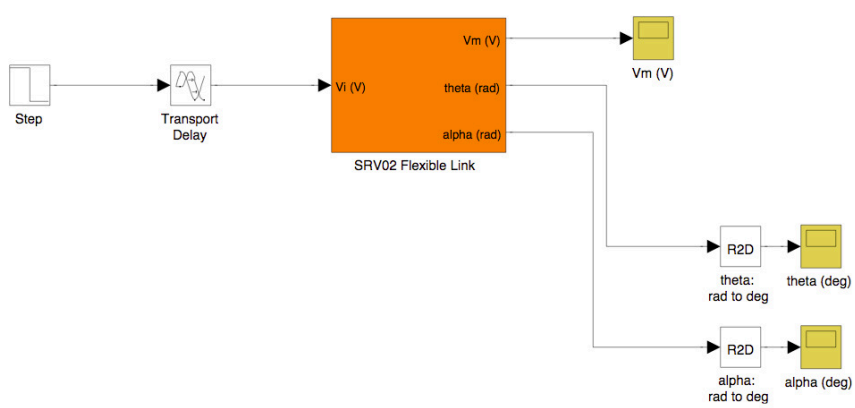


Figure 7: A screenshot of the exisiting Simulink model, **model_validation.mdl**.

You wish to compare the responses of both the physical system and state-space model to a step input. To do this, you will need to add a State-Space block (located in Simulink

— Continuous). You will need to feed your state-space block with the same step input as is being fed to your physical system. You will eventually enter in your state-space matrices A, B, C and D in the MATLAB workspace, so in order for your State-Space block to recognize these in Simulink, set its Function Block Parameters for A, B, C and D as themselves (i.e., A: as "A"). Furthermore, you'll need to output the states θ and α (i.e., states 1 and 2 of the four states) from your state-space block to the existing scope blocks for θ and α. To access only states 1 and 2 from your State-Space block, you can make use of the Demux block (located in Simulink — Commonly Used Blocks), which splits vector signals into scalars or smaller vectors. To join alike states from your state-space model and your physical model, you can use the Mux block, which works like the inverse of the Demux block.
The correct Simulink model looks something like:

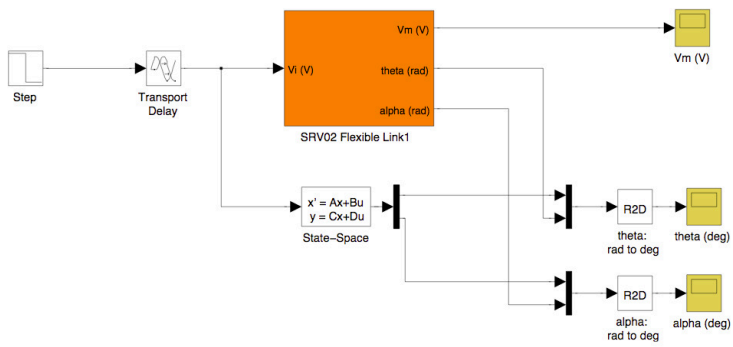


Figure 8: A screenshot of the correctly-modified Simulink model

Now, you must evaluate the symbolic matrices you found for A and B, which involves finding the equivalent moment of inertia for the rotary flexible beam system. Finding J_r is quite a challenging problem: it involves finding the external gears inertias, the motor equivalent inertia, the motor shaft inertia, the load inertia (in this case, the load is the flexible beam module), and relating these parameters via gear ratios while considering gearbox efficiency. Furthermore, finding the equivalent viscous damping for the system is usually done experimentally. For these reasons, the values for J_r and B_r are given here:

$$\begin{cases} J_r = 0.0020841\ kg \cdot m^2 \\ J_b = 0.0038038\ kg \cdot m^2 \\ B_r = 0.015\ \frac{N \cdot s}{m} \\ B_b = 0\ \frac{N \cdot s}{m}\ \text{(negligible beam damping)} \end{cases}$$

Recall that the moment of inertia of the flexible beam about its end is given by $J_b = \frac{1}{3} m_b L_b^2$ (by assuming that the flexible beam can be modelled as an infinitely thin but rigid rod), so you can calculate this explicitly to evaluate your state-space matrices. Another consideration is that the servo motor actuator is controlled by voltage and not by torque (the resulting motor torque is a function of the voltage applied by the servo motor). To transform the state-space model from your prelab in terms of voltage, one must do the following in sequence:

Step 1: let matrix entry $A(3,3) = A(3,3) - \eta_g \eta_m \cdot \frac{K_g^2 k_t k_m}{R_m} B(3)$, where $K_g = 70$ is the total gear ratio for your setup, $k_t = 0.0077$ is the servo motor torque constant, $k_m = 0.0077$

is the motor back-EMF constant, $\eta_g = 0.90$ is the gear efficiency, $\eta_m = 0.69$ is the motor efficiency, and $R_m = 2.6$ is the motor armature resistance;

Step 2: let matrix entry $A(4,3) = A(4,3) - \eta_g \eta_m \cdot \frac{K_g^2 k_t k_m}{R_m} B(4)$;

Step 3: only now should you modify B to $B = \eta_g \eta_m \cdot \frac{K_g k_t}{R_m} B$.

We've prepared a MATLAB script for you to speed up the process: open **lab1_statespace_student**, input your symbolic matrices that you computed in your prelab, and run the script to evaluate your state-space matrices A and B.

Q3: What are the numerically-evaluated matrices A and B for the state-space model?
Answer: You should get

$$A = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 623.77 & -40.49 & 0 \\ 0 & -965.53 & 40.49 & 0 \end{bmatrix}$$

and

$$B = \begin{bmatrix} 0 \\ 0 \\ 61.775 \\ -61.775 \end{bmatrix}.$$

Q4: Given that the physical system's sensors are limited to reading the rotor angle and the deflection angle, what should the numerical matrices C and D for the state-space model be? Pay attention to matrix dimensions here.
Answer: You should get

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

and

$$D = \begin{bmatrix} 0 \\ 0 \end{bmatrix},$$

which can be discerned from the state-space equation $y = Cx + Du$.

With your modified Simulink model and evaluated matrices, you can now build your Simulink model and see whether your state-space model accurately describes the physical system. Make sure that you have entered A, B, C, and D into the MATLAB command window and set the run time in the top toolbar of your Simulink model to 9.9. Now, select the *Quarc* drop-down menu from the main toolbar and click Build. Make sure that both signals (i.e., the physical system's and the state-space model's) are converted to degrees via the *R2D* Simulink block. When you are ready, click Run, which can be found in the same drop-down menu. Plot the θ and α arrays with respect to time, and include these plots in your lab report.

Q5: How well does your model represent the physical system? Give one good reason why the model doesn't have an identical response to that of the physical system. Answer: The model represents the system quite well. The flexible beam model's response (i.e., α-time plot) matches the physical system's response very well for the first few oscillations, but doesn't capture the lower amplitude oscillations and the higher frequency oscillations. This is due to the fact that the flexible beam is modelled as a torsional spring & rigid beam and not as a higher-order nonlinear beam. The rotor base model's response (i.e., θ-time plot) has a steady-state error which is probably due to Coulomb friction or not having an accurate B_r. The plots should look something like this:

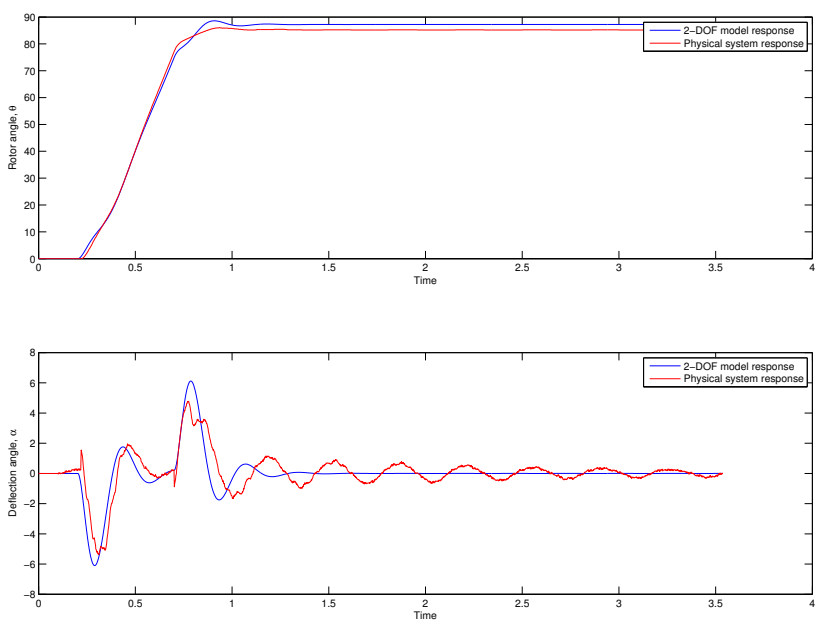
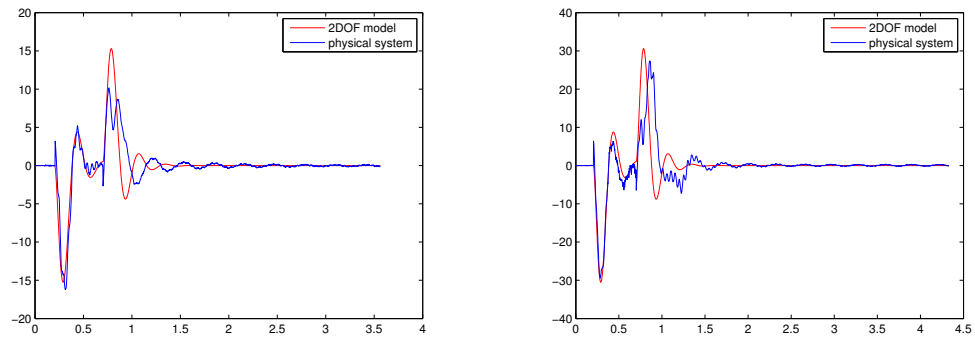


Figure 9: The correct plots of θ and α w.r.t. time for the model & physical system response to a step input.

Recall that we used a *linear spring* to model the flexible beam in our system, and by doing so we were able to use $V = \frac{1}{2}K_b\alpha^2$ to characterize the system's potential energy. This is like making a linear approximation as the flexible beam is likely to deflect nonlinearly and as a function of the length along the beam. To test where your approximation and hence your state-space model is valid, you will use larger step inputs and compare the responses of your model and of the physical system.

Q6: How accurate is your state-space model when the input step response is increased from 2 to 3 as an initial value? What about increasing the input step response's initial value from 2 to 4? Give a plot of the responses of α of your model and the physical system.

The higher frequency vibrations of the physical system are more prevalent and are not captured by the 2-DOF model. Hence the model is less accurate with a larger input. The plots should look like:



(a) (b)

Figure 10: The correct plot of θ and α w.r.t. time for the model & physical system response to a step input with initial value of a) 5 and b) 10.

3.4 Higher-Order Model for the Flexible Beam

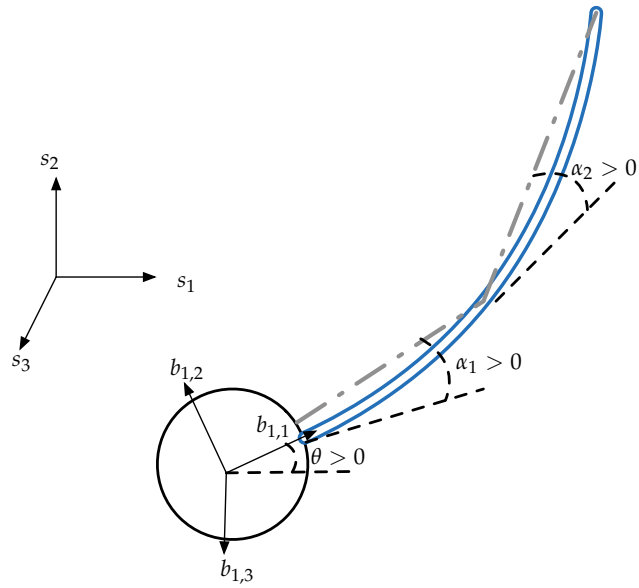


Figure 11: Top view of a 3-DOF model of the rotary flexible beam. The reference coordinate axes are shown by $\{s_1, s_2, s_3\}$, the rotor base coordinate axes are shown by $\{b_{1,1}, b_{1,2}, b_{1,3}\}$, and the two flexible beam body coordinate axes are left out for clarity. The rotation angle, θ, and the deflection angles, α_1 and α_2, are shown.

As was discussed earlier, a realistic model for a flexible beam involves an infinite-dimensional system. As modelling an infinite-dimensional system is an intractable problem from an engineering standpoint, one can approximate this system by breaking it up into a set of finite elements with solvable problems. You will do this with the rotary flexible beam system by introducing another deflection angle, making the system a 3-DOF system and changing the generalized coordinate vector to

$$q(t) = \begin{bmatrix} \theta(t) \\ \alpha_1(t) \\ \alpha_2(t) \end{bmatrix},$$

where α_1 measures the deflection angle from the starting point to the midpoint of the beam, and α_2 measures the deflection angle from the midpoint to the end of the beam. Refer to Figure 11 for a schematic of these deflection angles. As was done for the 2-DOF system, we will use linear springs and account for viscous damping for each half-element of the beam. The kinetic and potential energies for the 3-DOF system are given:

$$\begin{cases} T = \frac{1}{2}J_r\dot{\theta}^2 + \frac{1}{2}J_{b_1}\left(\dot{\theta} + \dot{\alpha}_1\right)^2 + \frac{1}{2}J_{b_2}\left(\dot{\theta} + \dot{\alpha}_1 + \dot{\alpha}_2\right)^2 \\ V = \frac{1}{2}K_{b_1}\alpha_1^2 + \frac{1}{2}K_{b_2}\alpha_2^2 \end{cases}$$

where J_{b_1} and J_{b_2} are the moment of inertias of the first half-element and the second half-element about their ends, and K_{b_1} and K_{b_2} are the spring torsion coefficients of the first and second half-elements, respectively. In this higher-order system, the generalized force

(torques here) for coordinate θ remains the same, and those for coordinates α_1 and α_2 can be reasoned out. They are given:

$$\begin{cases} Q_\theta = \tau - B_r\dot{\theta} \\ Q_{\alpha_1} = -B_{b_1}\dot{\alpha}_1 \\ Q_{\alpha_2} = -B_{b_2}\left(\dot{\alpha}_1 + \dot{\alpha}_2\right). \end{cases}$$

One can deduce that $J_{b_1} = J_{b_2} = \frac{1}{4}J_b$, as these are symmetric half-elements of the flexible

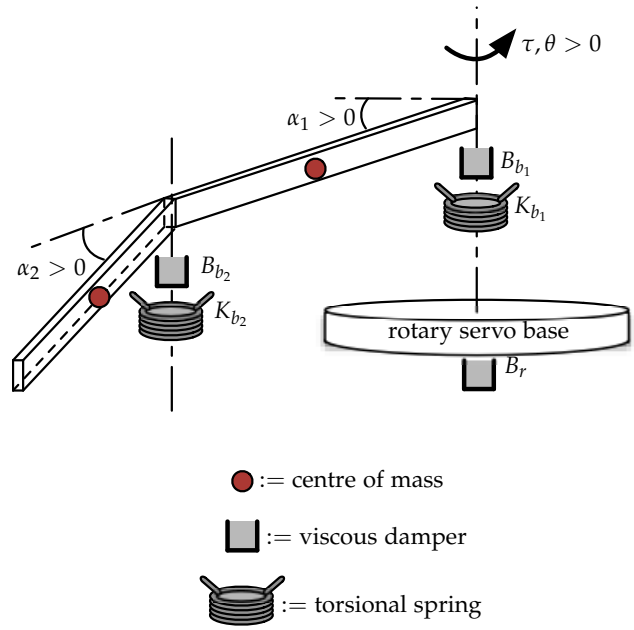


Figure 12: The 3-DOF model used to describe the motion of the rotary flexible beam.

beam. Applying the angular form of Hooke's Law, $T = K_b\alpha$, where T is the torque exerted by the spring, the spring torsion coefficients can be related: $K_{b_1} = K_{b_2}$, where $K_{b_1} = 2K_b$, as we are using linear springs. One can also use a result from the study of vibrations for underdamped systems to relate the viscous damping coefficients of both half-elements:

$$\zeta = \frac{B}{2\sqrt{J_b K}}$$

where ζ is the damping ratio of the flexible beam and J_b is the moment of inertia of the flexible beam. The viscous damping coefficients are then related as follows: $B_{b_1} = B_{b_2}$, where $B_{b_1} = \sqrt{2}B_b$. You may want to consult Figure 12 to intuitively think about this 3-DOF model. You now have enough information to find the EOMs for this system ....

Q7: What are the three equations of motion for the higher-order system?
Hint: Isolate $\ddot{\theta}$ by subtracting your EOM for θ by your EOM for α_1 (similar to what you did in the prelab). You can let $B_{b_1} = B_{b_2} = \sqrt{2}B_b = 0$ as the viscous damping of the beam is negligible. **You may want to use Mathematica** to

speed up the process of isolating these EOMs for $\ddot{\theta}$, $\ddot{\alpha}_1$ and $\ddot{\alpha}_2$.
Answer: the EOMs are

$$\begin{cases} \ddot{\theta} = \frac{K_{b_1}}{J_r}\alpha_1 + \frac{B_{b_1}}{J_r}\dot{\alpha}_1 + -\frac{B_r}{J_r}\dot{\theta} + \frac{\tau}{J_r} \\ \ddot{\alpha}_1 = \left(\frac{-K_{b_1}}{J_r} - \frac{2K_{b_1}}{J_b}\right)\alpha_1 + \frac{2K_{b_2}}{J_b}\alpha_2 + \left(\frac{-B_{b_1}}{J_r} - \frac{2B_{b_1}-2B_{b_2}}{J_b}\right)\dot{\alpha}_1 + \frac{2B_{b_2}}{J_b}\dot{\alpha}_2 - \frac{\tau}{J_r} \\ \ddot{\alpha}_2 = \frac{2K_{b_1}}{J_b}\alpha_1 - \frac{4K_{b_2}}{J_b}\alpha_2 + \frac{2B_{b_1}-4B_{b_2}}{J_b}\dot{\alpha}_1 - \frac{4B_{b_2}}{J_b}\dot{\alpha}_2 \end{cases}$$

To find these, one can use the Euler-Lagrange equations. We first compute the following:

$$\begin{cases} \frac{\partial L}{\partial \theta} = 0 \\ \frac{\partial L}{\partial \alpha_1} = -K_{b_1}\alpha_1 \\ \frac{\partial L}{\partial \alpha_2} = -K_{b_2}\alpha_2 \\ \frac{\partial L}{\partial \dot{\theta}} = J_r\dot{\theta} + J_{b_1}(\dot{\theta} + \dot{\alpha}_1) + J_{b_2}(\dot{\theta} + \dot{\alpha}_1 + \dot{\alpha}_2) \\ \frac{\partial L}{\partial \dot{\alpha}_1} = J_{b_1}(\dot{\theta} + \dot{\alpha}_1) + J_{b_2}(\dot{\theta} + \dot{\alpha}_1 + \dot{\alpha}_2) \\ \frac{\partial L}{\partial \dot{\alpha}_2} = J_{b_2}(\dot{\theta} + \dot{\alpha}_1 + \dot{\alpha}_2) \\ \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta}}\right) = J_r\ddot{\theta} + J_{b_1}(\ddot{\theta} + \ddot{\alpha}_1) + J_{b_2}(\ddot{\theta} + \ddot{\alpha}_1 + \ddot{\alpha}_2) \\ \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\alpha}_1}\right) = J_{b_1}(\ddot{\theta} + \ddot{\alpha}_1) + J_{b_2}(\ddot{\theta} + \ddot{\alpha}_1 + \ddot{\alpha}_2) \\ \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\alpha}_2}\right) = J_{b_2}(\ddot{\theta} + \ddot{\alpha}_1 + \ddot{\alpha}_2) \end{cases}$$

This yields the following EOMs:

$$\begin{cases} J_r\ddot{\theta} + J_{b_1}(\ddot{\theta} + \ddot{\alpha}) + J_{b_2}(\ddot{\theta} + \ddot{\alpha}_1 + \ddot{\alpha}_2) = \tau - B_r\dot{\theta} \\ J_{b_1}(\ddot{\theta} + \ddot{\alpha}) + J_{b_2}(\ddot{\theta} + \ddot{\alpha}_1 + \ddot{\alpha}_2) + K_{b_1}\alpha_1 = -B_{b_1}\dot{\alpha}_1 \\ J_{b_2}(\ddot{\theta} + \ddot{\alpha}_1 + \ddot{\alpha}_2) + K_{b_2}\alpha_2 = -B_{b_1}(\dot{\alpha}_1 + \dot{\alpha}_2). \end{cases}$$

To isolate these EOMs in terms of $\ddot{\theta}, \ddot{\alpha}_1, \ddot{\alpha}_2$, subtract the first equation by the second to yield the first isolated EOM:

$$\ddot{\theta} = \frac{K_{b_1}}{J_r}\alpha_1 + \frac{B_{b_1}}{J_r}\dot{\alpha}_1 + -\frac{B_r}{J_r}\dot{\theta} + \frac{\tau}{J_r}.$$

Next, substitute this EOM (isolated for $\ddot{\theta}$) into the third equation, and then substitute the second equation thereafter to get:

$$\ddot{\alpha}_1 = \left(\frac{-K_{b_1}}{J_r} - \frac{2K_{b_1}}{J_b}\right)\alpha_1 + \frac{2K_{b_2}}{J_b}\alpha_2 + \left(\frac{-B_{b_1}}{J_r} - \frac{2B_{b_1} - 2B_{b_2}}{J_b}\right)\dot{\alpha}_1 + \frac{2B_{b_2}}{J_b}\dot{\alpha}_2 - \frac{\tau}{J_r}.$$

You can now solve for the third EOM (isolating for $\ddot{\alpha}_2$) by substituting. If you followed these steps, you now know how tedious engineering could be...

Q8: What is the state-space representation for this system? Please include A, B, C and D with symbolic entries.

Answer: your state vector should be

$$\begin{bmatrix} \theta \\ \alpha_1 \\ \alpha_2 \\ \dot{\theta} \\ \dot{\alpha}_1 \\ \dot{\alpha}_2 \end{bmatrix}$$

and your resulting symbolic matrices for A and B should be

$$A = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & \frac{K_{b_1}}{J_r} & 0 & 0 & \frac{B_{b_1}}{J_r} & 0 \\ 0 & \left(\frac{-K_{b_1}}{J_r} - \frac{2K_{b_1}}{J_b}\right) & \frac{2K_{b_2}}{J_b} & 0 & \left(\frac{-B_{b_1}}{J_r} - \frac{2B_{b_1}}{J_b} + \frac{2B_{b_2}}{J_b}\right) & \frac{2B_{b_2}}{J_b} \\ 0 & \frac{2K_{b_1}}{J_b} & \frac{-4K_{b_2}}{J_b} & 0 & \frac{2B_{b_1} - 4B_{b_2}}{J_b} & \frac{-4B_{b_2}}{J_b} \end{bmatrix},$$

$$B = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{1}{J_r} \\ \frac{-1}{J_r} \\ 0 \end{bmatrix}.$$

As we wish to output all three states, we have that

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix},$$

$$D = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}.$$

Recall that the state-space matrices must be transformed to account for a voltage input rather than a torque input, as was done for the 2-DOF system. Using the state-space model that you came up with, you can now create a Simulink model to simulate your state-space model's response to a step function, as was done in Section 3.3.4. You may build a Simulink model similar to that shown in Figure 13, following the steps and using the same input as in Section 3.3.4. You will again have to follow the steps to convert the model from a torque input to a voltage input (except now the matrix indices are slightly different as there are two more states). You will compare the 3-DOF model's responses to all three step inputs that were used in Section 3.3.4.

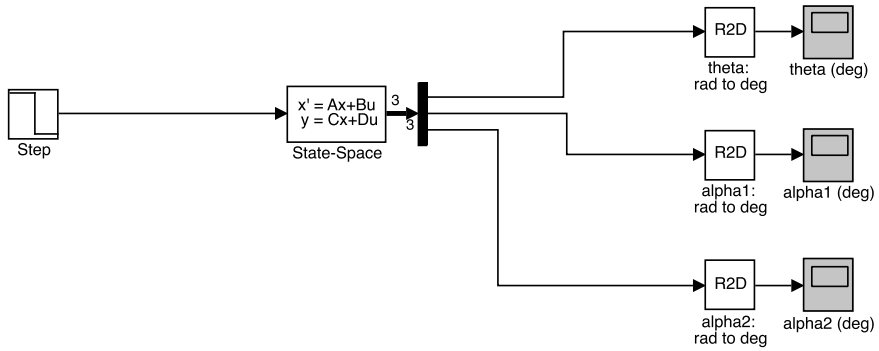


Figure 13: A candidate Simulink model to simulate and output the response of the 3-DOF system to the step input used for the 2-DOF system.

Q9: How will you compare the 3-DOF model's response to the physical system's response as was found in Section 3.3.4, using your Simulink data?
Answer: You should add α_1 and α_2, and then compare this value's response to that of the physical system's.

Q10: Compare the 2-DOF and 3-DOF model's accuracy by comparing their individual responses to that of the physical system, as was found in Section 3.3.4. Do this for all three step inputs, i.e., step inputs starting at 2, 3 and 4. Which model is more accurate for these step inputs? Does modelling a physical system with a higher-order model ensure better accuracy?
Answer: In some instances, the 3-DOF model is slightly more accurate than the 2-DOF model in describing the motion of the flexible beam **when** $B_{b_1} \neq 0$. This makes sense physically, because one needs a nonzero damping term on the first-half of the flexible beam to ensure that the second-half of the flexible beam not go unstable. A higher-order model is not always better: one must verify the physical feasibility of a model and make sure that the additional discretization be representative of the physical system.

Please include plots of your 3-DOF model's responses to the three step input, plotting your comparison variable from Q9 with respect to time.

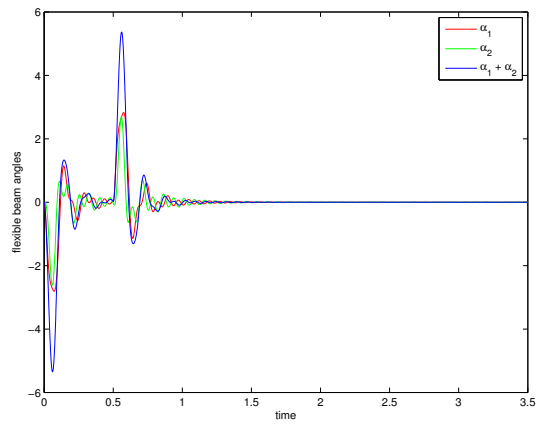


Figure 14: A plot of the response in the deflection angles, α_1 and α_2, due to a step input. The deflection angles are added together to compare the 2-DOF tip deflection to the 3-DOF tip deflection.

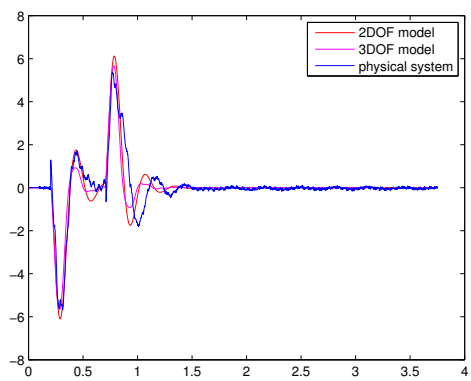


Figure 15: A comparison of the 2-DOF model, 3-DOF model and physical system responses to a step input with initial value of 2.

3.5 Deliverables

To recap the lab objectives, this is what you need to include in your lab report to obtain full marks:

(i) prelab question (i) through (iii);

(ii) the answers to questions Q2, Q5, Q6, Q8 and Q10 (include plots where requested);

Note: Save your lab report and have an electronic copy on-hand in all future labs, as you will need some of the values obtained in this lab to build upon this physical system.

IV Lab 2: Controllability & Observability of a Rotary Pendulum

4.1 Introduction

The purpose of this lab is to study a new system and to investigate the controllability and observability of this system. The prelab will focus on modelling a rotary pendulum using the same modern approach as was applied in Lab III. In the lab, you will move on to study whether the rotary pendulum is controllable or observable, using both a mathematical and an experimental approach.

4.2 Prelab: Finding the Equations of Motion & State-Space Representation

Rather than using first principles to model the rotary pendulum, you will use the Euler-Lagrange equations and follow the same treatment as in Section 3.2 of Lab 1. Recall that the Lagrangian of a system is given by

$$L = T - V.$$

For the rotary pendulum, you can define the generalized coordinate vector using the rotary shaft angle, θ, and the pendulum rod angle, α, as $q(t)^T = [\theta(t)\ \alpha(t)]$. To find the equations of motion for θ and α, you may use the Euler-Lagrange equations:

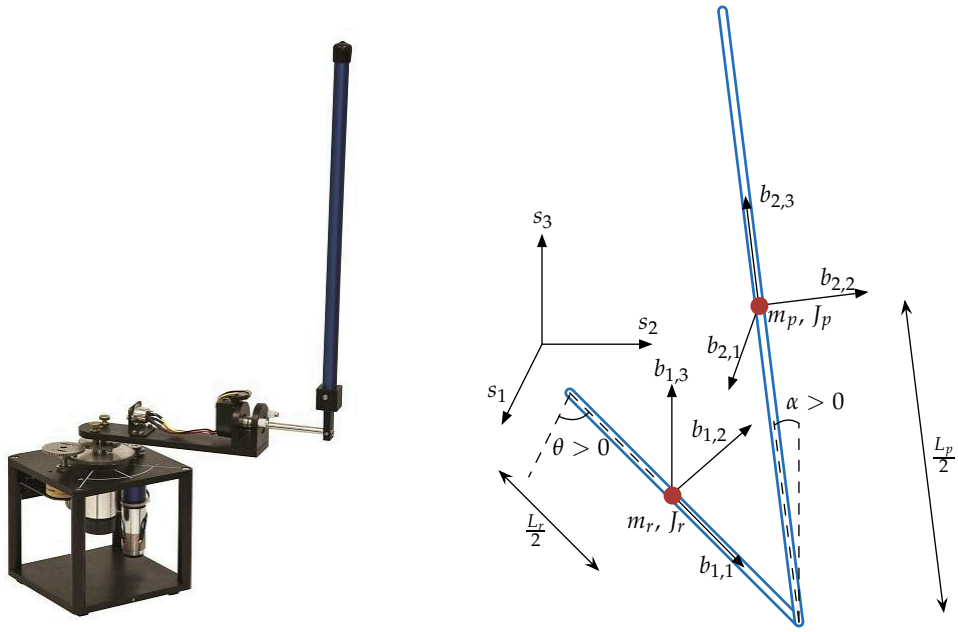
$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_i}\right) - \frac{\partial L}{\partial q_i} = Q_{q_i}.$$

As was done in Lab III, you may logic-out the generalized forces Q_θ and Q_α: since the rotary shaft is actuated and the pendulum rod is not, and accounting for viscous friction, one can infer that $Q_\theta = \tau - B_r\dot{\theta}$ and $Q_\alpha = -B_p\dot{\alpha}$, where τ is the torque applied by the servo motor, and B_r, B_p are the viscous damping coefficients for the rotary shaft and pendulum rod, respectively. As before, deriving the kinetic energy, T, and the potential energy, V, is not in the scope of this course. They are provided for this reason:

$$\begin{cases} T = \left(\frac{1}{2}m_pL_r^2 + \frac{1}{8}m_pL_p^2 - \frac{1}{8}m_pL_p^2\cos^2(\alpha) + \frac{1}{2}J_r\right)\dot{\theta}^2 + \left(\frac{1}{2}J_p + \frac{1}{8}m_pL_p^2\right)\dot{\alpha}^2 - \frac{1}{2}m_pL_pL_r\cos(\alpha)\dot{\theta}\dot{\alpha} \\ V = \frac{1}{2}m_pL_pg\cos(\alpha) \end{cases}$$

where J_p, m_p, and L_p are the moment of inertia about the centre of mass, mass, and length of the pendulum, respectively; J_r, m_r, and L_r are the moment of inertia about the centre of mass, mass, and length of the rotary shaft, respectively; and g is the acceleration due to gravity (on earth). Students wishing to find T and V on their own can do so using the model shown in Figure 16. The complicated terms in T come from the expression for the velocity of the pendulum at its centre of mass (you can find this velocity expression by using kinematics).

The final steps are left for you to complete ...



(a) (b)

Figure 16: (a) Physical rotary pendulum system [1]; (b) model of the physical system with two rigid bodies connected together, where rigid body 1 (rotary shaft) rotates in the horizontal plane and rigid body 2 (pendulum rod) rotates in the vertical plane. The reference coordinate axes are shown by $\{s_1, s_2, s_3\}$, and the body coordinate axes for rigid bodies 1 and 2 are shown by $\{b_{1,1}, b_{1,2}, b_{1,3}\}$ and $\{b_{2,1}, b_{2,2}, b_{2,3}\}$, respectively. The centres of mass for bodies 1 and 2 are shown in red; the horizontal rotation angle, θ, the vertical rotation angle, α, are shown. Important distances are labelled.

4.2.1 Prelab Questions:

(i) What is the equation of motion for generalized coordinate θ?
Answer: it is

$$\left(m_p L_r^2 + \frac{1}{4} m_p L_p^2 - \frac{1}{4} m_p L_p^2 \cos^2(\alpha) + J_r\right) \ddot{\theta} - \frac{1}{2} m_p L_p L_r \cos(\alpha) \ddot{\alpha}$$
$$+ \frac{1}{2} m_p L_p^2 \sin(\alpha) \cos(\alpha) \dot{\theta} \dot{\alpha} + \frac{1}{2} m_p L_p L_r \sin(\alpha) \dot{\alpha}^2 = \tau - B_r \dot{\theta}$$

This is obtained via the Euler-Lagrange equation. One calculates:

$$\begin{cases} \frac{\partial L}{\partial \theta} = 0 \\ \frac{\partial L}{\partial \dot{\theta}} = \left(m_p L_r^2 + \frac{1}{4} m_p L_p^2 - \frac{1}{4} m_p L_p^2 \cos^2(\alpha) + J_r\right) \dot{\theta} - \frac{1}{2} m_p L_p L_r \cos(\alpha) \dot{\alpha} \\ \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta}}\right) = \left(m_p L_r^2 + \frac{1}{4} m_p L_p^2 - \frac{1}{4} m_p L_p^2 \cos^2(\alpha) + J_r\right) \ddot{\theta} + \frac{1}{2} m_p L_p^2 \sin(\alpha) \cos(\alpha) \dot{\alpha} \dot{\theta} \\ + \frac{1}{2} m_p L_p L_r \sin(\alpha) \dot{\alpha}^2 - \frac{1}{2} m_p L_p L_r \cos(\alpha) \ddot{\alpha} \end{cases}$$

which yields the above EOMs.

(ii) What is the equation of motion for generalized coordinate α?
Answer: it is

$$-\frac{1}{2}m_pL_pL_r\cos(\alpha)\ddot{\theta}+\left(J_p+\frac{1}{4}m_pL_p^2\right)\ddot{\alpha}-\frac{1}{4}m_pL_p^2\sin(\alpha)\cos(\alpha)\dot{\theta}^2$$
$$-\frac{1}{2}m_pL_pg\sin(\alpha)=-B_p\dot{\alpha}$$

This is obtained via the Lagrange calculations. One calculates:

$$\begin{cases}\frac{\partial L}{\partial\alpha}=\frac{1}{4}m_pL_p^2\cos(\alpha)\sin(\alpha)\dot{\theta}^2+\frac{1}{2}m_pL_pL_r\sin(\alpha)\dot{\theta}\dot{\alpha}+\frac{1}{2}m_pL_pg\sin\alpha\\ \frac{\partial L}{\partial\dot{\alpha}}=\left(J_p+\frac{1}{4}m_pL_p^2\right)\dot{\alpha}-\frac{1}{2}m_pL_pL_r\cos(\alpha)\dot{\theta}\\ \frac{d}{dt}\left(\frac{\partial L}{\partial\dot{\alpha}}\right)=\left(J_p+\frac{1}{4}m_pL_p^2\right)\ddot{\alpha}+\frac{1}{2}m_pL_pL_r\sin(\alpha)\dot{\theta}\dot{\alpha}-\frac{1}{2}m_pL_pL_r\cos(\alpha)\ddot{\theta}\end{cases}$$

which yields the above EOMs.

(iii) What does it mean for a system to be controllable? What does it mean for a system to be observable?
Answer:
Controllability: We say that a system is controllable if it is possible to steer the state of a system from any initial value to any desired value in finite time by using an admissible control input.
Observability: We say that a system is observable if the initial value of the state can be recovered by using the state output trajectory (i.e., the output over time except at time $t=0$).

4.3 Linearization

You may notice that your EOMs that you calculated in the prelab are nonlinear equations. In this section, you will linearize these equations about a set equilibrium point.

Q1: What is the *linearized* equation of motion for generalized coordinate θ about equilibrium point $\theta_0=0$ and $\alpha_0=\pi$?
Hint: Recall that to linearize a multivariate function f of variables $z^T=[\theta\ \alpha\ \dot{\theta}\ \dot{\alpha}\ \ddot{\theta}\ \ddot{\alpha}]$ around an equilibrium point $z_0^T=[\theta_0\ \alpha_0\ \dot{\theta}_0\ \dot{\alpha}_0\ \ddot{\theta}_0\ \ddot{\alpha}_0]$, you compute

$$f_{\text{lin}}=f(z_0)+\left(\frac{\partial f(z)}{\partial\theta}\right)\Bigg|_{z_0}(\theta-\theta_0)+\left(\frac{\partial f(z)}{\partial\alpha}\right)\Bigg|_{z_0}(\alpha-\alpha_0)+\cdots+\left(\frac{\partial f(z)}{\partial\ddot{\alpha}}\right)\Bigg|_{z_0}(\ddot{\alpha}-\ddot{\alpha}_0).$$

Answer: it is

$$\left(m_pL_r^2+J_r\right)\ddot{\theta}+\frac{1}{2}m_pL_pL_r\ddot{\alpha}=\tau-B_r\dot{\theta}.$$

This is by virtue of the linearization technique explained above. Note that the right-hand side of the EOM are already linear. One computes the following, where f is equal

to the lefthand side of the EOM:

$$\begin{cases} f(z_0) = 0 \\ \frac{\partial f}{\partial \theta}\Big|_{z_0}(\theta - 0) = 0 \\ \frac{\partial f}{\partial \alpha}\Big|_{z_0}(\alpha - \pi) = 0 \\ \frac{\partial f}{\partial \dot{\theta}}\Big|_{z_0}(\dot{\theta} - 0) = 0 \\ \frac{\partial f}{\partial \dot{\alpha}}\Big|_{z_0}(\dot{\alpha} - 0) = 0 \\ \frac{\partial f}{\partial \ddot{\theta}}\Big|_{z_0}(\ddot{\theta} - 0) = \left(m_p L_r^2 - \frac{1}{4} m_p L_p^2 \cos^2(\alpha) + \frac{1}{4} m_p L_p^2 + J_r\right)\Big|_{z_0} \ddot{\theta} \\ \frac{\partial f}{\partial \ddot{\alpha}}\Big|_{z_0}(\ddot{\alpha} - 0) = \left(-\frac{1}{2} m_p L_p L_r \cos(\alpha)\right)\Big|_{z_0} \ddot{\alpha}. \end{cases}$$

Q2: What is the *linearized* equation of motion for generalized coordinate α about equilibrium point $\theta_0 = 0$ and $\alpha_0 = \pi$? **Note:** After doing this linearization, you should have a $(\alpha - \pi)$ term; change this term to α for simplicity.
Answer: it is

$$\frac{1}{2} m_p L_p L_r \ddot{\theta} + \left(J_p + \frac{1}{4} m_p L_p^2\right) \ddot{\alpha} + \frac{1}{2} m_p L_p g \alpha = -B_p \dot{\alpha}$$

This is by virtue of the linearization technique explained above. Note that the right-hand side of the EOM are already linear. One computes the following, where f is equal to the lefthand side of the EOM:

$$\begin{cases} f(z_0) = 0 \\ \frac{\partial f}{\partial \theta}\Big|_{z_0}(\theta - 0) = 0 \\ \frac{\partial f}{\partial \alpha}\Big|_{z_0}(\alpha - \pi) = \frac{1}{2} m_p L_p g(\alpha - \pi) \\ \frac{\partial f}{\partial \dot{\theta}}\Big|_{z_0}(\dot{\theta} - 0) = 0 \\ \frac{\partial f}{\partial \dot{\alpha}}\Big|_{z_0}(\dot{\alpha} - 0) = 0 \\ \frac{\partial f}{\partial \ddot{\theta}}\Big|_{z_0}(\ddot{\theta} - 0) = \frac{1}{2} m_p L_p L_r \ddot{\theta} \\ \frac{\partial f}{\partial \ddot{\alpha}}\Big|_{z_0}(\ddot{\alpha} - 0) = \left(J_p + \frac{1}{4} m_p L_p^2\right) \ddot{\alpha}. \end{cases}$$

Then, one replaces $(\alpha - \pi)$ by α, to simplify things. This can be done since the system is linear.

Q3: What is your state-space model, including symbolic matrices A and B?
Hint: Write the linearized EOMs in matrix form, i.e.,

$$\begin{bmatrix} e_{11} & e_{12} \\ e_{21} & e_{22} \end{bmatrix} \begin{bmatrix} \ddot{q}_1 \\ \ddot{q}_2 \end{bmatrix} + \begin{bmatrix} f_{11} & f_{12} \\ f_{21} & f_{22} \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \end{bmatrix} + \begin{bmatrix} g_{11} & g_{12} \\ g_{21} & g_{22} \end{bmatrix} \begin{bmatrix} q_1 \\ q_2 \end{bmatrix} = \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix},$$

and group all non double-derivative terms to the right. Now, let

$$J_T = det \begin{bmatrix} e_{11} & e_{12} \\ e_{21} & e_{22} \end{bmatrix},$$

recalling that the determinant of a matrix $M \in \mathbb{R}^{2\times 2}$ is the product of its entries on the diagonal subtracted by the product of its entries on the off-diagonal. You can now explicitly solve for $\begin{bmatrix} \ddot{\theta} \\ \ddot{\alpha} \end{bmatrix}$ and put it in a relatively compact form. You should let your state be

$$x(t) = \begin{bmatrix} \theta(t) \\ \alpha(t) \\ \dot{\theta}(t) \\ \dot{\alpha}(t) \end{bmatrix}$$

when writing our your state-space representation.
Answer: the state-space representation is

$$\begin{bmatrix} \dot{\theta}(t) \\ \dot{\alpha}(t) \\ \ddot{\theta}(t) \\ \ddot{\alpha}(t) \end{bmatrix} = \frac{1}{J_T} \begin{bmatrix} 0 & 0 & J_T & 0 \\ 0 & 0 & 0 & J_T \\ 0 & \frac{1}{4}m_p^2 L_p^2 L_r g & -\left(J_p + \frac{1}{4}m_p L_p^2\right) B_r & \frac{1}{2}m_p L_p L_r B_p \\ 0 & -\frac{1}{2}m_p L_p g \left(J_r + m_p L_r^2\right) & \frac{1}{2}m_p L_p L_r B_r & -\left(J_r + m_p L_r^2\right) B_p \end{bmatrix} \begin{bmatrix} \theta(t) \\ \alpha(t) \\ \dot{\theta}(t) \\ \dot{\alpha}(t) \end{bmatrix}$$
$$+ \frac{1}{J_T} \begin{bmatrix} 0 \\ 0 \\ J_p + \frac{1}{4}m_p L_p^2 \\ -\frac{1}{2}m_p L_p L_r \end{bmatrix} \tau$$

4.4 Validity of Linearization

In this section, you will observe in what domain your linearization of the EOMs from the Prelab is valid. To do this, you will compare the physical system's and your state-space model's response to various step inputs that will increase in magnitude, similar to the approach taken in Section 3.3.4 of Lab 1. To do this, you must first evaluate your symbolic matrices A and B, thus the following values are given:

$$\begin{cases} B_p = 0.0 \ \frac{N \cdot s}{m} \\ B_r = 0.137 \ \frac{N \cdot s}{m} \\ L_p = 0.3365 \ m \\ L_r = 0.2159 \ m \\ m_p = 0.1270 \ kg \\ m_r = 0.2570 \ kg \end{cases}$$

Recall that the moment of inertia of a thin rod about its centre of mass is $J = \frac{1}{12}mL^2$ (use this to find J_p and J_r). Again, you must transform the state-space model from your prelab in terms of voltage rather than in terms of torque (as was done in Lab 1). Follow these steps in sequence to do so:

Step 1: let matrix entry $A(3,3) = A(3,3) - \frac{K_g^2 k_t k_m}{R_m} B(3)$, where $K_g = 70$ is the total gear ratio for your setup, $k_t = 0.0077$ is the servo motor torque constant, $k_m = 0.0077$ is

the motor back-EMF const $R_m = 2.6$ is the motor armature resistance;

Step 2: let matrix entry $A(4,3) = A(4,3) - \frac{K_g^2 k_t k_m}{R_m} B(4)$;

Step 3: only now should you modify B to $B = \frac{K_g k_t}{R_m} B$.

We've prepared a MATLAB script for you to speed up the process: open **lab2_statespace_student**, input your symbolic matrices that you computed in your prelab, and run the script to evaluate your state-space matrices A and B.

Q4: What are the numerically-evaluated matrices A and B for the state-space model?
Answer: You should get

$$A = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 81.38 & -93.49 & 0.0038 \\ 0 & -122.03 & 89.97 & -0.0058 \end{bmatrix}$$

and

$$B = \begin{bmatrix} 0 \\ 0 \\ 83.6412 \\ -80.4864 \end{bmatrix}.$$

Q5: Given that the physical system's sensors are limited to reading the rotor shaft angle and the pendulum angle, what are the numerical matrices C and D for the state-space model?
Answer: You should get

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

and

$$D = \begin{bmatrix} 0 \\ 0 \end{bmatrix},$$

which can be discerned from the state-space equation $y = Cx + Du$.

Now that you have a state-space representation for the physical system, you can start constructing a Simulink model that will interact with the Quanser SRV02 servo motor, which you will use to investigate the domain of validity, controllability & observability for your linearized model. Please **save all of your MATLAB and Simulink files** as you may need to consult them in future labs.

4.4.1 Setting up the Rotary Pendulum

First, let us assemble and wire the physical system. Examine the close-up assembly shown in Figure 17, and replicate this at your workstation. Note that the high-gear configuration is used here. Use the wiring diagram shown in Figure 18 to connect the rotary pendulum to a power source and the data acquisition board.

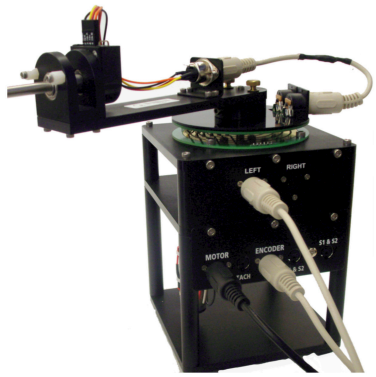


Figure 17: A close-up of the assembly of the rotary pendulum module and the Quanser SRV02 plant [1].

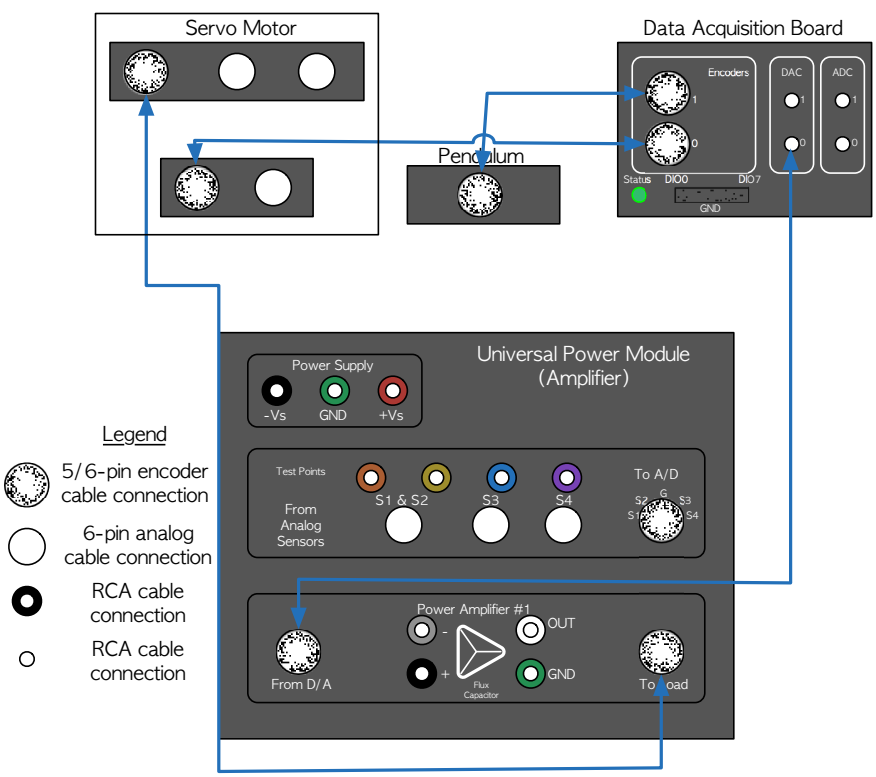


Figure 18: A wiring diagram for the Quanser SRV02 and rotary pendulum module.

Note: The power amplifier must be turned on before you can experiment with the

physical system. The power switch is located at the back of the amplifier (good luck finding it). Make sure to turn off the power amplifier before you leave the lab.

4.4.2 Simulink Model

You will be simulating various inputs and initial condition combinations with both your state-space model and the physical system. By now, you should be comfortable with modifying Simulink models. Open the file **validation_model.mdl** and modify it to incorporate your state-space model, as shown in Figure 19.

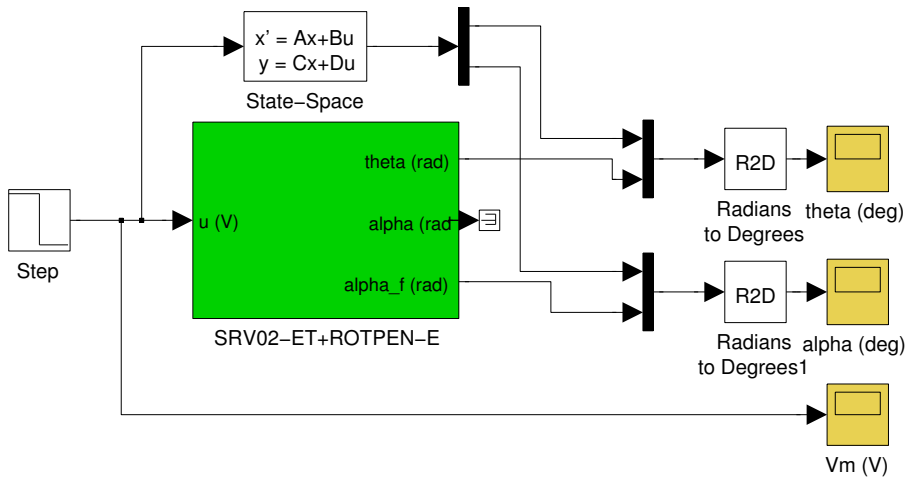
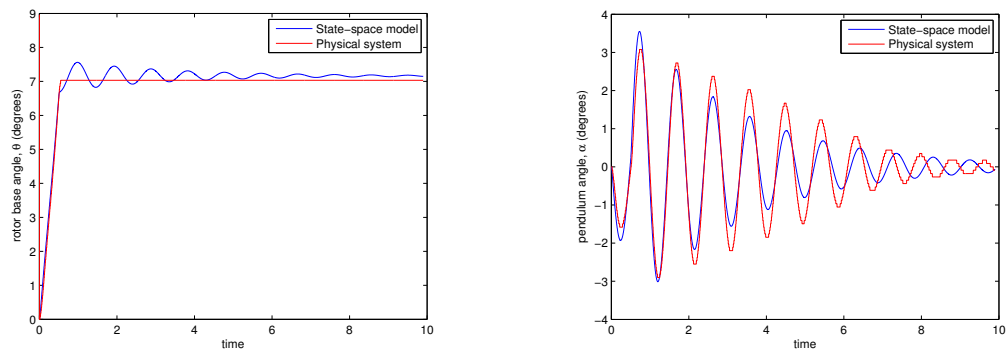


Figure 19: A Simulink model that interacts with the rotary pendulum system and incorporates the state-space model [1].

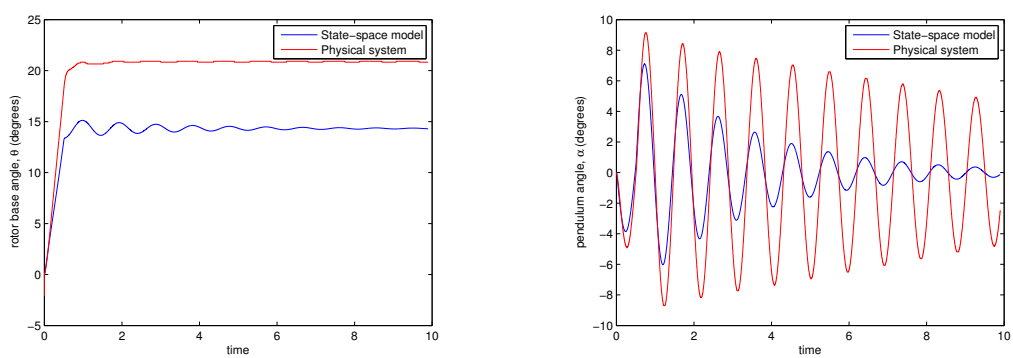
Repeat the steps outlined in Section 3.3.4 of Lab 1 to obtain time plots of the responses of the physical system and the state-space model. This time, use step inputs with initial values of 0.3, 0.6 and 0.9, and stop running the model after about 5 seconds for all trials.

Q6: Plot the responses of the physical system and the state-space model to three different step function with initial values of 0.3, 0.6 and 0.9 (leaving the Step time at 0.5 seconds). From these three time plots, what can you conclude about the domain of validity of your linearization?
Answer: As is shown in the first plot, the state-space model does a pretty good job at modelling the physical system for a step input with initial value of 0.3. The physical system's overall damping is less for the pendulum but higher for the rotor base. In the second and third plots, one can see that the linearized system is less accurate. This is because the domain of validity for the linearization is small around the linearization about $\alpha = \pi$, so perturbing the pendulum angle more from this equilibrium makes the model less accurate. Note that the rotor base state-space dynamics seems to be underdamped whereas the pendulum state-space dynamics seems to be overdamped; the underdamped and over damped effects worsen with a larger step input.



(a) (b)

Figure 20: Responses of the physical rotary pendulum system and the linearized state-space model to a step input with initial value of 0.3, step time of 0.5 and final value of 0, where (a) is the rotor base angle and (b) is the pendulum angle.



(a) (b)

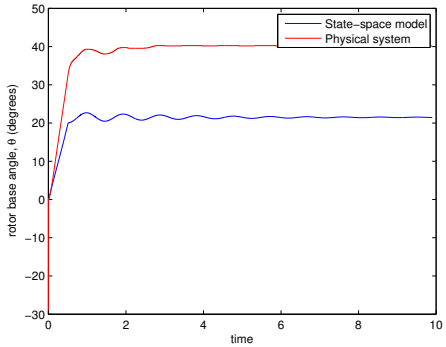
Figure 21: Responses of the physical rotary pendulum system and the linearized state-space model to a step input with initial value of 0.6

4.5 Controllability & Observability

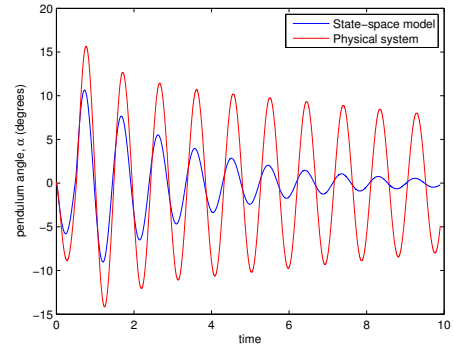
In this section, you will use your evaluated state-space matrices to find $\mathcal{C}_{(A,B)}$ and $\mathcal{O}_{(A,C)}$. Afterwards, you can determine whether the rotary pendulum system is controllable and observable.

Q7: What are the controllability and observability matrices for the linear state-space representation of the rotary pendulum? Is the rotary pendulum system controllable? Is it observable?
Answer: You should get

$$\mathcal{C}_{(A,B)} = \begin{bmatrix} 0 & 83.64 & -7820.62 & 724693.53 \\ 0 & -80.48 & 7525.80 & -693855.77 \\ 83.64 & -7820.62 & 724693.53 & -67147820.71 \\ -80.48 & 7525.80 & -693855.77 & 64287501.90 \end{bmatrix}$$



(a)



(b)

Figure 22: Responses of the physical rotary pendulum system and the linearized state-space model to a step input with initial value of 0.9.

and

$$\mathcal{O}_{(C,A)} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 81.38 & -93.49 & 0.0038 \\ 0 & -122.03 & 89.97 & -0.0058 \\ 0 & -7609.66 & 8742.27 & 81.02 \\ 0 & 7322.89 & -8412.72 & -121.68 \end{bmatrix}.$$

The system is both controllable and observable as

$$\text{rank}(\mathcal{C}_{(A,B)}) = \text{rank}(\mathcal{O}_{(A,C)}) = 4.$$

Q8: The encoder that reads the rotor base angle suddenly breaks. With this change, is the rotary pendulum system controllable? Is it observable? If the system is uncontrollable or unobservable, what (initial) states can we not (recover) access, and what does this mean in regards to our physical system?
Hint: What is in the null space of $\mathcal{C}_{(A,B)}$, $\mathcal{O}_{(A,C)}$, and what does this tell us about the system?
Answer: the controllability matrix remains the same, and thus the system is still controllable. The state-space matrix C becomes

$$C = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix},$$

which in turn changes the observability matrix to

$$\mathcal{O}_{(C,A)} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & -122.03 & 89.97 & -0.0058 \\ 0 & 0 & 0 & 0 \\ 0 & 7322.89 & -8412.72 & -121.68 \end{bmatrix},$$

which has rank 3. Thus the system is not observable. Note that the null space of $\mathcal{O}_{(C,A)}$ is

$$\text{Null}(\mathcal{O}_{(C,A)}) = \left\{ \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \right\}$$

This means that if your rotor shaft encoder is damaged, you cannot recover the initial rotor shaft angle by using output data.

Q9: The encoder that reads the pendulum angle suddenly breaks. With this change, is the rotary pendulum system controllable? Is it observable? If the system is uncontrollable or unobservable, what (initial) states can we not (recover) access, and what does this mean in regards to our physical system?
Answer: the state-space matrix C becomes

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix},$$

which in turn changes the observability matrix to

$$\mathcal{O}_{(C,A)} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 81.38 & -93.49 & 0.0038 \\ 0 & 0 & 0 & 0 \\ 0 & -7609.66 & 8742.27 & 81.02 \\ 0 & 0 & 0 & 0 \end{bmatrix},$$

which has rank 4. Thus the system is still observable. It is also still controllable, as the controllability matrix remains the same. This means that one can estimate the pendulum angle state trajectory by simply observing the system inputs and outputs and by building a state observer.

4.6 Deliverables

To recap the lab objectives, this is what you need to include in your lab report to obtain full marks:

(i) prelab question (i) and (ii);

(ii) the answers to questions Q1 through Q3 and Q6 through Q9 (include plots where requested);

Note: Save your lab report and have an electronic copy on-hand in all future labs, as you will need some of the values obtained in this lab to build upon this physical system.

V Lab 3: Stability, Feedback Control & Luenberger Observer for Balancing an Inverted Rotary Pendulum

5.1 Introduction

The purpose of this lab is to study another equilibrium of the rotary pendulum system and to develop a control technique that balances the pendulum around this equilibrium using partial state feedback and a linear observer. The prelab focuses on the linearization of the EOMs for the rotary pendulum about the equilibrium point where the pendulum is completely inverted, and also focuses on analyzing the stability of the system at this equilibrium point. In the lab, you will first use full state feedback and pole placement techniques to develop a controller that balances the inverted pendulum. Then, you will carry out the same task for the case where you do not have access to the full state information. To do this, you will develop a Luenberger observer to estimate the states that you do not have access to and you will use the estimated states to develop a controller that balances the inverted pendulum.

Figure 23: Qanser SRV02 with pendulum module, shown in an inverted orientation [1].

5.2 Prelab: Linearization, State-Space Representation & Stability Analysis

(i) What is the *linearized* equation of motion for generalized coordinate θ about equilibrium point $\theta_0 = 0$ and $\alpha_0 = 0$? You'll need to refer to your answers for the EOMs of the rotary pendulum from Section 4.2 in Lab 2.

Hint: Recall that to linearize a multivariate function f of variables $z^T = [\theta\ \alpha\ \dot{\theta}\ \dot{\alpha}\ \ddot{\theta}\ \ddot{\alpha}]$ around an equilibrium point $z_0^T = [\theta_0\ \alpha_0\ \dot{\theta}_0\ \dot{\alpha}_0\ \ddot{\theta}_0\ \ddot{\alpha}_0]$, you compute

$$f_{\text{lin}} = f(z_0) + \left(\frac{\partial f(z)}{\partial \theta}\right)\bigg|_{z_0}(\theta - \theta_0) + \left(\frac{\partial f(z)}{\partial \alpha}\right)\bigg|_{z_0}(\alpha - \alpha_0) + \cdots + \left(\frac{\partial f(z)}{\partial \ddot{\alpha}}\right)\bigg|_{z_0}(\ddot{\alpha} - \ddot{\alpha}_0).$$

Answer: it is

$$\left(m_p L_r^2 + J_r\right)\ddot{\theta} - \frac{1}{2} m_p L_p L_r \ddot{\alpha} = \tau - B_r \dot{\theta}.$$

This is by virtue of the linearization technique explained above. Note that the right-hand side of the EOM are already linear. One computes the following, where f is equal to the lefthand side of the EOM:

$$\begin{cases} f(z_0) = 0 \\ \frac{\partial f}{\partial \theta}\big|_{z_0}(\theta - 0) = 0 \\ \frac{\partial f}{\partial \alpha}\big|_{z_0}(\alpha - 0) = 0 \\ \frac{\partial f}{\partial \dot{\theta}}\big|_{z_0}(\dot{\theta} - 0) = 0 \\ \frac{\partial f}{\partial \dot{\alpha}}\big|_{z_0}(\dot{\alpha} - 0) = 0 \\ \frac{\partial f}{\partial \ddot{\theta}}\big|_{z_0}(\ddot{\theta} - 0) = \left(m_p L_r^2 - \frac{1}{4} m_p L_p^2 \cos^2(\alpha) + \frac{1}{4} m_p L_p^2 + J_r\right)\big|_{z_0}\ddot{\theta} \\ \frac{\partial f}{\partial \ddot{\alpha}}\big|_{z_0}(\ddot{\alpha} - 0) = \left(-\frac{1}{2} m_p L_p L_r \cos(\alpha)\right)\big|_{z_0}\ddot{\alpha}. \end{cases}$$

(ii) What is the *linearized* equation of motion for generalized coordinate α about equilibrium point $\theta_0 = 0$ and $\alpha_0 = 0$?
Answer: it is

$$-\frac{1}{2} m_p L_p L_r \ddot{\theta} + \left(J_p + \frac{1}{4} m_p L_p^2\right)\ddot{\alpha} - \frac{1}{2} m_p L_p g \alpha = -B_p \dot{\alpha}$$

This is by virtue of the linearization technique explained above. Note that the right-hand side of the EOM are already linear. One computes the following, where f is equal to the lefthand side of the EOM:

$$\begin{cases} f(z_0) = 0 \\ \frac{\partial f}{\partial \theta}\big|_{z_0}(\theta - 0) = 0 \\ \frac{\partial f}{\partial \alpha}\big|_{z_0}(\alpha - 0) = -\frac{1}{2} m_p L_p g \cos(\alpha)\big|_{z_0}\alpha \\ \frac{\partial f}{\partial \dot{\theta}}\big|_{z_0}(\dot{\theta} - 0) = 0 \\ \frac{\partial f}{\partial \dot{\alpha}}\big|_{z_0}(\dot{\alpha} - 0) = 0 \\ \frac{\partial f}{\partial \ddot{\theta}}\big|_{z_0}(\ddot{\theta} - 0) = -\frac{1}{2} m_p L_p L_r \cos(\alpha)\big|_{z_0}\ddot{\theta} \\ \frac{\partial f}{\partial \ddot{\alpha}}\big|_{z_0}(\ddot{\alpha} - 0) = \left(J_p + \frac{1}{4} m_p L_p^2\right)\big|_{z_0}\ddot{\alpha}. \end{cases}$$

(iii) What is your state-space model, including symbolic matrices A and B?
Hint: Write the linearized EOMs in matrix form, i.e.,

$$\begin{bmatrix} e_{11} & e_{12} \\ e_{21} & e_{22} \end{bmatrix} \begin{bmatrix} \ddot{q}_1 \\ \ddot{q}_2 \end{bmatrix} + \begin{bmatrix} f_{11} & f_{12} \\ f_{21} & f_{22} \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \end{bmatrix} + \begin{bmatrix} g_1 \\ g_2 \end{bmatrix} = \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix},$$

and group all non double-derivative terms to the right. Now, let

$$J_T = det \begin{bmatrix} e_{11} & e_{12} \\ e_{21} & e_{22} \end{bmatrix},$$

recalling that the determinant of a matrix $M \in \mathbb{R}^{2\times 2}$ is the product of its entries on the diagonal subtracted by the product of its entries on the off-diagonal. You can now explicitly solve for $\begin{bmatrix} \ddot{\theta} \\ \ddot{\alpha} \end{bmatrix}$ and put it in a relatively compact form. You should let your state be

$$\mathbf{x}(t) = \begin{bmatrix} \theta(t) \\ \alpha(t) \\ \dot{\theta}(t) \\ \dot{\alpha}(t) \end{bmatrix}$$

when writing our your state-space representation. It may be easier to consult & modify your answer from Section 4.2.
Answer: the state-space representation is

$$\begin{bmatrix} \dot{\theta}(t) \\ \dot{\alpha}(t) \\ \ddot{\theta}(t) \\ \ddot{\alpha}(t) \end{bmatrix} = \frac{1}{J_T} \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & \frac{1}{4}m_p^2 L_p^2 L_r g & -\left(J_p + \frac{1}{4}m_p L_p^2\right) B_r & -\frac{1}{2}m_p L_p L_r B_p \\ 0 & \frac{1}{2}m_p L_p g \left(J_r + m_p L_r^2\right) & -\frac{1}{2}m_p L_p L_r B_r & -\left(J_r + m_p L_r^2\right) B_p \end{bmatrix} \begin{bmatrix} \theta(t) \\ \alpha(t) \\ \dot{\theta}(t) \\ \dot{\alpha}(t) \end{bmatrix}$$
$$+ \frac{1}{J_T} \begin{bmatrix} 0 \\ 0 \\ J_p + \frac{1}{4}m_p L_p^2 \\ \frac{1}{2}m_p L_p L_r \end{bmatrix} \tau$$

(iv) Now, evaluate your symbolic state-space matrices A and B using the following values:

$$\begin{cases} J_p = 0.001199\ kg \cdot m^2 \\ J_r = 0.000998\ kg \cdot m^2 \\ B_p = 0.0024\ \frac{N \cdot s}{m} \\ B_r = 0.0024\ \frac{N \cdot s}{m} \\ L_p = 0.3365\ m \\ L_r = 0.2159\ m \\ m_p = 0.1270\ kg \\ m_r = 0.2570\ m. \end{cases}$$

Deduce what your state-space matrices C and D should be given the rotary pendulum sensory equipment. What are the poles of the open-loop system? What can you conclude about the stability of this open-loop system? Does this make sense from a physical point of view?
Hint: When we talk about systems without any controls intervention, we are referring to *internal* stability. For a review of internal stability, see your MTHE 332 course notes [3, p. 162].
Notes:

(a) the viscous damping coefficients for the pendulum and rotor base have changed from Lab 2: this is because the orientation of the pendulum has changed;

(b) you still have to transform your state-space matrices to account for a voltage input rather than a torque input. It may be easier to you your MATLAB script from Lab 2.

Answer: The state-space matrices should be

$$A = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 81.38 & -46.05 & -0.93 \\ 0 & 122.03 & -44.31 & -1.39 \end{bmatrix},$$

$$B = \begin{bmatrix} 0 \\ 0 \\ 83.64 \\ 80.48 \end{bmatrix},$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

and

$$D = \begin{bmatrix} 0 \\ 0 \end{bmatrix}.$$

The open loop poles can be found by computing the roots of the system's characteristic equation, i.e., finding the roots of $\det(sI - A) = 0$. In MATLAB, this is easily done by using the command $eig(A)$. The open-loop poles are

$$\{-48.63,\ 7.05,\ -5.86,\ 0\}.$$

It can be concluded that the system is internally unstable about the chosen equilibrium as $\text{spec}(A) \cap \mathbb{C}^+ \neq \emptyset$. This makes physical sense as the inverted pendulum is not in a stable orientation and any perturbation about this equilibrium point will drive the pendulum away from the inverted orientation.

(v) Recall that in Lab 2 you linearized the rotary pendulum EOMs about another equilibrium. Was the linearized system stable at this equilibrium (and if so, what type of stability is ensured)? Does this make sense from a physical point of view?

Answer: Yes, the linearized system about the pendulum resting downwards is internally stable, as $\text{spec}(A) \cap \mathbb{C}^+ = \emptyset$:

$$\{-45.24,\ -1.11 + 6.57i,\ -1.11 - 6.57i,\ 0\},$$

and since for each eigenvalue $\lambda \in \text{spec}(A)$, its geometric multiplicity equals its algebraic multiplicity (since $A \in \mathcal{M}^{4\times 4}$ and there are four distinct eigenvalues of A, thus there is one eigenvector for each corresponding eigenvalue). This makes physical sense because when the pendulum is perturbed from this equilibrium point, it returns to it.

(vi) If you were tasked with *designing* a stabilizing control for the inverted pendulum for a specific application (e.g., for the balancing of a monopedal robot whose body is balanced via servo motor), what methods would you use to do so other than the ones presented in what follows (i.e., other than pole placement techniques)?
Answer: I'll leave this to the imagination of the students/TA.

5.3 Pole Placement & Full State Feedback

Preliminary question:

Q1: Can we hope to use full state feedback to steer the linearized rotary pendulum system, linearized about $\alpha = 0$ (the unstable equilibrium)?
Answer: To verify this, one must compute the controllability matrix of the system:

$$\mathcal{C}_{(A,B)} = \begin{bmatrix} 0 & 83.64 & -3926.74 & 190939.11 \\ 0 & 80.48 & -3818.93 & 189168.17 \\ 83.64 & -3926.74 & 190939.11 & -9279982.88 \\ 80.48 & -3818.93 & 189168.17 & -9191635.19 \end{bmatrix}$$

The system's controllability matrix has full rank, and hence the system is controllable. Thus, we can hope to use full state feedback and pole placement techniques to control the system.

Control technique:

In this section, you will be designing a feedback controller that stabilizes the system about the inverted pendulum orientation. That is, to balance the pendulum in the inverted orientation, you must design a feedback gain that renders the closed-loop system *internally asymptotically stable*. You saw in Lab 2 that when linearizing the EOMs of the rotary pendulum about the stable equilibrium (i.e., the pendulum oriented downwards), there is a domain of validity for your linearization. The same applies for the linearization that you performed in this prelab, so when designing the balance controller, you will need to lift up the pendulum from the downwards orientation to the inverted orientation while holding the servo base, and then have the controller engage in a small domain centered at the unstable equilibrium. Thus, the controller will be of the form

$$u(t) = \begin{cases} K(\mathbf{x_d} - \mathbf{x}(t)) & |\alpha| < \epsilon; \\ 0 & \text{otherwise}, \end{cases}$$

where ϵ is the angle designating when the controller will engage and x_d is the desired reference state. To balance the pendulum in the inverted orientation, one should choose a desired reference state of

$$\mathbf{x_d} = \begin{bmatrix} \theta_d \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

where θ_d is a desired rotor base angle (one may wish that the rotor base track a desired trajectory, thus $\theta_d = \theta_d(t)$). To design an appropriate control gain that will balance the inverted pendulum, consider our closed-loop system when the controller is engaged:

$$\begin{cases} \dot{\mathbf{x}}(t) = (A - KB)\,\mathbf{x}(t) + B\mathbf{x_d}; \\ \mathbf{y}(t) = C\mathbf{x}(t). \end{cases} \tag{4}$$

Let us review some theory from class:

Recall that for $A \in \mathcal{M}^{n\times n}(\mathbb{R})$ and for $B \in \mathbb{R}^n$, if the pair (A, B) is controllable, then there exists a transformation $T \in \mathcal{M}^{n\times n}(\mathbb{R})$ where $\det(T) \neq 0$ and

$$\tilde{A} = TAT^{-1} = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & & 0 \\ \vdots & \vdots & & \ddots & \vdots \\ 0 & 0 & 0 & 0 & 1 \\ -a_0 & -a_1 & -a_2 & \dots & -a_{n-1} \end{bmatrix}$$

where $\chi_A = \chi_{\tilde{A}} = s^n + a_{n-1}s^{n-1} + \cdots + a_1 + a_0$ (i.e., $\tilde{A}$ is uniquely determined by the coefficients of the characteristic polynomial of A), and

$$\tilde{B} = TB = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix},$$

which is called the canonical controllable form (recall that (A, B) is controllable if and only if $(\tilde{A}, \tilde{B})$ is controllable since system controllability is invariant under similarity transformations). To find an appropriate feedback gain K for your system (4) that will place the system's poles in $\mathbb{C}^-$, you will first design a feedback gain $\tilde{K}$ for the transformed system and then transform $\tilde{K}$ into K. To do that, design $\tilde{K} = [k_0,\ k_1,\ \dots,\ k_{n-1}]$ so that you obtain the desired poles:

$$\tilde{A} + \tilde{B}\tilde{K} = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & & 0 \\ \vdots & \vdots & & \ddots & \vdots \\ 0 & 0 & 0 & 0 & 1 \\ (k_0 - a_0) & (k_1 - a_1) & (k_2 - a_2) & \dots & (k_{n-1} - a_{n-1}) \end{bmatrix}$$

which yields $\chi_{(\tilde{A}+\tilde{B}\tilde{K})} = s^n + (a_{n-1} - k_{n-1})s^{n-1} + \cdots + (a_1 - k_1) + (a_0 - k_0)$. In class, you found an explicit formula for computing the transformation T:

$$T = \mathcal{C}_{(\tilde{A},\tilde{B})}\mathcal{C}^{-1}_{(A,B)}.$$

Back to your lab:

You can use these results to design the appropriate feedback gain K to balance the inverted pendulum.

Q2: Find the coefficients of the characteristic polynomial for your open-loop state-space model by multiplying the eigenvalues together, as shown in the following:

$$\chi_A = (s - \lambda_1)(s - \lambda_2)(s - \lambda_3)(s - \lambda_4) = s^4 + a_3 s^3 + a_2 s^2 + a_1 s^1 + a_0.$$

What is $\tilde{A}$?
Answer: You should get

$$\tilde{A} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 2013.52 & 98.98 & -47.44 \end{bmatrix}.$$

Q3: Find a $\tilde{K}$ such that the poles of the closed-loop system are $\{-2.8+2.86i,\ -2.8-2.86i,\ -30,\ -40\}$.
Hint: You can use the *sym2poly(p)* MATLAB command to find the coefficients of the characteristic polynomial generated by the desired poles.
Answer: You should get

$$\tilde{K} = [19223.52 \quad 9854.89 \quad 1707.00 \quad 28.15].$$

You get this by the following MATLAB commands:

$$a = poly(A);$$
$$syms\ \ s;$$
$$b = sym2poly((s - (-2.8 + 2.86i)) * (s - (-2.8 - 2.86i)) * (s - (-30)) * (s - (-40)));$$
$$K_tilde = [b(5) - a(5) \quad b(4) - a(4) \quad b(3) - a(3) \quad b(2) - a(2)];$$

Q4: Calculate the transformation $T = \mathcal{C}_{(\tilde{A},\tilde{B})}\mathcal{C}^{-1}_{(A,B)}$. A quick way to find $\mathcal{C}_{(A,B)}$ is to use the *ctrb(A,B)* MATLAB command. What is the ensuing feedback gain, K?
Hint: $A - BK = T^{-1}(TAT^{-1} + TBKT^{-1})T \implies A - BK = T^{-1}(\tilde{A} + \tilde{B}\tilde{K})T$, thus $\tilde{K} = KT^{-1}$.
Answer: You should get

$$K = [-5.25 \quad 28.10 \quad -2.75 \quad 3.21].$$

You will now use this feedback gain to experimentally balance the physical rotary pendulum system. Open **balancing_model.mdl** and make sure that your state-space matrices A, B, C and D as well as your feedback gain K is loaded in the MATLAB workspace. The Simulink model is shown in Figure 24. You may need to connect the physical system to the computer, as was done in Section 4.4.1 of Lab 2. The feedback controller is activated by the Control Switch block when the pendulum angle is within an ϵ-range of $\alpha = 0$ (recall that $\alpha = 0$ corresponds to the pendulum being completely inverted). While the system is balancing the inverted pendulum, you will input to the system a square wave of amplitude 10, so that the servo motor (and hence θ) tracks the square wave. This is done by changing the Amplitude block to 10, but only **after** the pendulum has started balancing. Note that you will have to hold the servo motor when initially lifting the pendulum to the inverted orientation, and then let it go once the pendulum is completely inverted. It may take a few tries to get the pendulum to balance.

The curious reader may concerned with the workings of the Find State X subsystem block found in the Simulink model shown in Figure 24. This subsystem block uses a numerical differentiator technique to find the derivative estimations of the measured signals, θ and α. This technique is based on using a high-gain observer [4] to estimate the derivatives of the signals. While the angular velocity of the rotor base and the pendulum are not measured and outputted from the physical system (neither are they outputted from the state-space model), one can usually estimate these variables by using such methods.

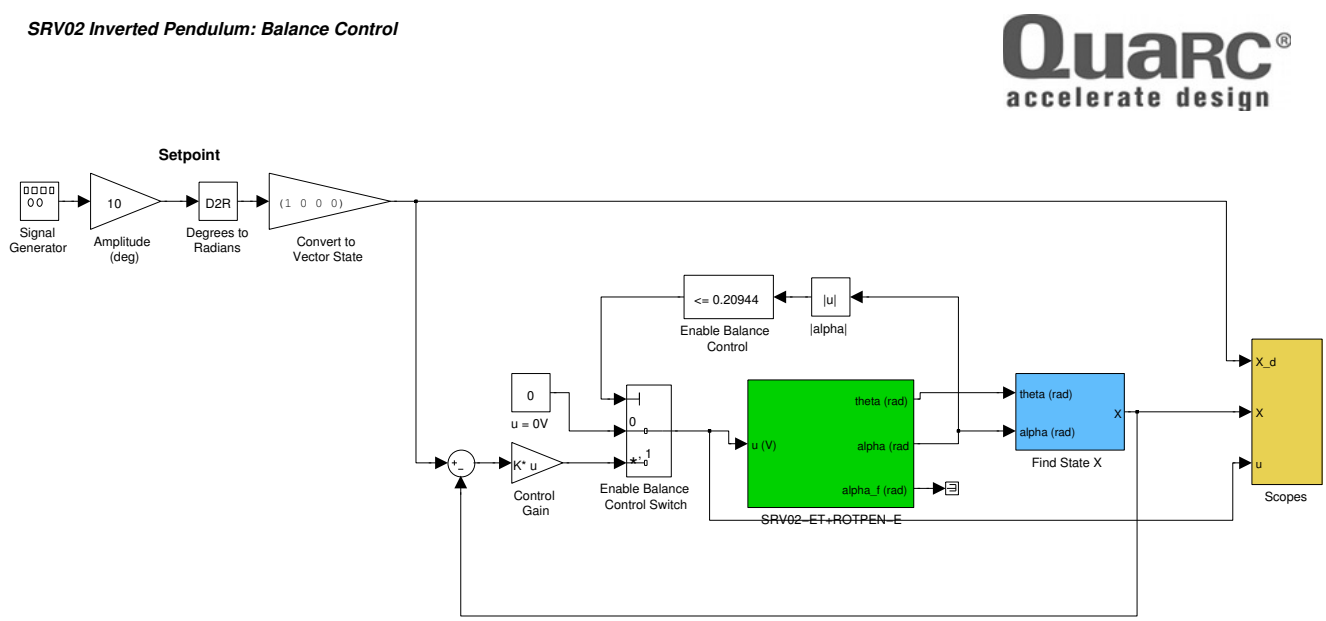


Figure 24: A Simulink model which interacts with the physical rotary pendulum system and uses state feedback in a small domain about the unstable equilibrium (i.e., inverted pendulum orientation) to balance the inverted pendulum. [1]

Q5: Record the results of the balancing experiment and plot your data. On one plot, compare the square wave input for the rotor base and the measured rotor base angle, θ. On another plot, include the pendulum angle data, α. Does your feedback controller manage to balance the inverted pendulum while tracking the square wave trajectory? What happens if you keep increasing the square wave amplitude, and why does this happen? What happens when you change the poles of the closed-loop system to be more negative in the left half-plane of $\mathbb{C}$ (i.e., designing the feedback gain, K, such that the closed-loop poles are much more negative)? For balancing the inverted pendulum, discuss the *trade-offs between designing* the control system such that the poles of the closed-loop system are close to the origin or far in the left half-plane.

Answer: Yes, it should be able to balance the inverted pendulum while tracking the square wave with amplitude of 10. If the amplitude is increased, then the pendulum angle may exceed the domain of validity of the linearization, and the feedback controller may not be able to balance the inverted pendulum. Instructors should note that students' answers will vary, as the tension in the slip-gear will change the input required to balance the pendulum. When changing the poles of the closed-loop system by changing F, you change the rate of stabilization and the overshoot/undershoot of the stabilization. If one were to place these poles far in the left half-plane, then stabilization would be very immediate, but this immediate response comes at the cost of a large overshoot. In the case of the inverted pendulum, where the linearized system is only valid in a small neighbourhood about the inverted orientation, a large overshoot may be detrimental. Conversely, a very slow stabilization may allow the pendulum to fall outside the linearization's domain of validity, which also causes negative effects. Hence one must design and tune the feedback gain, K, for this application such that balancing stabilization is achieved quickly enough without a large overshoot. Here are the plots:

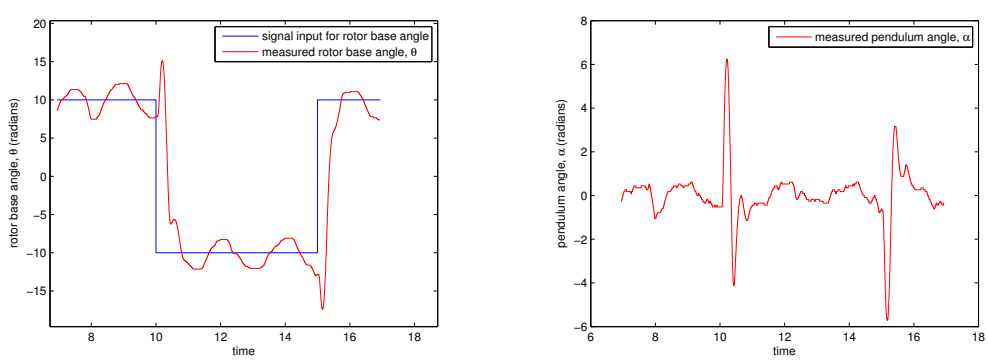


Figure 25: (a) A plot of the desired and measured rotor base angle trajectories for the closed-loop feedback controller system while balancing the pendulum in the inverted orientation and while the rotor base tracks a square wave trajectory with amplitude 10, and; (b) a plot of the measured pendulum angle while the physical system tracks this square wave trajectory.

5.4 Linear Observer

Recall our linear time-invariant system dynamics

$$\dot{\mathbf{x}}(t) = A\mathbf{x}(t) + Bu(t) \qquad \mathbf{y}(t) = C\mathbf{x}(t)$$

and consider the case where the encoder that reads the pendulum angle is damaged, i.e.,

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

In this case, you do not have access to one of the system's internal states, α. However, you may be able to determine the unknown state by using the system's input and output data. To do this, you will construct a linear dynamical system that produces an estimate of the physical system's state, also known as a *linear state observer*.

Let us review some theory from class:

Let us denote the state estimate by $\hat{\mathbf{x}}(t)$. David G. Luenberger proposed the following linear state observer [5]:

$$\dot{\hat{\mathbf{x}}}(t) = A\hat{\mathbf{x}}(t) + Bu(t) + L(\mathbf{y}(t) - C\hat{\mathbf{x}}(t)), \tag{5}$$

where L is the observer gain. Note that the input and measured output data are both used in (5). The observer in (5) is composed of two parts: the first part, $A\hat{\mathbf{x}}(t)+Bu(t)$, is the rate of change of $\hat{\mathbf{x}}(t)$ as computed by the physical system's state-space model; the second part, $L(\mathbf{y}(t) - C\hat{\mathbf{x}}(t))$, is the difference between the observed output and the estimated output, scaled by the observer gain. Notice that without the second term, (5) becomes familiar:

$$\dot{\hat{\mathbf{x}}}(t) = A\hat{\mathbf{x}}(t) + Bu(t). \tag{6}$$

Let us denote the estimation error by $\tilde{\mathbf{x}}(t)$ and define it by $\tilde{\mathbf{x}}(t) = \mathbf{x}(t) - \hat{\mathbf{x}}(t)$. If one were to study the estimation error of the simplified observer (6), then they would conclude that $\tilde{\mathbf{x}}(t)$ would go to zero when $\text{spec}(A) \subset \mathbb{C}^-$, as the error dynamic for (6) is

$$\begin{aligned} \dot{\tilde{\mathbf{x}}}(t) &= A\left(\mathbf{x}(t) - \hat{\mathbf{x}}(t)\right) + Bu(t) - Bu(t) \\ &= A\tilde{\mathbf{x}}(t). \end{aligned}$$

That is, when matrix A has all of its eigenvalues in the left half-plane, then the state estimate would converge to the actual state. But this convergence depends on the physical system's dynamics (thus the convergence rate can be quite slow, depending on the system) and the spectral constraints on A are very restrictive. In contrast, the error dynamic for the Luenberger observer in (5) is

$$\dot{\tilde{\mathbf{x}}}(t) = (A - LC)\tilde{\mathbf{x}}(t). \tag{7}$$

Thus one can design an observer gain L such that $\text{spec}(A-LC) \subset \mathbb{C}^-$ (i.e., an *asymptotically stable observer*) and one can tune L such that $\hat{\mathbf{x}}(t)$ converges to $\mathbf{x}(t)$ fast enough for the desired application. An important question to ask is, "when does such an observer exist"? In class, you proved that one can build an observer with these properties when (C, A) is *detectable*.

Back to your lab:

Throughout this section, we'll be studying the rotary pendulum linearized about its stable equilibrium (i.e., the system from Lab 2). You will need your MATLAB files from Lab 2.

Q6: From your previous assessment on the controllablility & observability of the rotary pendulum system when its pendulum encoder is damaged (Lab 2), can you hope to observe the unknown state α by designing and building an appropriate state observer? What does system detectability mean?
Answer: Yes; since the system is observable when

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix},$$

it is also detectable, as detectability is a weaker notion of observability. System detectability implies that all of the system's unobservable states are asymptotically stable, but there are no unobservable states for this system.

One way to *design an observer is by eigenvalue assignment*: the Luenberger observer error, $\tilde{\mathbf{x}}(t)$, governed by the differential equation (7), has a characteristic polynomial chosen by the designer. That is, the observer designer chooses L such that the eigenvalues of $(A - LC)$ are at desired locations $p_1, p_2, \ldots, p_n$. This design method has some benefits over other methods, mainly:

(i) theoretically, the observer error can decrease faster if the eigenvalues of $(A - LC)$ are more negative (however, there is a trade-off which you will explore), and;

(ii) if the observed states are being used for state feedback, then the least negative eigenvalue of $(A - LC)$ should be more negative than the eigenvalues of the state feedback system $A - BK$. Loosely speaking, this dictates that the observer dynamics will converge to the unobserved states *faster* than the system dynamics will change these states.

One may notice a similarity between finding a feedback gain K such that $A - BK$ has eigenvalues in $\mathbb{C}^-$ and finding an observer gain L such that $A - LC$ also has eigenvalues in $\mathbb{C}^-$. Recall that the eigenvalues of a matrix and its transpose are the same. The problem of observer design is in fact the *dual* of the state feedback problem, with the following equivalences [6]:

$$\begin{aligned} (A - LC)^T &= A^T - C^T L^T \\ &\leftrightarrow A - BK \end{aligned}$$

$$A \leftrightarrow A^T \quad B \leftrightarrow C^T \quad K \leftrightarrow L^T.$$

In Section 5.3, you solved for K by using the canonical controllable form and a similarity transformation. In MATLAB, there's a shortcut to solving for K which uses the *place*

command. To solve for the feedback gain with desired poles at $\{-2.8 + 2.86i,\ -2.8 - 2.86i,\ -30,\ -40\}$, you can do the following in MATLAB:

$$p = [-2.8 + 2.86i \quad -2.8 - 2.86i \quad -30 \quad -40];$$

$$K = place(A, B, p);$$

Q7: In MATLAB, how could you use the *place* command to design observer gain L such that $A - LC$ has its poles at $p = \{p_1,\ p_2,\ p_3,\ p_4\}$? **Hint:** duality.
Answer: Using the duality of the observer and feedback design problems, and the equivalences stated above, you could use the MATLAB command

$$L = place(A', C', p)';$$

to complete the task.

5.4.1 Designing & Validating an Observer

You will now design an observer for the linearized rotary pendulum system, linearized about the *stable equilibrium*. For this section of the lab, you'll need the state-space matrices A, B, C and D that you calculated in Section 4.4 of Lab 2. You should alter your C matrix so that only the rotor base angle, θ, is outputted, as you are studying the case where the encoder reading the pendulum angle is damaged. Figure 26 shows a Simulink block diagram for a Luenberger observer; you will integrate this observer subsystem into a Simulink model to compare the actual and estimated states.

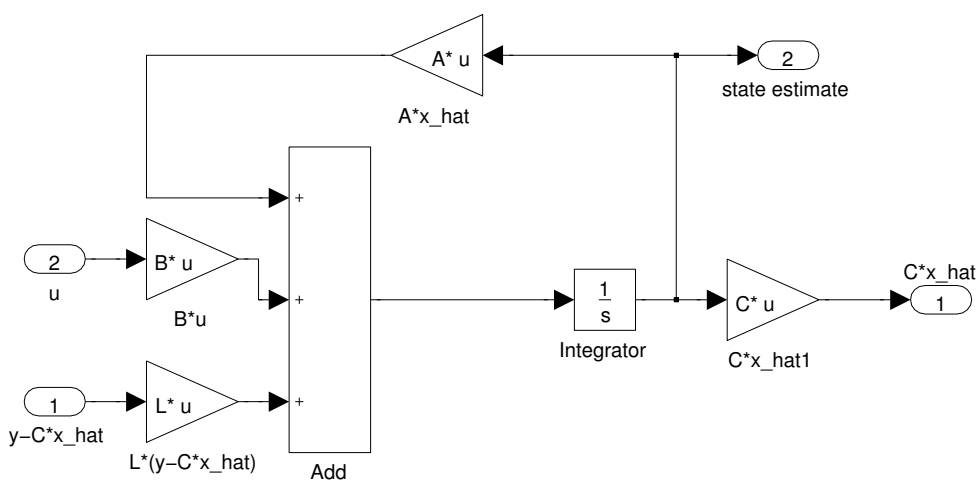


Figure 26: A Simulink block diagram of a Luenberger observer. This is a subsystem block that can be incorporated into a Simulink model, along with a state-space subsystem block, to compare the estimated and the actual states.

Open the Simulink model **luenberger_openloop_model.mdl**. Note that the regular State-Space block is not used to model the linearized rotary pendulum system; in place, we've created a subsystem block that does this, and it is shown in Figure 27 along with the Luenberger observer subsystem block.

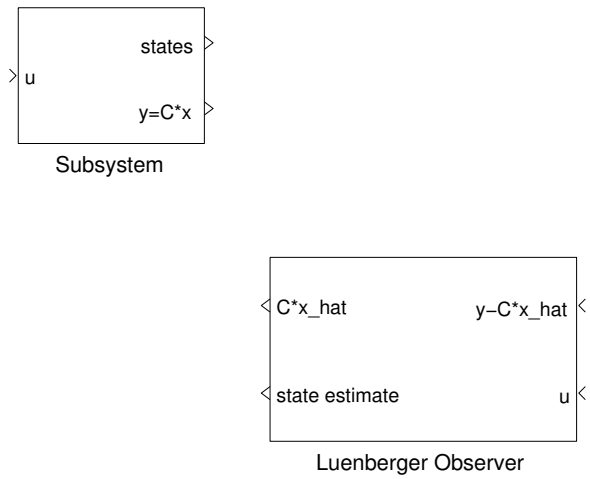


Figure 27: A Simulink block diagram of a state-space subsystem & Luenberger observer subsystem. The state-space subsystem block allows one to output the usual $y = Cx$ output as well as outputting all of the states at each time, making it useful for state estimate validations.

Connect these two Simulink subsystem blocks so that the observer can estimate the states of the linearized state-space model of the rotary pendulum. Now, add a Signal Generator block, a Sum or Add block, and a DegreesToRadians block to the Simulink model and use a sine signal with amplitude 10 as your input (transforming your input to radians using the DegreesToRadians block). In the state-space subsystem, set the initial conditions (found in the integrator block) of your states to zero (in MATLAB, this is done with the vector $[0; 0; 0; 0]$), and set the initial conditions of your observer subsystem to non-zero values between $[-10, 10]$. Make sure to output your states and your state estimates to your workspace using To Workspace blocks.

Q8: Run various tests using this model while changing the observer's poles in $\mathbb{C}^-$. First, *design and tune your observer* so that the state estimates converges to the actual states very quickly; second, try to achieve state estimates that don't deviate from the actual states too much. *What metrics you would use to evaluate* your observer (and the resulting observer gain design) were it to be used for: state feedback to balance the inverted pendulum, and; monitoring an industrial process where large spikes cause the industrial plant to shut down.
Answer: Tuning the observer gain such that the state estimates converge very quickly to the actual states (i.e., the observer poles are far in the left half-plane) comes at the cost of a very large overshoot. Thus, it is important to design the observer gain so that this tradeoff be exploited for a given application. For the inverted pendulum application, since the domain of validity for the linearized system is a small neighbourhood about

the inverted orientation, a large overshoot in the pendulum angle state estimation would make the ensuing feedback response overcompensate unnecessarily. Conversely, a slow state estimate convergence can make the ensuing feedback response much to slow to keep the inverted pendulum in the linearized system's domain of validity. This is why this physical stabilization problem is not considered - it is much too finicky to work, and may require techniques other than gain tuning to make it work properly. Some metrics may include overshoot, settling time, energy used in state feedback, etc. For the monitoring application, one would choose to design the observer poles close to the origin since large state estimate overshoots would cost the industrial plant much lost time. The main metric here is overshoot, among other engineering metrics.

5.4.2 Testing your Observer on the Physical System

In this section, you will verify that the observer you built for the state-space model works for the physical system. This will help you identify the benefits and drawbacks of different observer tunings for gain L. Open the Simulink model **luenberger_openloop_quanser.mdl**. When accessing the rotary pendulum subsystem block, one will notice that the Luenberger observer is nested into this subsystem, as shown in Figure 28. This subsystem outputs the measured and the estimated pendulum angle, so that these angles can be compared directly.

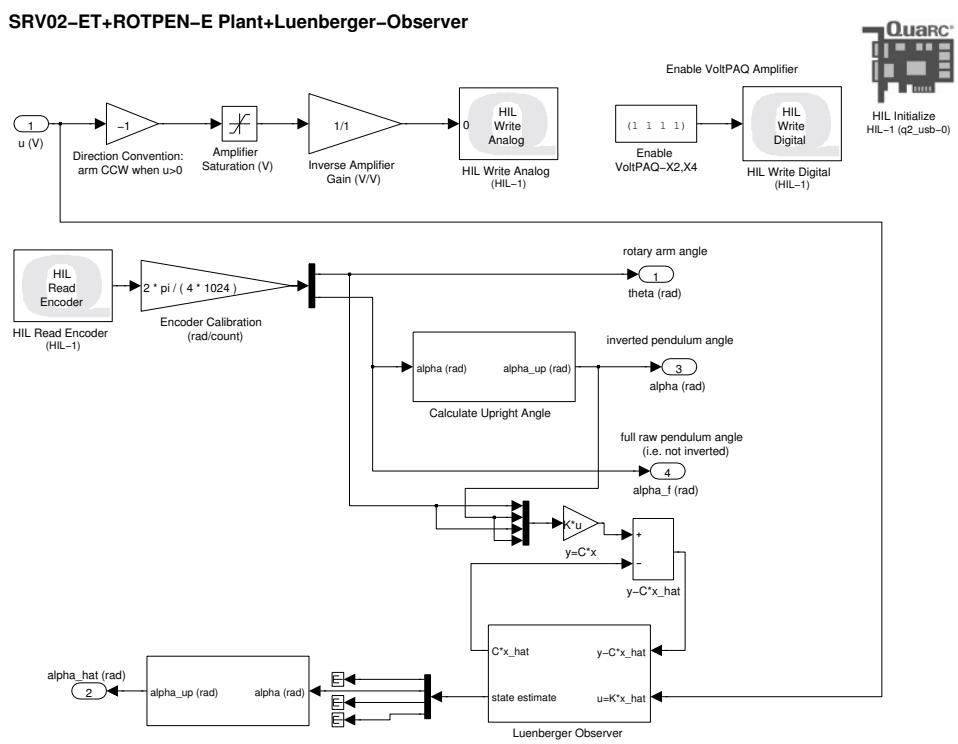


Figure 28: A Simulink model of a Luenberger observer estimating & outputting the pendulum angle, $\hat{\alpha}$, for the physical rotary pendulum system. Note that in the Luenberger observer, measured states $\dot{\theta}$ and $\dot{\alpha}$ are non needed, as they are scaled by C.

Q9: Build & run **luenberger_openloop_quanser.mdl**, with a sine wave input with amplitude of 1 and frequency of 5 rad/s. First, use $[0,\ 0,\ 0,\ 0]$ as the initial conditions for the observer. Plot the measured and the estimated pendulum angle, α and $\hat{\alpha}$, with respect to time when your observer has poles $\{-5,\ -6,\ -7,\ -8\}$. Repeat the experiment for observer poles $\{-15,\ -16,\ -17,\ -18\}$, and then for observer poles $\{-45,\ -46,\ -47,\ -48\}$ (creating the same plots for each experiment). Repeat these for the case where the initial conditions of the observer are not identically zero. From these plots, what can you conclude about the relationship between the accuracy of the state estimates produced by your Luenberger observer and *the design and tuning* of the observer gain, L? Give two different real-life applications where state observers could be used, and what *metrics* would you develop to evaluate these observers in the setting of their respective applications (specifically look at the relationship between observer behaviour and pole placement via observer gain, L).

Answer: one will note that the observer error after a crest of the sine wave input cannot be completely eliminated for these three observer gains, as some observer error is present in all three plots. The penalty for having very negative observer poles is apparent: while the observer dynamic is very fast, it quickly acts on a control input and then reacts to reduce the estimate output error term $(L(y - C\hat{x}(t))$, and does so frequently. For observer poles at $\{-15,\ -16,\ -17,\ -18\}$, the observer gives a most accurate estimate, but the observer is still a bit too sensitive to control inputs. Due to the constant change in the input signal, the observer estimate error never has time to converge to zero. I will leave the engineering-based responses to be answered by the current TA.

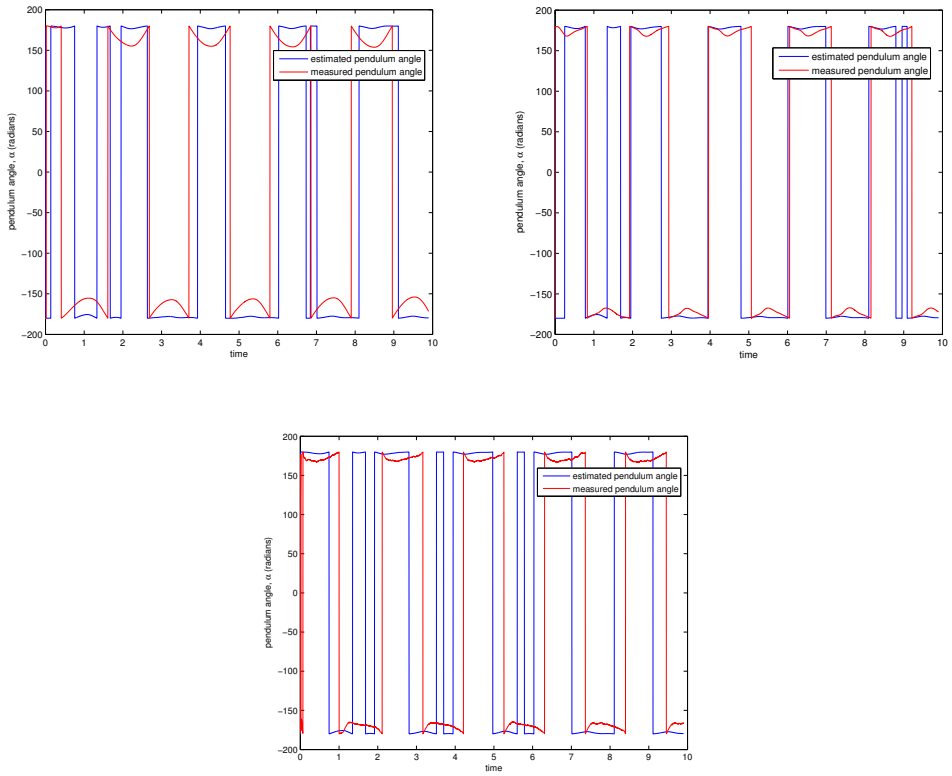


Figure 29: Plots comparing the measured and estimated pendulum angle response to a sine wave input of amplitude of 0.5 and frequency of 3 rad/s for observer gains a) $L = \{-5,\ -6,\ -7,\ -8\}$, b) $L = \{-15,\ -16,\ -17,\ -18\}$ and c) $L = \{-45,\ -46,\ -47,\ -48\}$.

5.5 Deliverables

To recap the lab objectives, this is what you need to include in your lab report to obtain full marks:

(i) prelab question (i) through (vi), and;

(ii) the answers to questions Q1, Q4, Q5, Q8 and Q9 (include plots where requested).

Note: Save your lab report and have an electronic copy on-hand in all future labs, as you will need some of the values obtained in this lab to build upon this physical system.

VI Lab 4: LQR Optimization for Feedback Control and Observer Design for the Rotary Flexible Beam

6.1 Introduction

The purpose of this lab is to design and optimize a feedback controller for the rotary flexible beam system. One way of achieving this is to use a Linear Quadratic Regulator, which automates the process of finding a state feedback gain K in a way that optimizes the system for certain design parameters. The prelab focuses on analyzing the stability of the open-loop rotary flexible beam system. In the lab, you will first design and optimize the closed-loop state feedback system by using the LQR technique to minimize the flexible beam's deflections while the rotor base tracks an input signal trajectory. The closed-loop system will have to adhere to set restrictions on settling time, percent overshoot and maximal flexible beam deflection. Next, you will observe the effects of using only partial state feedback on the system response to an input. Finally, you will attempt to minimize these system response effects by building an observer for the unobservable state and using the ensuing state estimate as feedback.

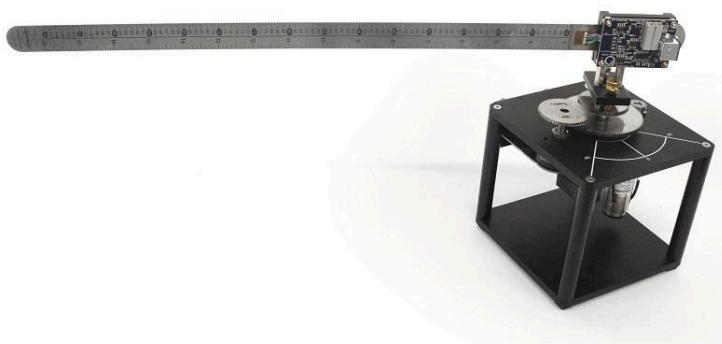


Figure 30: Qanser SRV02 with flexible beam module [1].

6.2 Prelab: Open-Loop Stability Analysis

Recall the 2-DOF state-space matrices that you found in Section 3.3.4 of Lab 1.

(i) What are the poles of the open-loop state-space representation of the rotary flexible beam system? What can you conclude about the stability of this open-loop system? Does this make sense from a physical point of view?
Hint: When we talk about systems without any controls intervention, we are referring to *internal* stability. For a quick review of internal stability, see your MTHE 332 course notes [3, p. 162].
Answer: The open-loop poles are $\{0,\ -8.13 + 22.47i,\ -8.13 - 22.47i,\ -24.21\}$, and both the algebraic and geometric multiplicities are equal, thus the system is internally stable.

(ii) Can we hope to use full state feedback to steer the rotary flexible beam system?
Answer: To verify this, one must compute the controllability matrix of the system:

$$\mathcal{C}_{(A,B)} = \begin{bmatrix} 0 & 61.77 & -2501.26 & 62750.59 \\ 0 & -61.77 & 2501.26 & -41639.42 \\ 61.77 & -2501.26 & 62750.59 & -980692.86 \\ -61.77 & 2501.26 & -41639.42 & 125861.00 \end{bmatrix}$$

The system's controllability matrix has full rank, and hence the system is controllable. Thus, we can hope to use full state feedback and pole placement techniques to control the system.

(iii) If the system is steerable, what *metrics* would you develop to evaluate your state feedback controller against others for different *engineering applications* (e.g., when the rotary pendulum is used to sort boxes on a conveyor belt in the warehouse of an online retailer; or when the rotary pendulum is mounted underwater for tidal wave energy production, and control input keeps it from becoming unstable)?
Answer: I'll leave this to the imagination of the students/TA

6.3 Designing Feedback using the Linear Quadratic Regulator

A fundamental problem in control theory is operating a control system at minimum cost. When concerned with linear differential equations (e.g., $\dot{\mathbf{x}}(t) = A\mathbf{x}(t) + Bu(t)$) and quadratic cost functions, then one is really concerned with the *LQ problem*. In this lab, we will use a main result from this area of literature [**?**] which is the solution to the linear quadratic regulator (LQR). The solution to the LQR will help you design a feedback controller which minimizes the deflections of the flexible beam while the rotor base tracks a square wave trajectory (within certain *design constraints* on settling time and overshoot).

Let us review some theory from class:

The LQ problem that we are dealing with is an infinite-horizon problem that studies the control system

$$\begin{cases} \dot{\mathbf{x}}(t) = A\mathbf{x}(t) + Bu(t), & \text{where } \mathbf{x}(0) = \mathbf{x}_0 \\ \mathbf{y}(t) = C\mathbf{x}(t) \end{cases}$$

subject to minimizing the cost function

$$\eta(u) = \int_0^\infty \mathbf{y}^T(t)\mathbf{y}(t) + u^T(t)u(t)\ dt = \int_0^\infty \mathbf{x}^T(t)C^TC\mathbf{x}(t) + u^T(t)u(t)\ dt.$$

You may think of the first part of the cost function, $\int_0^\infty \mathbf{y}^T(t)\mathbf{y}(t)dt$, as the *energy of the controlled output*, and the second part, $\int_0^\infty u^T(t)u(t)dt$, as the *energy of the control input*.

Two **main results** from class follow this reformulation: the first is that if (A, B, C) is a minimal realization, then there exists a positive definite symmetric solution $\Pi_\infty \in \mathcal{M}^{n\times n}(\mathbb{R})$ to the continuous algebraic Ricatti equation (CARE):

$$A^T\Pi_\infty + \Pi_\infty A - \Pi_\infty BB^T\Pi_\infty = -C^TC;$$

the next main result is that with (A, B, C) a minimal realization, the controller

$$u(t) = -B^T \Pi_\infty \mathbf{x}(t)$$

solves the LQ problem.

Q1: What does it mean to be a minimal realization? Is the state-space model of the rotary flexible beam system a minimal realization? Can one hope to use the LQR approach to optimize the feedback control inputs for the rotary flexible beam system?

Answer: Given any transfer function, any state-space model that is both controllable and observable and has the same input-output behaviour as the transfer function is said to be a minimal realization. As was shown in Lab 1, the state-space model of the rotary flexible beam accurately describes the physical system; since the transfer function describes the input and output behaviour of the physical system, one can conclude that the state-space model has the same input-output behaviour as the transfer function. Furthermore, the observability matrix for the state-space model is

$$\mathcal{O}_{(C,A)} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 623.77 & -40.49 & 0 \\ 0 & -965.53 & 40.49 & 0 \\ 0 & -25257.84 & 1639.61 & 623.77 \\ 0 & 25257.84 & -1639.61 & -965.53 \end{bmatrix}$$

which has full rank (see prelab for computation of $\mathcal{C}_{(A,B)}$, which also has full rank). Thus, (A, B, C) is a minimal realization, and one can use the main results for LQR theory to optimize the feedback control inputs for the physical system.

Back to your lab:

There are a few differences in the theory that you've learned from class and the treatment we will apply here. First, we wish to minimize some of the system's outputs more than others, e.g., our focus is on minimizing the flexible beam deflections. With this focus, we present a more generalized form of the quadratic cost function:

$$\eta(u) = \int_0^\infty \mathbf{x}^T(t) Q \mathbf{x}(t) + u^T(t) u(t) \, dt, \tag{8}$$

where Q is symmetric and $\mathbf{x}^T(t) Q \mathbf{x}(t) > 0$ for all t (positive-definite). You will set the matrix Q as a diagonal weighting matrix to assign different weights on the states within the cost function (8). In turn, this will determine how the control input $u(t)$ will minimize the cost function $\eta(u)$, and thus the weightings of matrix Q will have direct effects on the

ensuing feedback gain, K. Hence, the LQR design process is really an iterative process: tune the diagonal weightings of Q, observe the effects, compare against the desired performance metrics, and repeat.

In the theory from class, one had to verify that (A, B, C) was a minimal realization in order for $u(t) = -B^T \Pi_\infty \mathbf{x}(t)$ to solve the LQ problem. Using the generalized quadratic cost function in (8), one must only verify that (A, B) be stabilizable to ensure that this input solves the LQ problem. Can the LQR approach be used to optimize the feedback control inputs for the rotary flexible beam?
Answer: stabilizability follows from the fact that the system is controllable (see previous answer for $\mathcal{C}_{(A,B)}$ computation).

Open the Simulink model **lqr_model.mdl**, shown in Figure 31. You may have to connect the physical system to the computer, as was done in Section 3.3.1 of Lab 1. Notice the presence of a high-gain observer, used to estimate the time derivatives of the signals θ and α, as was discussed in Section 5.3 of Lab 3. In this lab, you will use these signal derivative estimates as feedback. Notice also that the output signal is subtracted from the input signal before being scaled by the feedback gain. Thus when the system's outputs match the system's inputs, a signal of all zeros will be input into the physical system; when the system inputs and outputs do not match, the feedback gain should be designed to drive the non-zero states to zero (this will work if K is designed to make the closed-loop system *asymptotically stable*).

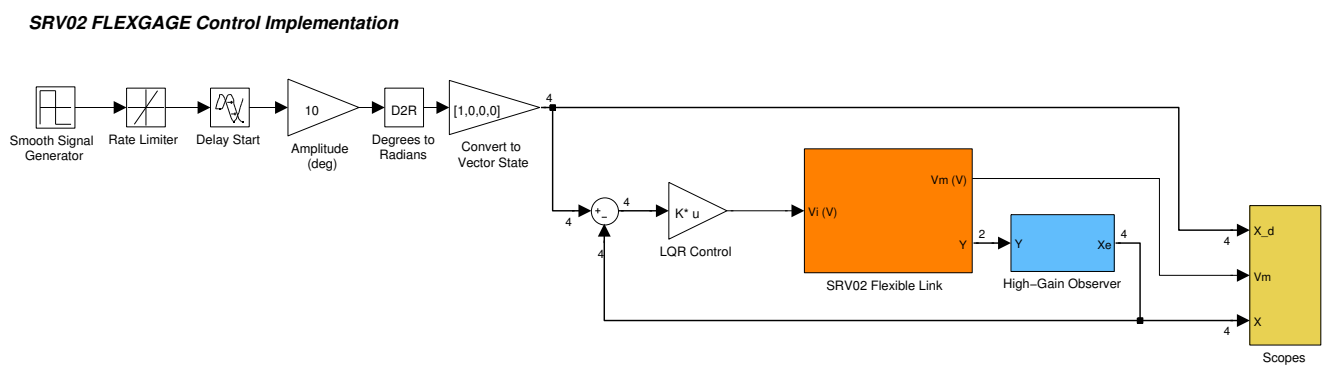


Figure 31: A Simulink model that inputs a signal into the rotary flexible beam system and observes its output. This model can be used to drive the rotor base to track a signal input while minimizing the flexible beam deflections via full state feedback. The feedback gain K is designed using a linear quadratic regulator.

Your first task will be to observe the effects of changing the diagonal weightings of Q. Using your state-space matrices from Section 6.2, and setting

$$Q = \begin{bmatrix} q_1 & 0 & 0 & 0 \\ 0 & q_2 & 0 & 0 \\ 0 & 0 & q_3 & 0 \\ 0 & 0 & 0 & q_4 \end{bmatrix}$$

you can use the MATLAB command *care* to solve the continuous algebraic Ricatti equation

(CARE). This will return the solutions to the Riccati equation, Π_∞, which you will use to compute your feedback gain, $K = -B^T\Pi_\infty$.

Q2: Using a diagonal weightings matrix for Q, express the cost function symbolically. What would be the effect of increasing the diagonal elements q_i on the feedback gain, K?
Answer:

$$\eta(u) = \int_0^\infty \left(\begin{bmatrix} \theta & \alpha & \dot{\theta} & \dot{\alpha} \end{bmatrix} \begin{bmatrix} q_1 & 0 & 0 & 0 \\ 0 & q_2 & 0 & 0 \\ 0 & 0 & q_3 & 0 \\ 0 & 0 & 0 & q_4 \end{bmatrix} \begin{bmatrix} \theta \\ \alpha \\ \dot{\theta} \\ \dot{\alpha} \end{bmatrix} + u^T(t)u(t) \right) dt$$

$$= \int_0^\infty q_1\theta^2 + q_2\alpha^2 + q_3\dot{\theta}^2 + q_4\dot{\alpha}^2 + u^2(t)\ dt$$

Increasing the weightings q_i would alter the cost function so as to give the corresponding state more influence over the cost, e.g., increasing q_1 would cause the control input $u(t)$ to have a bias towards minimizing the state scaled by q_1 (in our lab, this state is θ), and thus would increase the corresponding feedback gain element k_1 to compensate, where

$$K = \begin{bmatrix} k_1 \\ k_2 \\ k_3 \\ k_4 \end{bmatrix}.$$

This behaviour is the same for all weightings q_i. Note that due to the interconnectedness of the system states, changing one q_i may (and usually will) effect multiple states.

Q3: *Check your feedback gain with your TA*. Since the Simulink model *subtracts* the feedback from the input, make sure you define K as $K = B^T\Pi_\infty$ and **not** $-K = B^T\Pi_\infty$. Vary the weightings $q_1,\ q_2,\ q_3,\ q_4$ independently from starting values $[q_1,\ q_2,\ q_3,\ q_4] = [1,\ 1,\ 1,\ 1]$. Run **lqr_model.mdl** with a square wave input of amplitude 30 **total** (there is a signal amplitude & an amplitude block, multiply these to get 30) and with frequency of 0.33 Hz. What effects does each modification have on the feedback gain, $K = [k_1,\ k_2,\ k_3,\ k_4]$? Plot each of the state responses with respect to time. What effects does each modification have on the physical system's state responses? What weighting(s) has(have) noticeable effects on reducing the rise time (one in particular is noticeable)? Did any decrease the flexible beam's deflections? *Formulate problems and their specifications* that you anticipate when tuning particular weightings in various *engineering applications* (e.g., for tidal power generators - or flexible beams mounted on the ocean floor in locations of notably strong tidal power - where the flexible beam's deflections trigger a power generator, but can also be strained to failure).
Answer: Here are the plots and the corresponding feedback gains produced by the linear quadratic regulator:

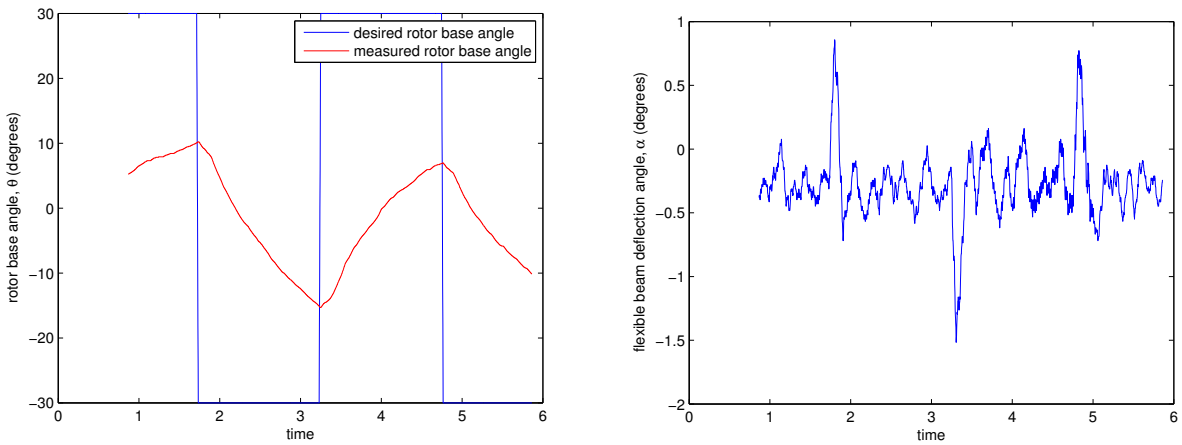


Figure 32: a) The rotor base response to a square wave input when $[q_1,\ q_2,\ q_3,\ q_4] = [1,\ 1,\ 1,\ 1]$, giving $[k_1,\ k_2,\ k_3,\ k_4] = [1,\ -9.5873,\ 0.6004,\ -0.4092]$, and b) the corresponding flexible beam deflection angle with respect to time.

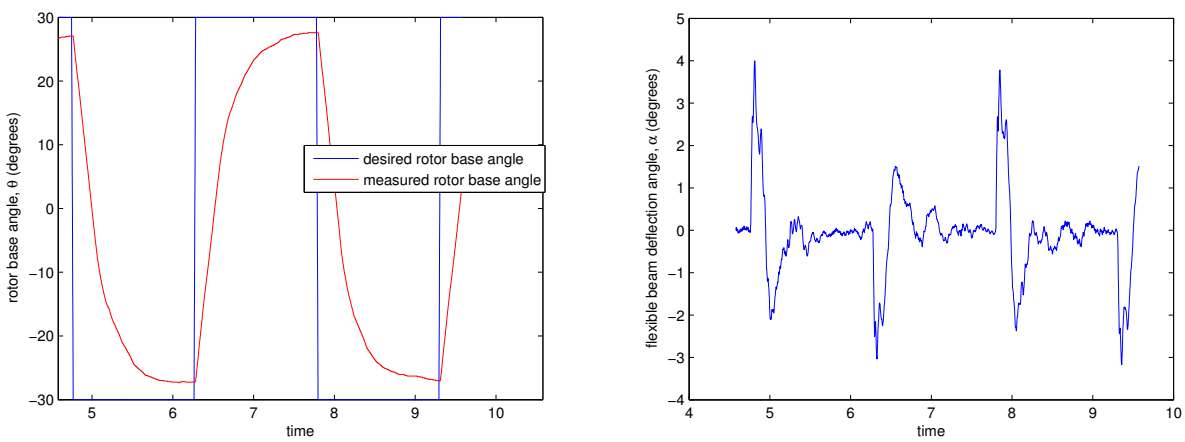


Figure 33: a) The rotor base response to a square wave input when $[q_1,\ q_2,\ q_3,\ q_4] = [20,\ 1,\ 1,\ 1]$, giving $[k_1,\ k_2,\ k_3,\ k_4] = [4.4721,\ -10.464,\ 0.798,\ -0.253]$, and b) the corresponding flexible beam deflection angle with respect to time.

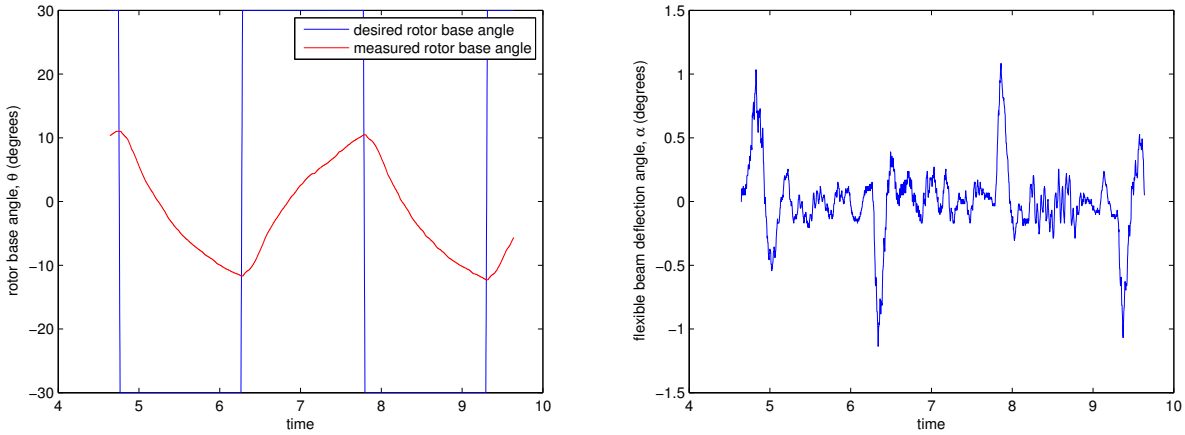


Figure 34: a) The rotor base response to a square wave input when $[q_1,\ q_2,\ q_3,\ q_4] = [1,\ 20,\ 1,\ 1]$, giving $[k_1,\ k_2,\ k_3,\ k_4] = [1,\ -10.408,\ 0.601,\ -0.413]$, and b) the corresponding flexible beam deflection angle with respect to time.

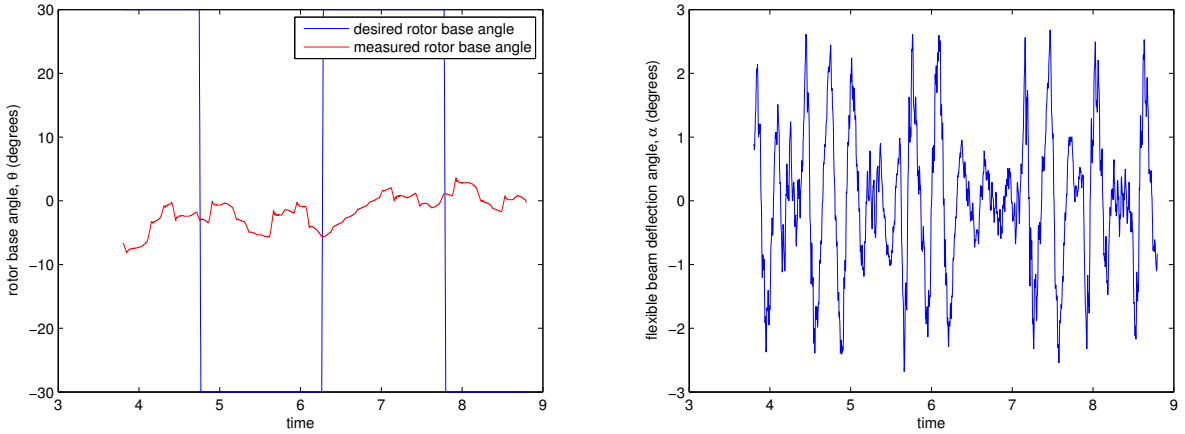


Figure 35: a) The rotor base response to a square wave input when $[q_1,\ q_2,\ q_3,\ q_4] = [1,\ 1,\ 20,\ 1]$, giving $[k_1,\ k_2,\ k_3,\ k_4] = [1,\ -10.85,\ 3.881,\ -0.133]$, and b) the corresponding flexible beam deflection angle with respect to time.

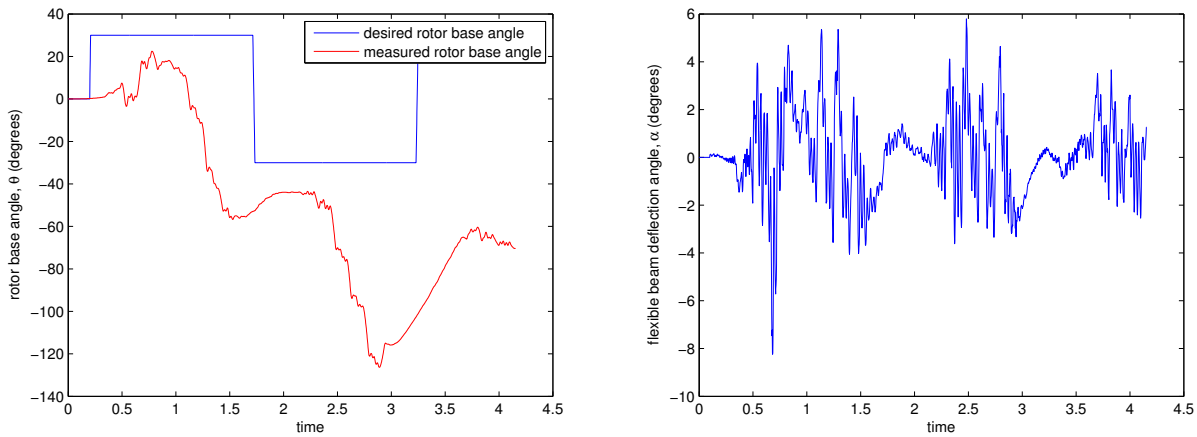


Figure 36: a) The rotor base response to a square wave input when $[q_1,\ q_2,\ q_3,\ q_4] = [1,\ 1,\ 1,\ 20]$, giving $[k_1,\ k_2,\ k_3,\ k_4] = [1,\ -43.48,\ 0.677,\ -3.449]$, and b) the corresponding flexible beam deflection angle with respect to time.

One will notice that increasing q_1 has an impact on changing k_1, and that the rotor base response is much faster (rise time and settling time is decreased). Also, increasing q_4 has a large impact on changing k_4, and thus from a theoretical standpoint, the angular velocity of the flexible beam should be reduced by increasing q_4. The other weightings, q_2 and q_3, have little effect on the system.

Q4: You will perform a similar experiment with the same input (square wave with amplitude of 30 and frequency of $0.33Hz$), except now there are *design restrictions* on the rotor base's response: we require that maximum percent overshoot to be 10%, the rise time be within 0.5 seconds and the steady-state error be within 5% for use in our application. In addition, the maximum flexible beam deflection should be no more than $10°$. Tune the diagonal weightings of Q until you've achieved the desired results, and then plot the rotor's base angle, θ, and the flexible beam's deflection angle, α, with respect to time. Also plot the control input with respect to time. What weightings in the matrix Q did you use to achieve these results, and what was the resulting feedback gain, K?

Answer: The weightings that were used to achieve the desired performance was $[q_1,\ q_2,\ q_3,\ q_4] = [130,\ 1,\ 1,\ 3.5]$, giving $[k_1,\ k_2,\ k_3,\ k_4] = [11.401,\ -25.031,\ 1.379,\ -0.436]$. Instructors should note that there can be large variances between flexible beam modules, so these answers should be marked correct not by whether students can replicate the same weightings but by whether they can achieve that achieve the required response characteristics. Here are the plots:

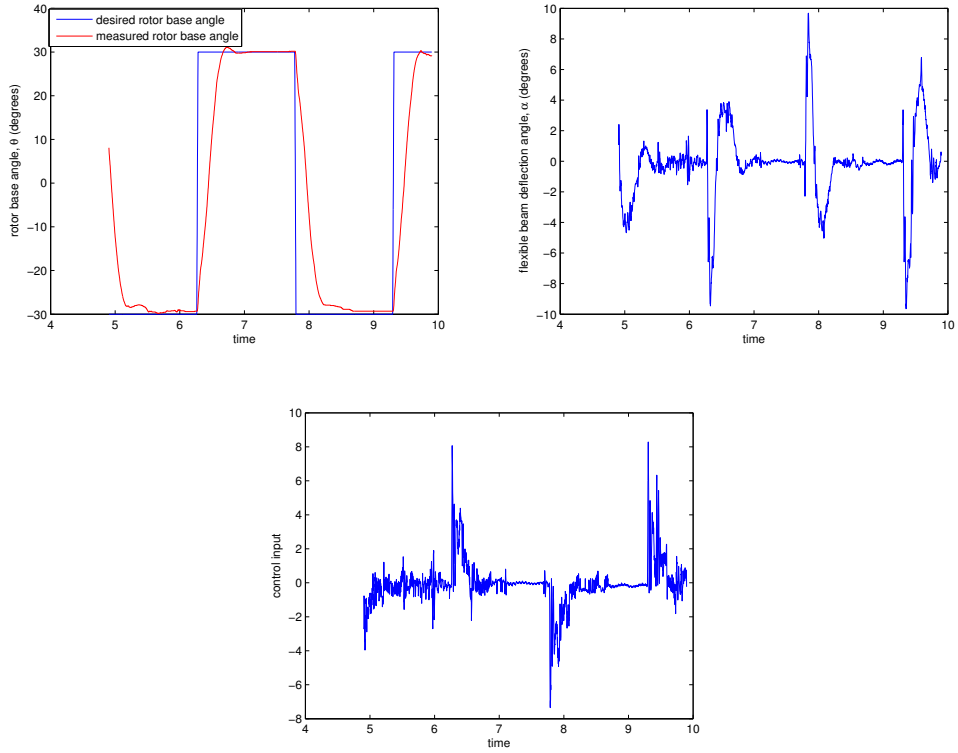


Figure 37: a) The rotor base angle, and (b) the flexible beam deflection angle responses to a square wave input, with matrix Q tuned to achieve the desired performance metrics.

6.4 Effects of Partial State Feedback on LQR

Now, you will observe the effects of using only partial state feedback in your closed-loop system on the optimality of the control system. Open the Simulink model **lqr_partial_feedback_model.mdl**. Notice that the only state feedback being used is θ and the high-gain observer estimate of $\dot{\theta}$, as shown in Figure 38.

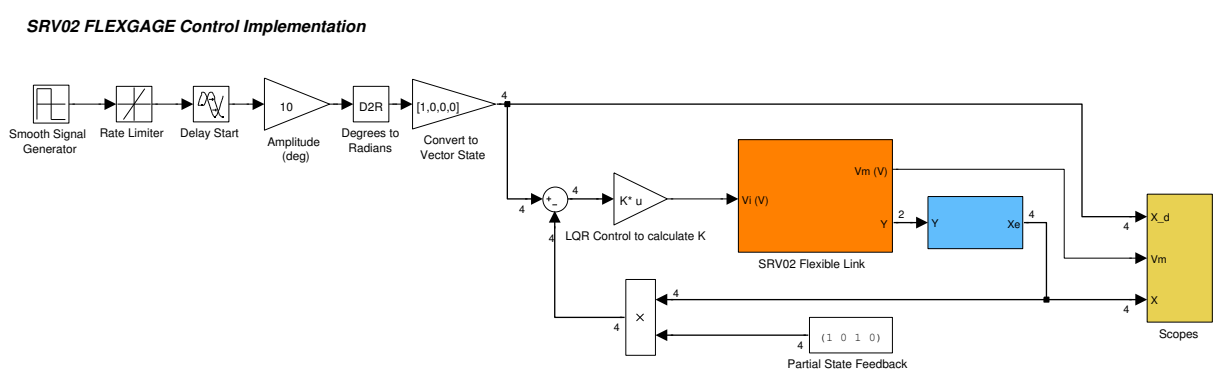


Figure 38: A Simulink model of the rotary flexible beam that utilizes partial state feedback and the linear quadratic regulator to design a feedback gain, K, that minimizes the deflections of the flexible beam while the rotor base tracks a square wave trajectory.

Q5: Similar to the last experiment, use the LQR technique and the same weighting matrix Q when computing your feedback gain, K. Plot the system responses to a square wave function with amplitude 30 and with frequency of 0.33 Hz. Compare the optimality of full state feedback and the partial state feedback closed-loop systems. Describe at least two previously-learned techniques that you may employ (and describe how) to obtain a better state feedback design, leading to superior system behaviour.

Answer: in the partial state feedback case, the closed-loop control system is less optimal than the full state feedback system: the beam deflects much more and the steady-state error is higher. Due to the design of the cost function, these increases in beam deflection and steady-state error result in a higher cost. The control input is slightly less, but due to the design of the cost function, this will lower the cost significantly.

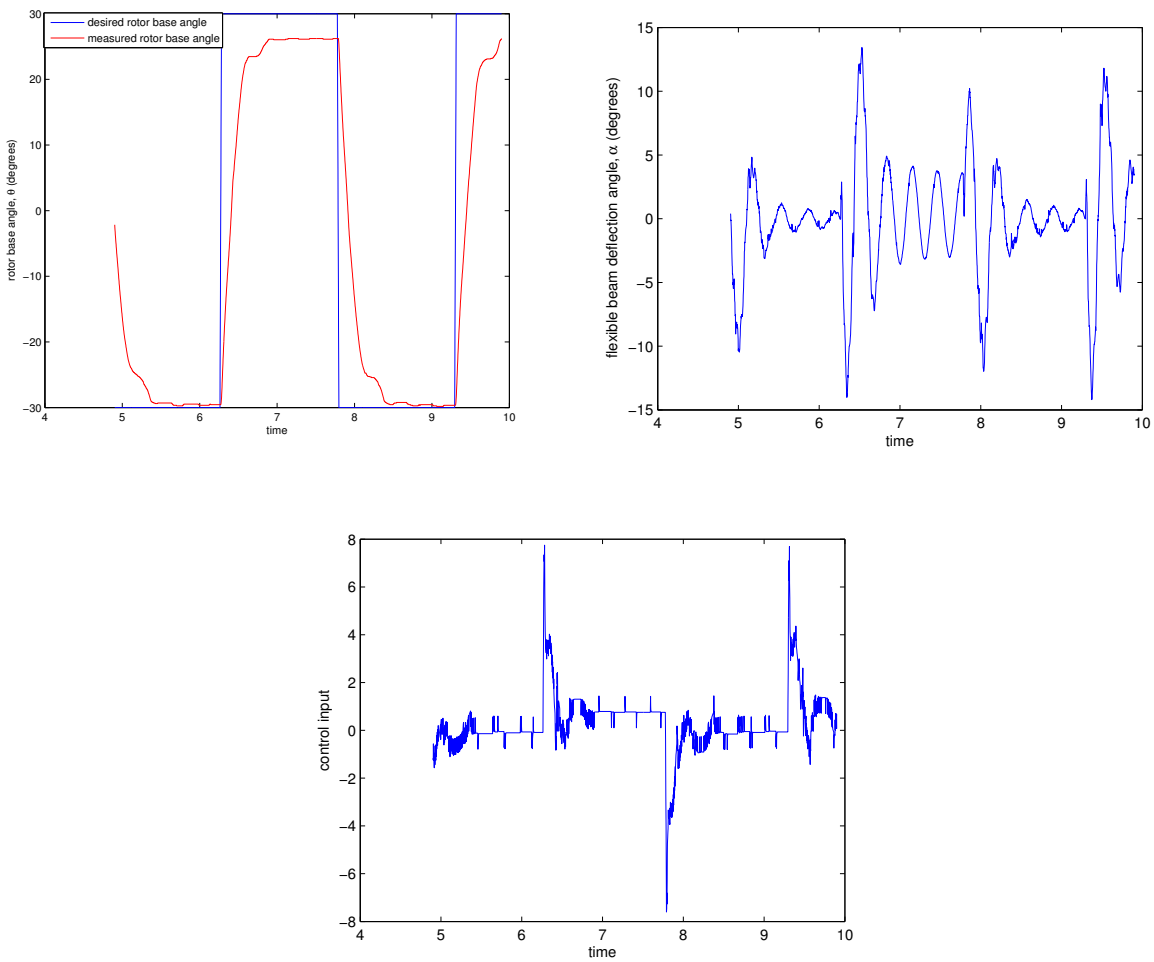


Figure 39: a) The rotor base angle, and (b) the flexible beam deflection angle responses to a square wave input when using only partial state feedback. This partial state feedback control system is less optimal than the full state feedback control system.

6.5 Designing an Observer & Using Observer Estimate as Feedback for LQ Problem

Q6: Is it possible to construct an observer for the rotary flexible link system, for the scenario where the strain gauge, which reads the flexible beam's deflection angle, is damaged?
Answer: with this modification, the state-space matrix C becomes

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

and the observability matrix becomes

$$\mathcal{O}_{(A,C)} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 623.77 & -40.49 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & -25257.84 & 1639.61 & 623.77 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

which has full rank. The controllability matrix does not change. Thus the modified system is controllable and observable, which implies that it is detectable and stabilizable. Hence, one can hope to construct an observer for the state α, and then use the high-gain observer subsystem block to estimate $\dot{\alpha}$.

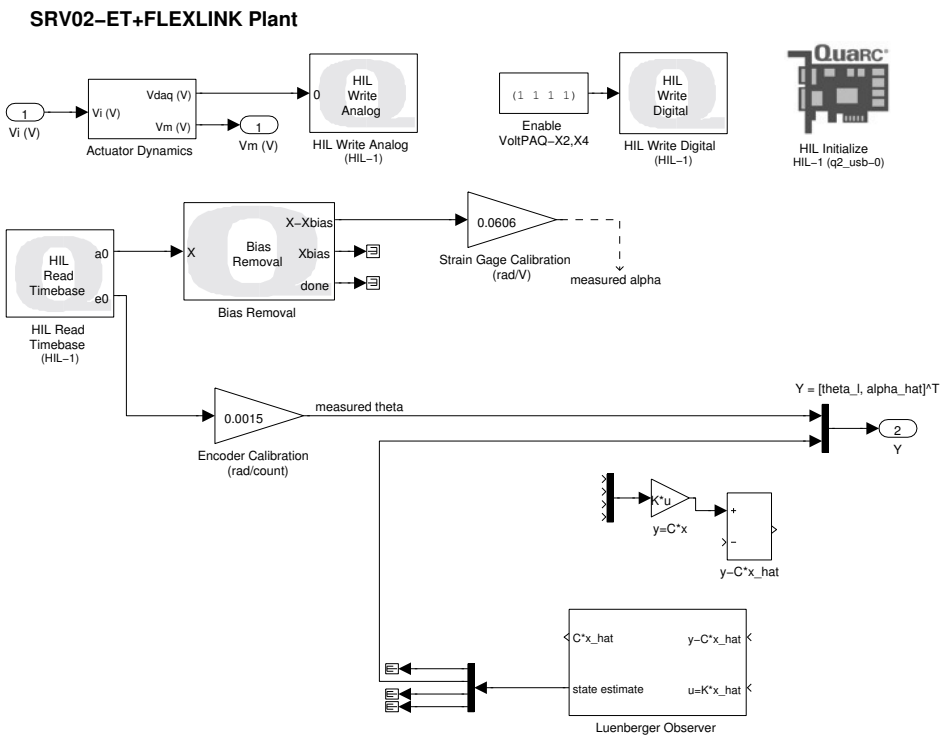


Figure 40: A Simulink subsystem block showing the rotary pendulum system and a Luenberger observer. These systems are not connected, however one can connect these in such a way to compare the measured and estimated pendulum angle, α and $\hat{\alpha}$, as well as use the pendulum angle estimate as LQR feedback. Note that this model will not run without modification.

Q7: Open the Simulink model **lqr_observer_model** with the Luenberger observer contained in the rotary pendulum's subsystem block, as shown in Figure 40. Follow the procedure from Section 5.4 of Lab 3 to build a Luenberger state observer to estimate state α. You will need to tune your observer gain, L, similarly to how you did so in Lab 3 so that the observer dynamics are faster than the system dynamics. However, now your feedback gain K is being produced by the linear quadratic regulator, so tuning Q will alter K and thus the observer gain L may have to be tuned concurrently. **Make sure** to change your output matrix, C, such that α is not an output. Use the estimate of the beam's deflection angle, $\hat{\alpha}$, to calculate an estimate for the angular velocity of the beam via the high-gain observer subsystem. Plot the responses of the rotor base angle and the flexible beam deflection angle due to a square wave input of amplitude 30 and frequency 0.33Hz. On the flexible beam deflection angle plot, also plot your state estimate (you'll need to use a To Workspace block for this). What are your gains L and K and what is your weighting matrix Q? What can you conclude about the optimality (i.e., rotor base trajectory tracking and beam's deflections) of this control system when comparing it to the partial state feedback scenario and the full state feedback scenario?

Answer: with Q weightings of $[q_1,\ q_2,\ q_3,\ q_4] = [130,\ 1,\ 1,\ 3.5]$, the resulting feedback

gain weightings are $[k_1,\ k_2,\ k_3,\ k_4] = [11.401,\ -25.031,\ 1.379,\ -0.436]$. The observer gain weightings are $[l_1,\ l_2,\ l_3,\ l_4] = [-40,\ -41,\ -42,\ -43]$. The plots below reveal that the state estimate, $\hat{\alpha}$, and α agree pretty well. As a result of using partial state feedback and state estimate feedback, the magnitude of the flexible beam deflections is reduced throughout in comparison to the partial state feedback closed-loop system. In comparison to the full state feedback system, the overall flexible beam deflections are greater, and the rotor angle response is almost the same. The control input for the state estimate feedback system is slightly less than that for the full state feedback system, although this decrease is only weighted by 1 in the cost function, whereas beam deflection is weighted by 3.5. Thus using the state estimates as feedback is a more optimal control system than using only partial feedback, but not as optimal as using full state feedback (as is expected).

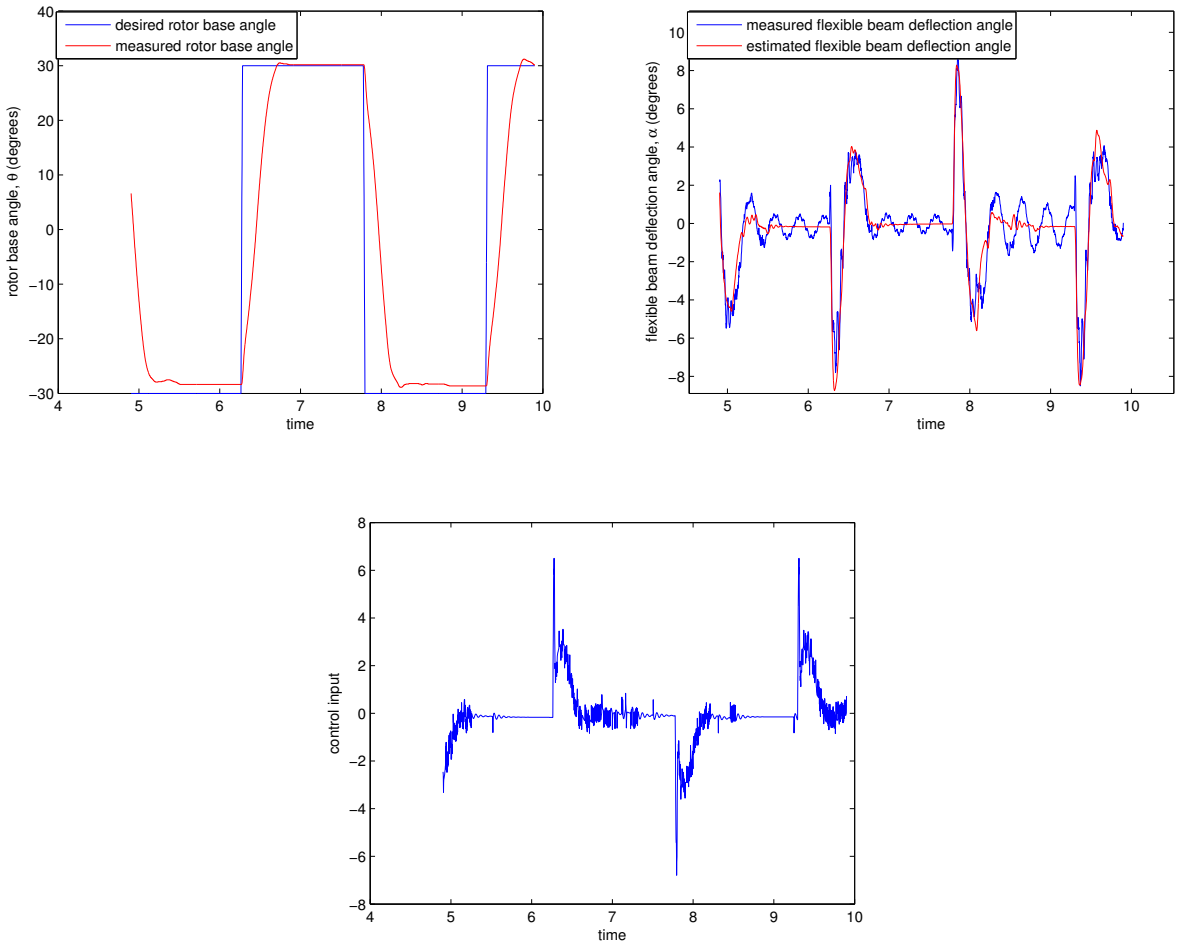


Figure 41: a) The rotor base angle, and (b) the flexible beam deflection angle responses to a square wave input when using partial state feedback θ, $\dot{\theta}$ and state estimate feedback, $\hat{\alpha}$, $\dot{\hat{\alpha}}$. This control system is less optimal than the full state feedback control system, but more optimal than the partial state feedback system.

6.6 Deliverables

To recap the lab objectives, this is what you need to include in your lab report to obtain full marks:

(i) prelab question (i) and (ii);

(ii) the answers to questions Q3, Q4, Q5 and Q7 (include plots where requested).

Note: Congratulations on finishing your MTHE 430 labs! Celebrations are in order.

Bibliography

[1] J. Apiarian, P. Karam, and M. Lévis, *Flexible Link Experiment for MATLAB/Simulink Users*. Quanser, 2011.

[2] J. Apiarian, P. Karam, and M. Lévis, *Inverted Pendulum Experiment for MATLAB/Simulink Users*. Quanser, 2011.

[3] A. Lewis, *A mathematical approach to classical control*. Queen's University, 2004.

[4] A. M. Dabroom and H. K. Khalil, "Discrete-time implementation of high-gain observers for numerical differentiation," *International Journal of Control*, vol. 72, no. 17, pp. 1523–1537, 1999.

[5] D. G. Luenberger, "An introduction to observers," *IEEE Transactions on automatic control*, vol. 16, no. 6, pp. 596–602, 1971.

[6] K. J. Aström and R. M. Murray, *Feedback systems: an introduction for scientists and engineers*. Princeton university press, 2010.