Experiment 1: Introduction to PC-based Data Acquisition and Real-Time Control

Tools/concepts emphasized: MATLAB, Simulink, Real-Time Workshop (RTW), QuaRC, Q4, data acquisition, and real-time control.

1. Introduction

All real-world applications of feedback control involve:

i) mathematical modeling of physical plants;

ii) system/parameter identification;

iii) feedback control design;

iv) off-line computer simulation to evaluate closed-loop system performance;

v) real-time feedback control implementation using analog/digital hardware; and

vi) on-line controller adjustment to optimize closed-loop system performance.

You have been familiarized with steps *i*), *iii*), and *iv*) in Automatic Control–ME 3413. In the Automatic Control Laboratory–ME 3411, we will reiterate some aspects of steps *i*), *iii*), and *iv*), as required; however, our primary focus will be on steps *ii*), *v*), and *vi*).

Traditionally, control systems have been designed and analyzed using analog methods such as the Laplace transform. In addition, until 1960's, a vast majority of industrial control systems were implemented using analog technology based on mechanics (e.g., moving bars, linkages, etc.), pneumatics, and electronics (e.g., resistors, capacitors, op-amps, etc.). However, with the advent of digital computer technology, control engineering has witnessed a significant shift towards digital implementation of feedback controllers [1]. In contrast to analog implementation of feedback control, digital implementation offers small size and low cost. Furthermore, digital controllers are inherently flexible since they can be changed by reprogramming, whereas analog controllers are changed by extensive rewiring [1].

In many current industrial and commercial applications of feedback control such as machine tools, robotics, automotive system, etc., microcontrollers are extensively used. Microcontrollers are typically programmed either in low-level machine language or in high-level languages such as C via PC interfaces. The programming of microcontrollers for implementing advanced control algorithms is a specialized task and requires trained personnel. However, in the last decade, with the advent of the fourth generation computer programming tools such as the computer-aided software engineering (CASE), it has become feasible to automatically generate C code from graphical control-system simulation tools such as Simulink. In particular, using the Simulink block library and RTW along with vendor-specific block libraries, one can generate C code from Simulink-based feedback control diagrams for real-time controller implementation on PC and DSP-based data acquisition and control boards (DACBs).

In the first laboratory exercise, we will focus on gaining familiarity with the Q4 DACB [2] and MATLAB, Simulink, RTW, and QuaRC [3] software. The Q4 DACB provides the following functionalities: analog to digital conversion (ADC), digital to analog conversion (DAC), digital I/O, and encoder readout. A Simulink compatible block library of Q4 functions is provided on each laboratory PC. The QuaRC software provides a user friendly graphical user interface (GUI) for implementing Simulink-based real-time control on the Q4 DACB. In addition, QuaRC can be used to display real-time experimental data on PCs. In this experiment, students will learn the basic functionalities of the Q4 DACB, QuaRC, and Simulink automated code generation features by implementing a simple loop-back example.

2. Background

In this section, we provide a brief overview of the hardware and software environment to be used throughout this laboratory course.

Q4 DACB: The Q4 is a general purpose DACB. It provides 4 single-ended ADCs, 4 DACs, 16 bits of digital inputs, 16 bits of digital outputs, 4 reconfigurable encoder counter/timers, and upto 4 encoder inputs. The Q4 DACB is accessed through the PC bus and is installed on an PCI bus internal to the laboratory PC. The aforementioned functions of the Q4 DACB can be accessed via an external terminal board.

MATLAB-Simulink-RTW: This is the preferred software environment for the control laboratory. Students enrolled in this laboratory course were familiarized with the MATLAB software in ME 3413. Simulink is a graphical control-system simulation program. The RTW toolbox enables automated C code generation from user-designed Simulink control-system diagrams.

QuaRC_Library: This is a library of Quanser-supplied DACB drivers (e.g., Q4) compatible with Simulink (See Figure 1). Some commonly used blocks of the Q4 library are HIL Read Analog (ADC), HIL Write Analog (DAC), and HIL Read Encoder (See Figure 2).

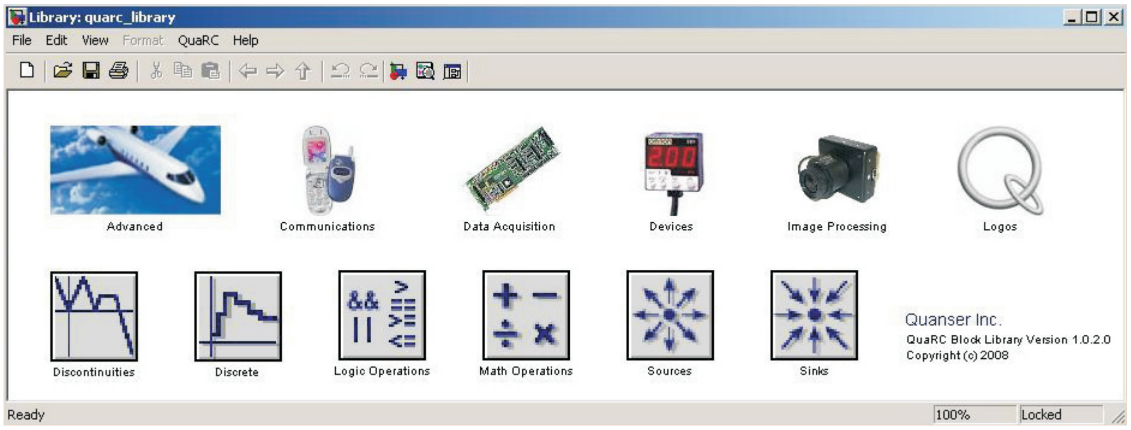

Figure 1: QuaRC_Library Block Library

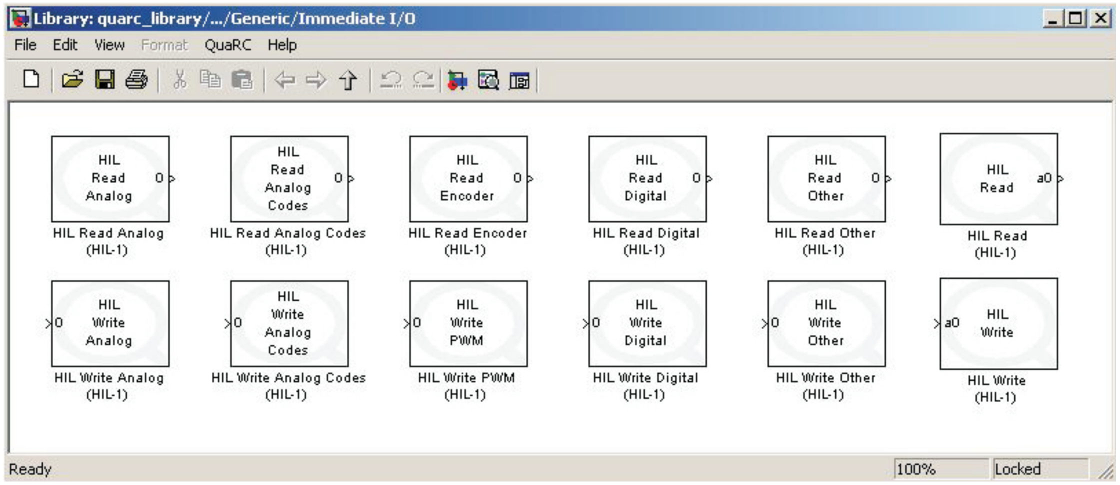

Figure 2: Immediate I/O Block Library

QuaRC: The QuaRC program interfaces the Simulink generated C code with the Q4 board in a seamless manner. The QuaRC program consists of two principal components, *viz.*, QuaRC client and QuaRC server. The QuaRC client is installed on the host computer with the Q4 DACB. The QuaRC server may be installed on the host or the remote computer. The user designs a Simulink control diagram and generates the C code on the remote computer. The C code from the remote computer is transferred to the host computer by the QuaRC server. The QuaRC client and host computer's processor communicate with the Q4 DACB for real-time data acquisition and control. The QuaRC client also relays the real-time data to the QuaRC server for plotting purposes.

3. Objective

i) Gain familiarity with various functions of the Q4 board.

ii) Learn the laboratory software environment consisting of MATLAB, Simulink, RTW, QuaRC, and QuaRC_Library.

iii) Design and implement a simple loop-back control system.

4. Equipment List

i) PC with Q4 DACB and terminal board

ii) Software environment: Windows, MATLAB, Simulink, RTW, and QuaRC

iii) Set of leads

5. Experimental Procedure

In this experiment, we will design a controller that outputs a user specified voltage to a selected DAC channel and measures the incoming voltage at a selected ADC channel.

i) Using the Q4 terminal board and a double-ended RCA connector, connect the channel 0 of DAC (analog output) to the channel 0 of ADC (analog input), as illustrated in Figure 3.

ii) From the **Start** button of the Windows toolbar, select the option sequence **Programs–MATLAB–R2007b–MATLAB R2007b** to launch the MATLAB application.

iii) In the MATLAB window, choose "C:\ControlLab\Experiment1" from the Current Directory window. This directory path choice will change the directory from the default MATLAB directory to the working directory for Experiment 1.

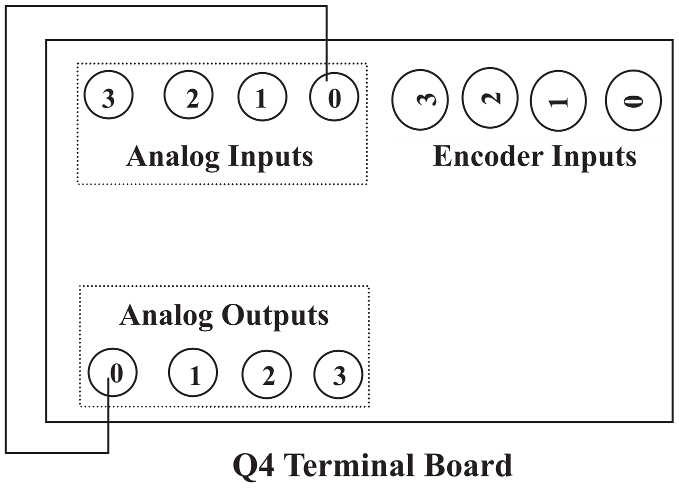

Figure 3: Wiring Diagram for the Loop-Back Experiment

iv) In the MATLAB window, at the command prompt, type Simulink and hit the **Enter** key. Next, in the MATLAB window, type quarc_library and hit the **Enter** key. The preceding two commands open the Simulink and the Q4 DACB drivers libraries, respectively.

v) From the Simulink tool bar, select **File–Open** to open "Template.mdl" file. The file "Template.mdl" is a blank Simulink model. This file has been created with a set of RTW options that enable C code generation for Visual C^{++}, RTX (a real-time kernel for Windows XP), and Q4 environment. You can determine the selected RTW-specific parameters by following the option sequence **Tools–RTW Options**. Please **do not** change any of the parameters while doing this.

vi) From the QuaRC_Library, to access Immediate I/O block library as shown in Figure 3, select **Data Acquisition–Generic–Immediate I/O** by double clicking each block. Then drag the icons labelled HIL Read Analog and HIL Write Analog into the blank "Template.mdl" model file. The HIL Initialize block can be accessed by selecting **Data Acquisition–Generic–Configuration**. In addition, from the Simulink block library, under the icons Sources and Sinks, select and drag the icons labelled Constant, Scope, and Display, respectively, into the "Template.mdl" model file. Using the copied icons, complete a Simulink block-diagram as shown in Figure 4. Next, set the value of the constant under the icon Constant to 1, to output 1 volt at the DAC. In addition, set the channel numbers under the icons HIL Read Analog and HIL Write Analog to 0. Finally, save the completed Simulink control-system diagram as “Experiment1.mdl.”

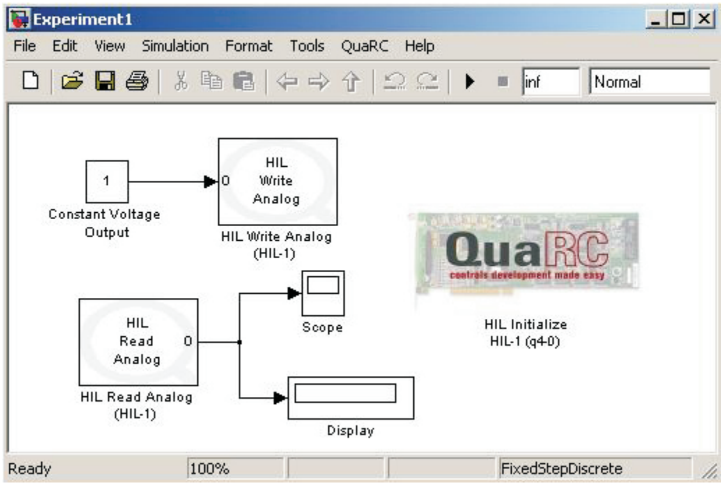

Figure 4: Simulink Block Diagram for the Loop-Back Experiment

vii) From the toolbar of “Experiment1.mdl” file, select the option sequence **QuaRC–Build** to link, compile, and generate the C^{++} code for the Simulink diagram. After the completion of C^{++} code generation process, MATLAB window shows “Model Experiment1 has been downloaded to target $\cdots$.”

viii) You can now perform the loop-back experiment. However, before proceeding, you **must** request your laboratory teaching assistant to approve your electrical connections and your Simulink control-system diagram.

ix) In the MATLAB window, click the black **Start** arrow button to acquire the real-time data for the loop-back experiment. You can change the output voltage at the DAC by changing the value of the constant in the Constant icon. Try experimenting, without exceeding the constant value by 5 volts.

x) After sufficient experimentation, press the black **Stop** square button in the MATLAB window to stop execution of your program on the Q4 DACB.

xi) Explore and document various menu options available in the QuaRC Server program.

6. Analysis/Assignment

i) In step *ix)* of Section 5, what is the value of the scope variable, displayed in the Dispaly, when you change the constant voltage applied at the DAC from 1 volt to 4 volt? Explain.

ii) Based on the loop-back experiment, develop a Simulink control-system diagram to run a diagnostic test on the 4 DAC and 4 ADC channels available on the Q4 DACB.

iii) Briefly explain the principle of operation of ADC and DAC.

References

1. K. J. Åström and B. Wittenmark, *Computer-Controlled Systems: Theory and Design*, Prentice-Hall, Upper Saddle River, NJ, 1997, 3rd Ed.

2. Online: http://www.quanser.com/english/html/solutions/fs_q4.html, website of Quanser Consulting Inc., (access link for Q4).

3. Online: http://www.quanser.com/english/html/solutions/fs_soln_software_QuaRC.html, website of Quanser Consulting Inc., (access link for QuaRC).

Experiment 2: System Identification and Control of an Electrical Network

Concepts emphasized: Passive filters, dynamic modeling, time-domain analysis, system type, and integral control.

1. Introduction

Physical measurements using electro-mechanical sensors are commonly performed by engineers. For example, a potentiometer can be used for position measurement of machine-bed traverse in lathes, milling machines, etc. Similarly, a thermocouple can be used for temperature measurement in process plants. Unfortunately, a vast majority of measurement sensors output spurious noise signals corrupting the measured quantities [1]. Electrical networks are often designed to filter the undesired noise from the sensor measurement. One such filter is the passive, low-pass R-C filter shown in Figure 1 [1]. This laboratory exercise is designed to provide the students fundamental principles of electrical network modeling, system identification, and closed-loop control. Specifically, the first part of this laboratory experiment exposes the students to the powerful techniques of ordinary differential equations and the Laplace transform for mathematical modeling of real-world dynamical systems [2, 3]. Next, the students learn to analyze the system time response to determine the unknown physical parameters of the system [2, 3]. Finally, the students design a feedback control system to manipulate the system characteristic such that the closed-loop system response follows a desired specification [2, 3].

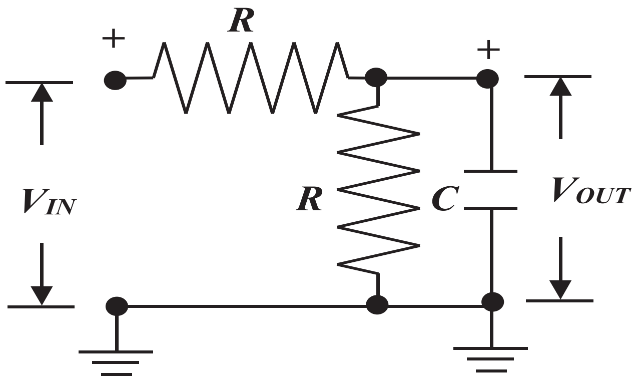

Figure 1: An R-C Filter Network

2. Background

Resistor: The voltage-current law governing a linear resistor is given by [1, 3]

$$R = \frac{V}{i}, \tag{2.1}$$

where i is the current flow through the resistor R when a voltage V is applied across the terminals of R. A resistor element is conventionally drawn as shown in Figure 2. Units: V (Volt–V), i (Ampere–Amp), R (Ohm–$\Omega = \frac{V}{Amp}$).

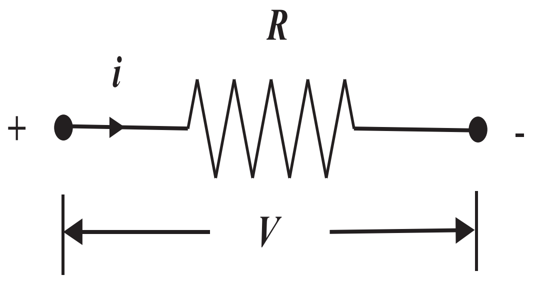

Figure 2: Diagrammatic Representation of a Resistor Element

Capacitor: A capacitor is constructed by introducing a nonconducting medium within the gap between two conductors. A capacitor can accumulate electric charge and can thus be used as an energy storage device (analogous to a spring in a mechanical system). The mathematical law governing the operation of a capacitor is given by [1, 3]

$$C = \frac{q}{V}, \tag{2.2}$$

where q is the amount of electric charge stored in the capacitor when a voltage V is applied across the terminals of C. Note that since

$$i = \frac{dq}{dt}, \tag{2.3}$$

using (2.2), Eq. (2.3) yields

$$i = C\frac{dV}{dt}. \tag{2.4}$$

A capacitor element is conventionally drawn as shown in Figure 3. Units: q (Coulomb), V (Volt–V), C (Farad $= \frac{Coulomb}{V}$).

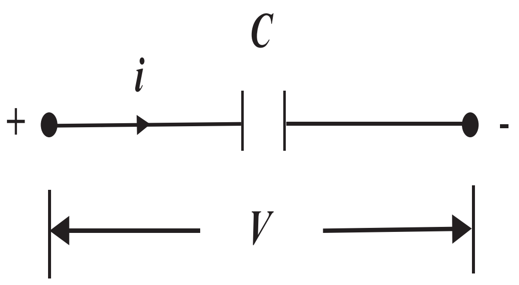

Figure 3: Diagrammatic Representation of a Capacitor Element

Kirchhoff's Current Law: The Kirchhoff's Current Law (KCL) states that the algebraic sum of all currents entering and leaving a node is zero [1, 3]. Thus, in Figure 4 at node A

$$i_1 + i_2 - i_3 = 0, \tag{2.5}$$

which can be rewritten as

$$i_3 = i_1 + i_2. \tag{2.6}$$

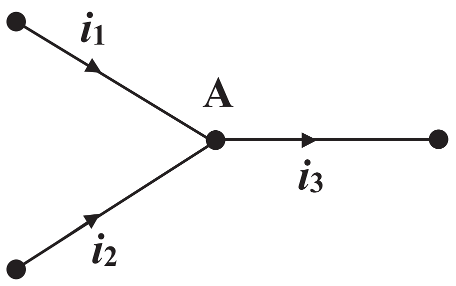

Figure 4: Current Flow at a Node

Step Response Analysis of a First-Order System: Consider the transfer function of a first-order system given by

$$\frac{Y(s)}{U(s)} = \frac{\alpha}{s+\beta}. \tag{2.7}$$

The step response of (2.7) can be obtained by computing the inverse Laplace transform of

$$Y(s) = \frac{\alpha}{s+\beta} \times \frac{A}{s}, \tag{2.8}$$

where $\frac{A}{s}$ is the Laplace transform of the step input of magnitude A applied at time $t = 0$. Next, the inverse Laplace transform of (2.8) yields

$$y(t) \;=\; \frac{A\alpha}{\beta}\left[1 - \mathrm{e}^{-\beta t}\right]. \tag{2.9}$$

A typical unit step ($A = 1$) response plot for a first-order system is shown in Figure 5. Note that (2.9) can be used to compute the steady-state response of (2.7) for the step input A. Alternatively, the final value theorem can be applied to (2.8) to obtain the steady-state response of (2.7) for the step input A [2, 3].

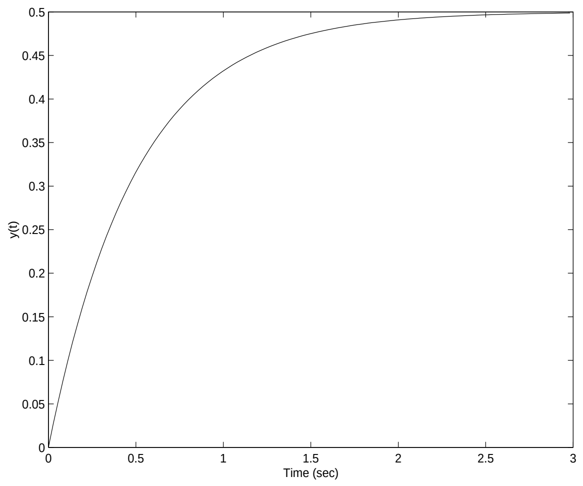

Figure 5: Unit Step Response of $\frac{1}{s+2}$

System Identification from Step Response: Consider the special case of (2.7) where $\frac{\alpha}{\beta} = D$ and D is known. In this case (2.9) can be rewritten as

$$y(t) \;=\; DA\left[1 - \mathrm{e}^{-\beta t}\right]. \tag{2.10}$$

The goal is to use (2.10) and the experimental step response data to determine the unknown system parameter β. By simple algebraic manipulation of (2.10), we obtain

$$\beta \;=\; -\frac{1}{t} \times \ln\left[\frac{DA - y(t)}{DA}\right]. \tag{2.11}$$

Next, with the known D and the magnitude of the step input A and by selecting a specific time instance t^*, within the transient response region, and the corresponding $y(t^*)$ from the experimental data, Eq. (2.11) can be used to determine β.

3. Objective

i) Modeling of the passive R-C network shown in Figure 1.

ii) Open-loop step response analysis for system identification.

iii) Integral control design for zero steady-state error response.

4. Equipment List

i) PC with Q4 data acquisition card and terminal board

ii) Software environment: Windows, MATLAB, Simulink, RTW, and QuaRC

iii) Two resistors of 100 K Ω

iv) One capacitor of unknown capacitance value

v) Set of leads and a breadboard

5. Experimental Procedure

i) Using the breadboard, set of leads, 100 K Ω resistors, and the capacitor of unknown capacitance, construct the electric network shown in Figure 6.

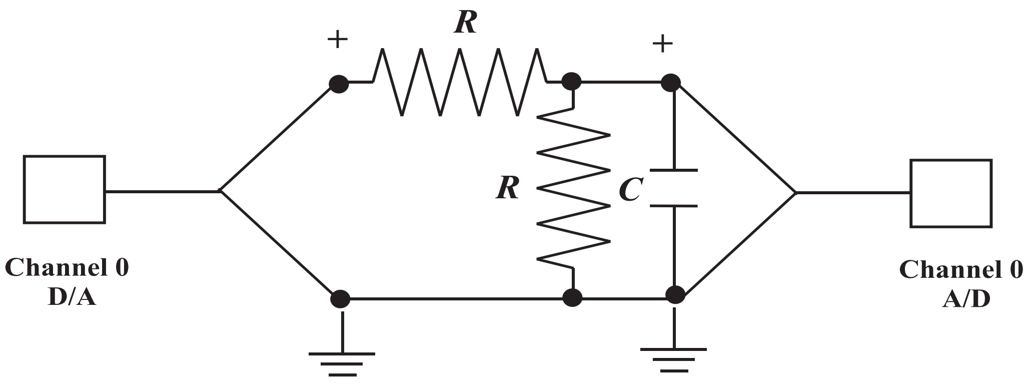

Figure 6: Wiring Diagram for the R-C Filter Network

ii) Start MATLAB using the procedure described in laboratory Experiment 1. In the MATLAB window, choose "C:\ControlLab\Experiment2" from the Current Directory window.

This directory path choice will change the directory from the default MATLAB directory to the directory where all files needed to perform Experiment 2 are stored.

iii) From the **File** menu of the MATLAB window, select the option **Open** to load the Simulink block-diagram "Experiment2a.mdl." This will load the plot window shown in Figure 7 and the model file for Experiment 2a (open-loop) shown in Figure 8 will appear on your desktop.

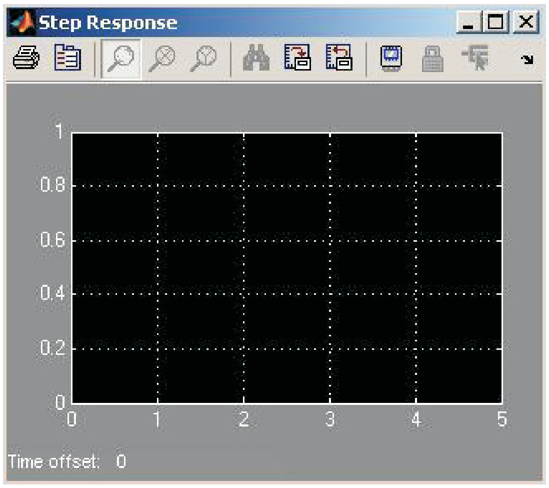

Figure 7: Plot Window for the Open-Loop Step Response of the R-C Network

iv) You can now perform an open-loop analysis of the electrical network shown in Figure 1. <u>However, before proceeding, you **must** request your laboratory teaching assistant to approve your electrical connections.</u>

v) In the MATLAB window, click the black **Start** arrow button to acquire the open-loop step response of the R-C electrical network. The experiment stops after 5 seconds.

vi) In the Simulink block-diagram, the To File block creates "Exp2AData.mat" which the plot data is saved on. Plot the open-loop step response from the MATLAB window by executing the following commands: load Exp2AData and plot(data2A(1,:),data2A(2,:)).

vii) Close the currently open plot windows and the Simulink diagram. From the **File** menu of the MATALB window, select the option **Open** to load the Simulink block-diagram "Experiment2b.mdl" shown in Figure 9 to your desktop. This will load the files for

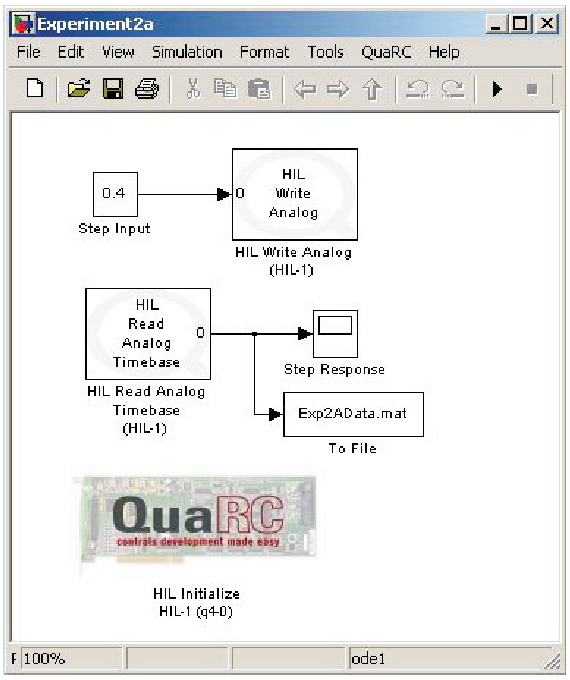
Experiment2a
File Edit View Simulation Format Tools QuaRC Help
0.4
Step Input
HIL Write Analog
HIL Write Analog (HIL-1)
HIL Read Analog Timebase
HIL Read Analog Timebase (HIL-1)
Step Response
Exp2AData.mat
To File
QuaRC controls development made easy
HIL Initialize HIL-1 (q4-0)
100%
ode1

Figure 8: Simulink Block-Diagram for the Open-Loop Step Response of the R-C Network

experiment 2 (closed-loop) and a plot window similar to the one shown in Figure 7 will appear on your desktop. Note that the feedback interconnection of the R-C circuit and the Simulink controller in Figure 9 (ignoring the saturation block) can be represented as shown in the closed-loop feedback diagram of Figure 10.

viii) In the MATLAB window, click the black **Start** arrow button to acquire the closed-loop step response of the R-C electrical network. The experiment stops after 20 seconds.

ix) In the Simulink block-diagram, the To File block creates "Exp2BData.mat" which the plot data is saved on. Plot the closed-loop step response from the MATLAB window by executing the following commands: load Exp2BData and plot(data2B(1,:),data2B(2,:)).

6. Analysis

i) Obtain the differential equation governing the response of the R-C circuit shown in Figure 1. In addition, determine the transfer function that maps the input voltage V_{IN} to the output voltage V_{OUT}; i.e., determine the transfer function $\frac{V_{\text{OUT}}(s)}{V_{\text{IN}}(s)}$.

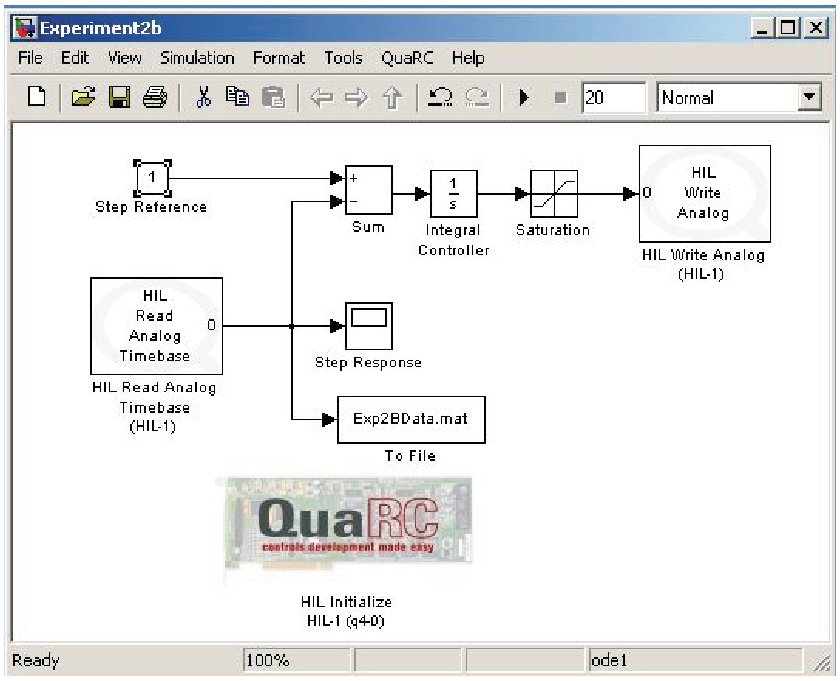

Figure 9: Simulink Block-Diagram for the Integral Control of the R-C Network

ii) Analyze the open-loop step response obtained in step *vi*) of Section 5 to *a*) determine the unknown capacitance value for the capacitor and *b*) determine the steady-state error for the applied step input. For part *a*), note that the terminal board of the Q4 data acquisition card introduces a capacitor of 1 μF in parallel to the unknown capacitor C in Figure 1. You must write a **function** .m file which accepts V_{IN}, R, t, and $V_{\text{OUT}}(t)$, as input arguments and returns the unknown capacitance value as the output.

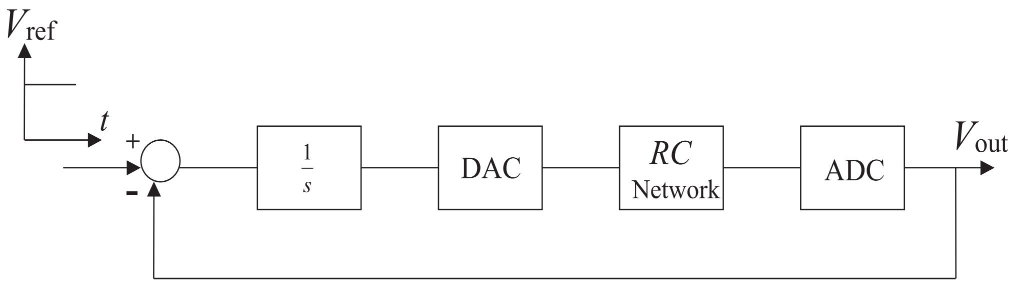

Figure 10: Closed-Loop Feedback Diagram of the R-C Network with Integral Controller

iii) Obtain the step response of the R-C network using Simulink. Compare the simulated response with the actual response and comment.

iv) Analyze the closed-loop step response obtained in step *ix*) of Section 5 to determine the steady-state error for the step input.

v) Design a proportional-plus-integral controller $\left(K_\mathrm{p} + \frac{K_\mathrm{i}}{s}\right)$ so that the step response of the closed-loop system has less than 5% overshoot and the settling time $T_\mathrm{s} \leq 0.3$ seconds. Simulate the closed-loop system step response using Simulink.

References

1. W. Bolton, *Mechatronics: Electronic Control Systems in Mechanical and Electrical Engineering*, Addison Wesley, New York, NY, 1999.

2. R. C. Dorf and R. H. Bishop, *Modern Control Systems*, Addison Wesley, Menlo Park, CA, 1998.

3. K. Ogata, *Modern Control Engineering*, Prentice-Hall, Upper Saddle River, NJ, 1997.

Experiment 3: Modeling, Identification, and Control of a DC-Servomotor

Concepts emphasized: Dynamic modeling, time-domain analysis, system identification, and position-plus-velocity feedback control.

1. Introduction

DC-motors that are used in feedback controlled devices are called DC-servomotors [1, 2, 3, 4]. Applications of DC-servomotors abound, e.g., in robotics, computer disk drives, printers, aircraft flight control systems, machine tools, flexible manufacturing systems, automatic steering control, etc. DC-motors are classified as *armature controlled* DC-motors and *field controlled* DC-motors [4].

This laboratory experiment will focus on the modeling, identification, and position control of an armature controlled DC-servomotor. In particular, we will first develop the governing differential equations and the Laplace domain transfer function model of an armature controlled DC-motor. Next, we will focus on the identification of the unknown system parameters that appear in the transfer function model of the DC-servomotor. Finally, we will develop and implement a position-plus-velocity, also known as proportional-plus-derivative (PD), feedback controller to ensure that the DC-motor angular position response tracks a step command.

2. Background

DC-motor modeling: A schematic representation of an armature controlled DC-motor is given in Figure 1. For an armature controlled DC-motor, the field current $i_{\rm f}$ is constant and the torque $T_{\rm m}$ generated at the DC-motor shaft is given by [2, 3, 4]

$$T_{\rm m} = K_{\rm T} i_{\rm a}, \tag{2.1}$$

where $K_{\rm T}$ is the given motor torque constant ($\frac{N-m}{Amp}$) and $i_{\rm a}$ is the armature current (Amp). Note that for an armature controlled DC-motor, the back e.m.f. induced in the armature due to armature rotation is directly proportional to the armature angular velocity $\omega_{\rm m}(t) \triangleq \frac{d\theta_{\rm m}}{dt}$ where $\theta_{\rm m}(t)$ is the angular position of the motor shaft. Thus, following [2, 3, 4]

$$V_{\rm b} = K_{\rm b}\frac{d\theta_{\rm m}}{dt}, \tag{2.2}$$

where K_{b} is a given motor constant ($\frac{V-sec}{rad}$).

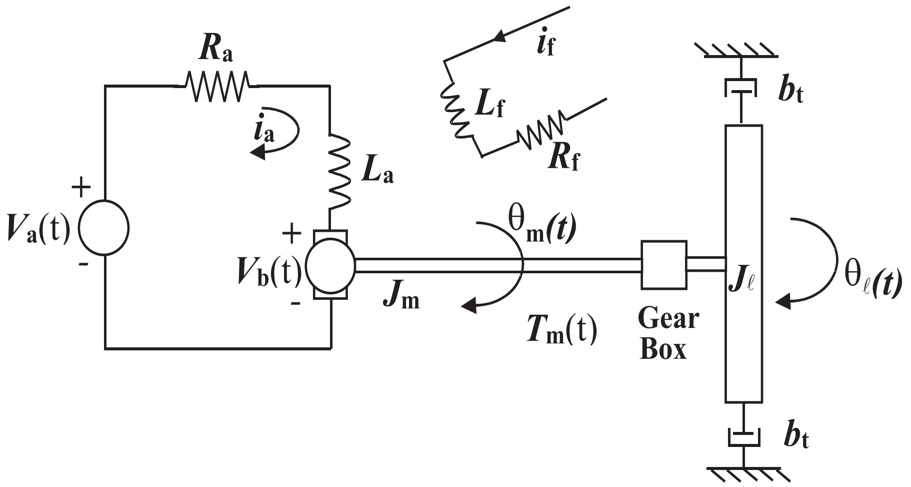

Figure 1: Armature Controlled DC-Motor

Next, note that the angular speed $\omega_{\mathrm{m}}(t)$ of an armature controlled DC-motor is controlled by the armature voltage V_{a}. The differential equation relating the armature current i_{a} and the back e.m.f. V_{b} to the armature voltage V_{a} can be obtained by applying Kirchhoff's Voltage Law (KVL) [1, 4]. In particular, according to the KVL, at any given instant of time, the algebraic sum of voltages around any loop in any electric network is zero. Thus, a direct application of the KVL to the armature circuit yields

$$L_{\mathrm{a}}\frac{di_{\mathrm{a}}}{dt} + R_{\mathrm{a}}i_{\mathrm{a}} + V_{\mathrm{b}} = V_{\mathrm{a}}. \tag{2.3}$$

Finally, we obtain the differential equation governing the motion of the mechanical load. First, note that in most applications, the DC-servomotor shaft is connected to a gear-box of a given gear-ratio K_{g} and the load is attached to the output shaft of the gear-box (e.g., see Figure 2). The gear-ratio K_{g} is give by $K_{\mathrm{g}} \overset{\triangle}{=} \frac{n_\ell}{n_{\mathrm{m}}}$, where n_ℓ and n_{m} are the number of teeth on the load-side and the motor-side gears, respectively. It can be easily shown that the gear-ratio K_{g} relates the motor shaft angular position θ_{m} to the gear-box output shaft angular position θ_ℓ by $K_{\mathrm{g}} = \frac{\theta_{\mathrm{m}}}{\theta_\ell}$. In addition, it can be shown that the load inertia J_ℓ acting at the output shaft of the gear-box when reflected at the motor shaft is given by $\frac{1}{K_{\mathrm{g}}^2}J_\ell$. Thus, an application of Newton's moment balance equation at the motor output shaft yields

$$J_{\mathrm{m}}\frac{d^2\theta_{\mathrm{m}}}{dt^2} + \frac{1}{K_{\mathrm{g}}^2}J_\ell\frac{d^2\theta_{\mathrm{m}}}{dt^2} + \frac{1}{K_{\mathrm{g}}^2}b_{\mathrm{t}}\frac{d\theta_{\mathrm{m}}}{dt} = T_{\mathrm{m}},$$

which can be rewritten as

$$J_{\rm eq}\frac{d^2\theta_\ell}{dt^2} + b_{\rm t}\frac{d\theta_\ell}{dt} \ = \ K_{\rm g}T_{\rm m}, \tag{2.4}$$

where $J_{\rm eq} = K_{\rm g}^2 J_{\rm m} + J_\ell$ is the total load inertia reflected at the motor shaft and $b_{\rm t}$ is the rotational viscous friction constant.

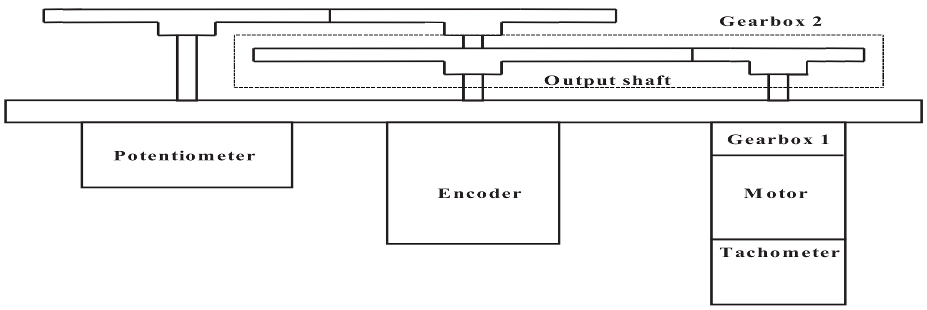

Figure 2: DC-Motor Experiment Test-Bed

Now, taking the Laplace transform of (2.1) and (2.4) and after some algebraic manipulations to eliminate the variables $T_{\rm m}$, $V_{\rm b}$, and $i_{\rm a}$, we obtain

$$\frac{\theta_\ell(s)}{V_{\rm a}(s)} \ = \ \frac{K_{\rm g}K_{\rm T}}{s\left(L_{\rm a}J_{\rm eq}s^2 + (L_{\rm a}b_{\rm t} + R_{\rm a}J_{\rm eq})s + R_{\rm a}b_{\rm t} + K_{\rm g}^2 K_{\rm T}K_{\rm b}\right)}. \tag{2.5}$$

In addition, the transfer function from input $V_{\rm a}$ to output ω_ℓ is given by

$$\frac{\omega_\ell(s)}{V_{\rm a}(s)} \ = \ \frac{K_{\rm g}K_{\rm T}}{L_{\rm a}J_{\rm eq}s^2 + (L_{\rm a}b_{\rm t} + R_{\rm a}J_{\rm eq})s + R_{\rm a}b_{\rm t} + K_{\rm g}^2 K_{\rm T}K_{\rm b}}. \tag{2.6}$$

Now, assuming two real, simple roots of the characteristic equation of (2.6), *viz.*, $p_{\rm e}$ and $p_{\rm m}$, partial fraction expansion of (2.6) yields

$$\frac{\omega_\ell(s)}{V_{\rm a}(s)} \ = \ \frac{K_{\rm e}}{s + p_{\rm e}} + \frac{K_{\rm m}}{s + p_{\rm m}}. \tag{2.7}$$

Next, using the inverse Laplace transform, the forced response of the system (with zero initial condition) to the input $V_{\rm a}(t)$ is given by

$$\omega_\ell(t) \ = \ \int_0^t \left[K_{\rm e}{\rm e}^{-p_{\rm e}(t-q)} + K_{\rm m}{\rm e}^{-p_{\rm m}(t-q)}\right] V_{\rm a}(q)dq. \tag{2.8}$$

In most practical applications of armature controlled DC-motors, $p_{\rm e} >> p_{\rm m}$; i.e., the electrical subsystem responds considerably faster than the mechanical subsystem. Hence, the first exponential term in (2.8) decays rapidly. Thus, the response $\omega_\ell(t)$ in (2.8) is dominated by the mechanical subsystem $\frac{K_\mathrm{m}}{s+p_\mathrm{m}}$. For simplicity, in DC-servomotor control applications the influence of the electrical subsystem component $(\frac{K_\mathrm{e}}{s+p_\mathrm{e}})$ on the response $\omega_\ell(t)$ in (2.8) is commonly neglected [2, 3, 4]. This can alternatively be viewed as neglecting the armature inductance effect, L_a. This simplification yields a first-order transfer function model which relates the DC-motor load angular velocity response ω_ℓ to the armature voltage input V_a, and is given by

$$\frac{\omega_\ell(s)}{V_\mathrm{a}(s)} = \frac{K_\mathrm{g} K_\mathrm{T}}{R_\mathrm{a} J_\mathrm{eq} s + R_\mathrm{a} b_\mathrm{t} + K_\mathrm{g}^2 K_\mathrm{T} K_\mathrm{b}}. \tag{2.9}$$

Before proceeding, note that, it can be shown that in the SI-Units used for K_T and K_b, the numerical values of K_T and K_b are identical [3]. Finally, the transfer function model of (2.9) can be equivalently written as

$$\frac{\omega_\ell(s)}{V_\mathrm{a}(s)} = \frac{K}{\tau s + 1}, \tag{2.10}$$

where K and τ are the dc-gain and the mechanical time-constant of the DC servomotor, respectively.

3. Objective

i) Analysis of DC-motor sensor characteristics.

ii) DC-motor system identification.

iii) PD control of the DC-motor to achieve the desired angular position step response characteristics.

4. Equipment List

i) PC with Q4 data acquisition card and terminal board

ii) Software environment: Windows, MATLAB, Simulink, RTW, and QuaRC

iii) SRV-02 DC-motor apparatus (See Figure 3) with potentiometer, optical encoder, and tachometer

iv) Universal power module: UPM-1503

v) Set of leads

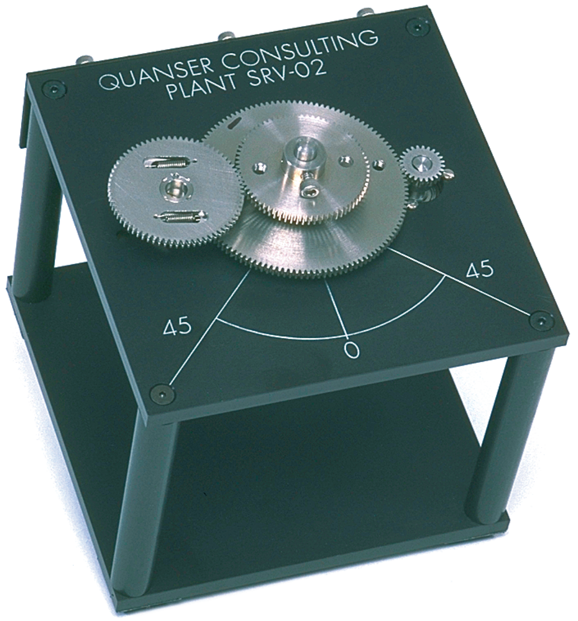

Figure 3: SRV-02 DC-Motor Apparatus

5. Experimental Procedure

i) Using the set of leads, universal power module, SRV-02 DC-motor apparatus, and the terminal board of the Q4 data acquisition card, complete the wiring diagram shown in Figure 4.

ii) Start MATLAB. In the MATLAB window, choose "C:\ControlLab\Experiment3" from the Current Directory window. This directory path choice will change the directory from the default MATLAB directory to the directory where all files needed to perform Experiment 3 are stored.

iii) You can now perform various steps of the DC-motor identification and control experiment. <u>However, before proceeding, you **must** request your laboratory teaching assistant to check your electrical connections.</u>

iv) From the **File** menu of the MATLAB window, select the option **Open** to load the Simulink block-diagram "Experiment3_Pot.mdl." shown in Figure 5 to your desktop. This will load

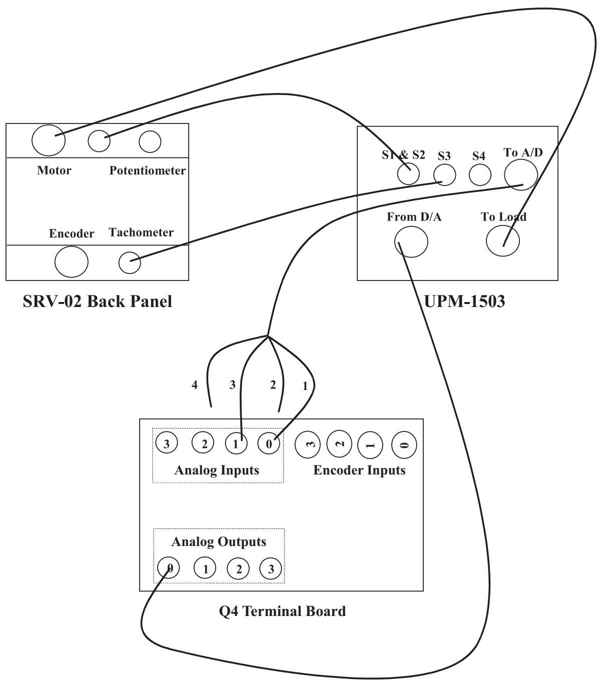

Figure 4: Wiring Diagram for DC-Motor ID and Control

the files for determining the gain of the potentiometer K_{pot} $(\frac{\text{radian}}{\text{Volt}})$. The potentiometer gain K_{pot} relates the potentiometer output voltage V_{pot} to the load angular displacement θ_ℓ by $\theta_\ell = K_{\text{pot}}V_{\text{pot}}$.

a) In the MATLAB window, click the black **Start** arrow button to acquire the potentiometer voltage response.

b) Rotate the load connected to the output shaft (center gear) until the potentiometer voltage in the Display block shows 0 Volt. Please ensure that you get continuous variation in the neighborhood of this 0 Volt reading. If you note a discontinuity in the reading, turn the load by 180° and this will provide you close to 0 Volt reading. Read the angular position θ_0° of the load, corresponding to the 0 Volt potentiometer reading, off the protractor marked on the SRV-02 apparatus.

c) Rotate the load to $\theta_0 + 90^\circ$ and note the corresponding potentiometer voltage reading in the Display block.

d) Rotate the load to $\theta_0 - 90^\circ$ and note the corresponding potentiometer voltage reading in the Display block.

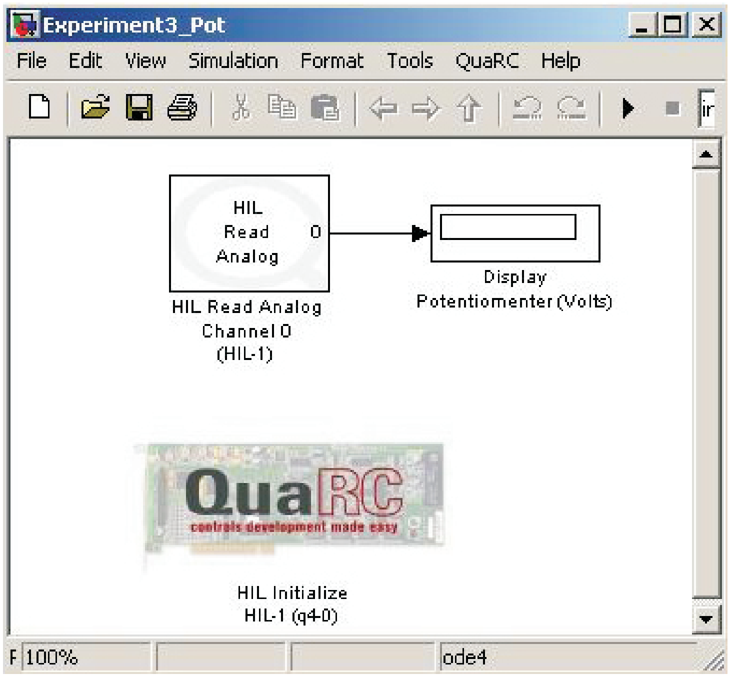
Experiment3_Pot
File Edit View Simulation Format Tools QuaRC Help
HIL Read Analog
0
Display Potentiometer (Volts)
HIL Read Analog Channel 0 (HIL-1)
QuaRC
controls development made easy
HIL Initialize HIL-1 (q4-0)
100%
ode4

Figure 5: Simulink Block-Diagram for Determining Potentiometer Gain

e) In the MATLAB window, click the black **Stop** square button when you finish collecting the potentiometer voltage response data.

v) Close the currently open the Simulink diagram. From the **File** menu of the MATLAB window, select the option **Open** to load the Simulink block-diagram "Experiment3_Tach.mdl." shown in Figure 6 to your desktop. This will load the files for determining the gain of the tachometer K_{tach} $(\frac{\text{radian}}{\text{second-Volt}})$ and a plot window. The tachometer gain K_{tach} relates the tachometer output voltage V_{tach} to the load angular velocity ω_ℓ by $\omega_\ell = K_{\text{tach}}V_{\text{tach}}$.

a) In the MATLAB window, click the black **Start** arrow button. This applies a constant 1 Volt input to the DC-motor.

b) Measure the steady-state load angular speed and the corresponding steady-state tachometer output voltage reading in the plot window. **Hint:** Find the time required for 20 complete revolutions of the load and the corresponding steady-state tachometer output voltage reading at the end of 20 revolutions.

c) In the MATLAB window, click the black **Stop** square button when you finish collecting the tachometer voltage response data.

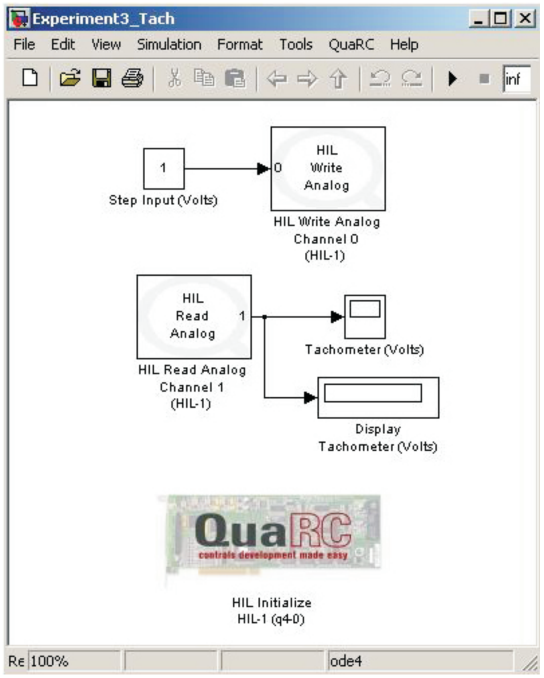
Experiment3_Tach
File Edit View Simulation Format Tools QuaRC Help
inf
1
Step Input (Volts)
0 HIL Write Analog
HIL Write Analog Channel 0 (HIL-1)
HIL Read Analog 1
HIL Read Analog Channel 1 (HIL-1)
Tachometer (Volts)
Display Tachometer (Volts)
QuaRC controls development made easy
HIL Initialize HIL-1 (q4-0)
Re 100%
ode4

Figure 6: Simulink Block-Diagram for Determining Tachometer Gain

vi) Close the currently open plot windows and the Simulink diagram. From the **File** menu of the MATLAB window, select the option **Open** to load the Simulink block-diagram "Experiment3_DCID.mdl." shown in Figure 7 to your desktop. In this diagram, the gain K_{tach} must be supplied by you. Run this part of the experiment to acquire the transient and steady-state angular velocity step response of the DC-motor under load.

vii) Close the currently open plot windows and the Simulink diagram. From the **File** menu of the MATLAB window, select the option **Open** to load the Simulink block-diagram "Experiment3_PDCont.mdl." shown in Figure 8 to your desktop. In this diagram, the gains K_{pot} and K_{tach} must be supplied by you. In addition, the gains K_{P} and K_{D} must be designed and supplied by you. In particular, design a PD feedback controller so that the DC-motor angular position step response exhibits a peak overshoot $M_{\text{p}} \leq 5\%$ with settling time $T_{\text{s}} \leq 1$ second. The feedback diagram of the DC-motor with the PD feedback controller is shown in Figure 9. The characteristic equation of the closed-loop system in Figure 9 can be used for the purpose of finding K_{P} and K_{D} such that the desired performance specifications are achieved. Before proceeding, you **must** request your laboratory

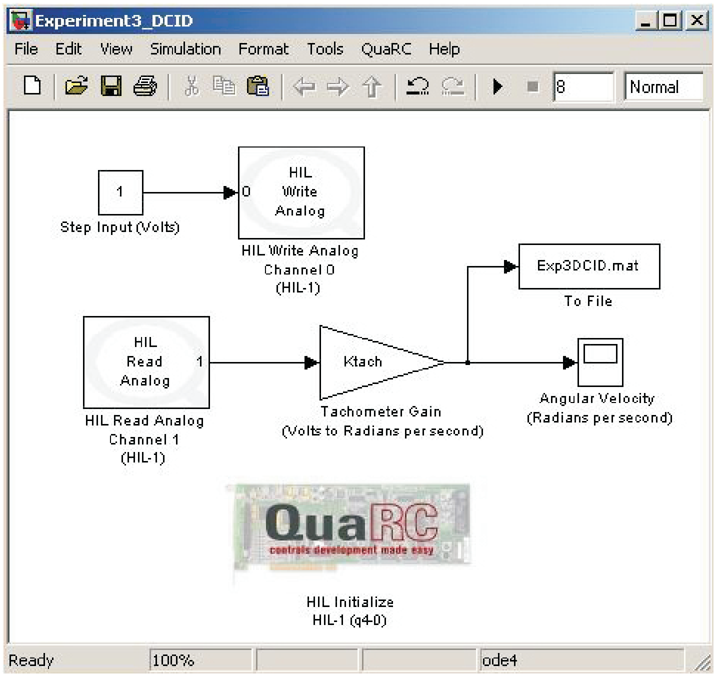
Experiment3_DCID
File Edit View Simulation Format Tools QuaRC Help
8
Normal
1
Step Input (Volts)
HIL Write Analog
HIL Write Analog Channel 0 (HIL-1)
Exp3DCID.mat
To File
HIL Read Analog
HIL Read Analog Channel 1 (HIL-1)
Ktach
Tachometer Gain (Volts to Radians per second)
Angular Velocity (Radians per second)
QuaRC controls development made easy
HIL Initialize HIL-1 (q4-0)
Ready
100%
ode4

Figure 7: Simulink Block-Diagram for DC-Servomotor System Identification

teaching assistant to approve your gain values. Run the experiment to record the angular position step response of the DC-motor.

6. Analysis

i) Calculate K_{pot} and K_{tach} from the experimental data collected in steps *iv)* and *v)* of Section 5.

ii) Analyze the open-loop angular velocity step response obtained in step *vi)* of Section 5 to determine the dc-gain K and the mechanical time constant τ of the DC-servomotor system.

iii) Obtain the angular velocity step response of the first-order system (2.10) with the parameters K and τ obtained in step *ii)* above. Compare the simulated angular velocity step response with the experimental response obtained in step *vi)* of Section 5 and comment.

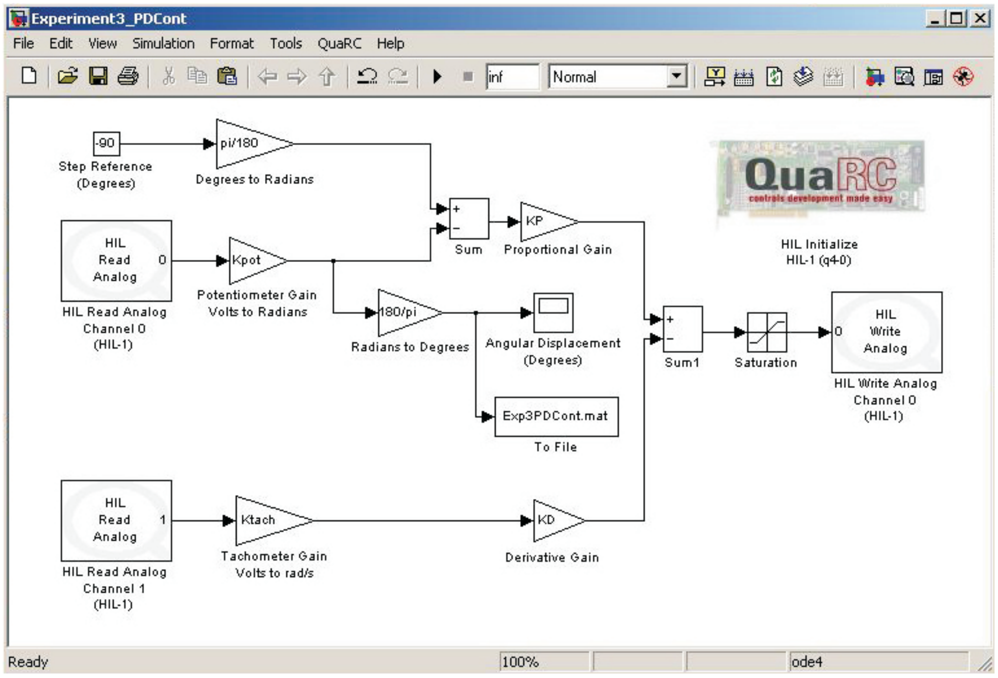
Experiment3_PDCont
File Edit View Simulation Format Tools QuaRC Help
inf
Normal
-90
Step Reference (Degrees)
pi/180
Degrees to Radians
HIL Read Analog
HIL Read Analog Channel 0 (HIL-1)
Kpot
Potentiometer Gain Volts to Radians
Sum
KP
Proportional Gain
180/pi
Radians to Degrees
Angular Displacement (Degrees)
Exp3PDCont.mat
To File
QuaRC
HIL Initialize HIL-1 (q4-0)
Sum1
Saturation
HIL Write Analog
HIL Write Analog Channel 0 (HIL-1)
HIL Read Analog Channel 1 (HIL-1)
Ktach
Tachometer Gain Volts to rad/s
KD
Derivative Gain
Ready
100%
ode4

Figure 8: Simulink Block-Diagram for DC-Servomotor PD Control

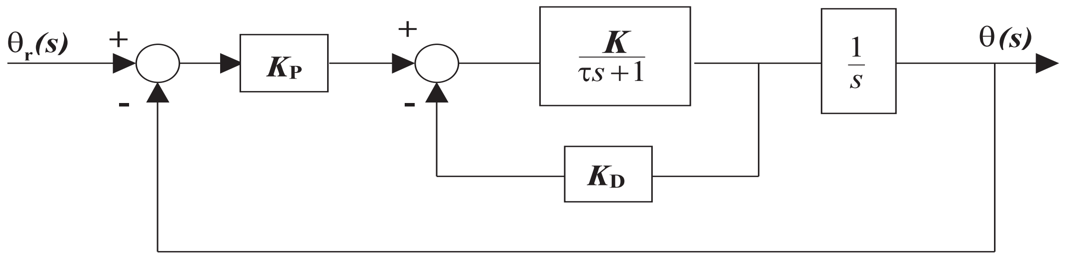

Figure 9: Closed-Loop Feedback Interconnection for PD Control of a DC-Motor

iv) Analyze the closed-loop angular position step response obtained in step *vii*) of Section 5 to determine if the performance specifications are satisfied. Comment on your results.

v) Design a proportional-integral-derivative (PID) controller so that the performance requirements specified in Section 5 are satisfied. Simulate the closed-loop angular position step response with the PID controller. Compare with the experimental closed-loop angular position step response obtained using the PD controller.

References

1. W. Bolton, *Mechatronics: Electronic Control Systems in Mechanical and Electrical Engineering*, Addison Wesley, New York, NY, 1999.

2. R. C. Dorf and R. H. Bishop, *Modern Control Systems*, Addison Wesley, Menlo Park, CA, 1998.

3. B. Kuo, *Automatic Control Systems*, Prentice-Hall, Englewood Cliffs, NJ, 1995.

4. K. Ogata, *Modern Control Engineering*, Prentice-Hall, Upper Saddle River, NJ, 1997.

Experiment 4: Modeling and Control of a Magnetic Levitation System

Concepts emphasized: Dynamic modeling, time-domain analysis, PI and PID feedback control.

1. Introduction

Magnetic levitation is becoming widely applicable in magnetic bearings, high-speed ground transportation, vibration isolation, etc., [1]. For example, magnetic bearings support radial and thrust loads in rotating machinery. In addition, magnetic suspension generates levitation action in rectilinear motion devices such as high-speed ground transportation systems. Magnetic levitation is immensely beneficial in the aforementioned rotary and rectilinear devices as it yields a non-contact support, without lubrication, thus eliminating friction. All practical magnetic levitation systems are inherently open-loop unstable and rely on feedback control for producing the desired levitation action.

The "maglev" experiment is a magnetic ball suspension system which is used to levitate a steel ball in air by the electromagnetic force generated by an electromagnet. The maglev system consists of an electromagnet, a ball rest, a ball position sensor, and a steel ball. The maglev system is completely encased in a rectangular enclosure divided into three distinct vertical chambers. The upper chamber houses an electromagnet such that one pole of the electromagnet is exposed to the middle chamber and faces a black post erect in the middle chamber. The post is designed such that with a 2.54cm steel ball at rest on its surface, the top of the ball surface is 14mm from the face of the electromagnet. The middle chamber is illuminated using a light bulb. The ball elevation from the top face of the post is measured using a sensor embedded in the post. The bottom chamber houses sensor circuitry for signal conditioning.

The objective of the experiment is to design a controller that levitates the steel ball from the post and makes it track a specified position trajectory. The maglev system can be decomposed into two subsystems, *viz.*, a mechanical subsystem and an electrical subsystem (current loop). The ball position in the mechanical subsystem can be controlled by adjusting the current through the electromagnet whereas the current through the electromagnet in the electrical subsystem can be controlled by applying controlled voltage across the electromagnet terminals. Thus, the voltage applied across the electromagnet terminals provides an indirect control of the ball position.

In this laboratory exercise, we will first develop the governing differential equation and the Laplace domain transfer function models of the electrical and mechanical subsystems. Next, we will design and implement a proportional-integral (PI) controller to guarantee that the electrical subsystem current response tracks the specified current command. Finally, we will design and implement a proportional-integral-derivative (PID) controller to ensure that the mechanical subsystem ball position response tracks the desired position command.

2. Background

Electrical Subsystem Modeling: A schematic representation of the maglev ideal electrical subsystem is given in Figure 1. The electromagnet coil has an inductance L (Henry) and a resistance R_ℓ (Ohm). The voltage V applied to the coil results in a current i governed by the differential equation [3]

$$V \;=\; iR_\ell + L\frac{di}{dt}. \tag{2.1}$$

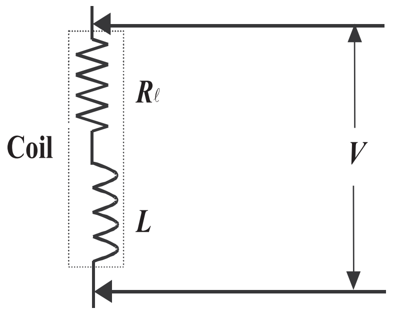

Figure 1: Ideal Electrical System

In order to determine the current in the coil, the mglev actual electrical subsystem (see Figure 2) is equipped with a resistor R_s in series with the coil such that the voltage V_s across R_s can be measured using an analog to digital converter. Now, the voltage V_s measured across R_s can be used to compute the current i in the coil. Note that with the sensing resistor R_s in the circuit the governing differential equation for the coil current becomes

$$V \;=\; i\,(R_\ell + R_\text{s}) + L\frac{di}{dt}. \tag{2.2}$$

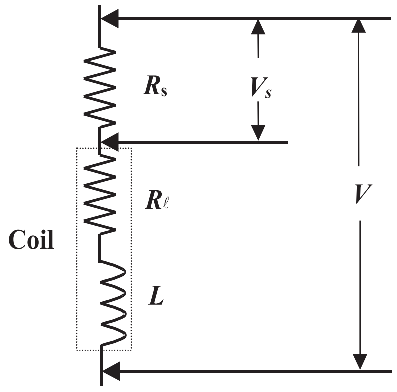

Figure 2: Actual Electrical System

Finally, taking the Laplace transform of (2.2), we obtain

$$G_e(s) \triangleq \frac{I(s)}{V(s)} = \frac{1}{Ls + (R_\ell + R_s)}, \tag{2.3}$$

where $I(s) \triangleq \mathcal{L}[i(t)]$ and $V(s) \triangleq \mathcal{L}[V(t)]$ and $\mathcal{L}$ is the Laplace operator.

Mechanical Subsystem Modeling: The force experienced by the ball under the influence of electromagnet is given by [2, 3]

$$F = mg - K_f \left(\frac{i}{x}\right)^2, \tag{2.4}$$

where i is the current in electromagnet (Ampere), x is the distance of the ball from the electromagnet face (mm), g is the gravitational constant ($\frac{\text{mm}}{\text{sec}^2}$), K_f is the magnetic force constant for the electromagnet-ball pair, and m is the mass of the steel ball (Kg). Using Newton's second law, we now obtain the differential equation governing the ball position as

$$m\frac{d^2x}{dt^2} = mg - K_f \left(\frac{i}{x}\right)^2. \tag{2.5}$$

Note that using (2.5), we can compute the steady-state electromagnet coil current i_{ss} that produces the desired steady-state constant ball position x_{ss}. Specifically, setting $\frac{d^2x}{dt^2} = 0$ in (2.5) yields

$$i_{ss} = \sqrt{\frac{mg}{K_f}} x_{ss}. \tag{2.6}$$

Now, theoretically one can use (2.6) to regulate the ball position. However, external disturbances, system parameter uncertainty/variation, etc., necessitate a feedback controller to improve the mechanical subsystem performance.

Next, defining a set of shifted variables

$$\hat{x}(t) \triangleq x(t) - x_{\text{ss}}, \tag{2.7}$$

$$\hat{\imath}(t) \triangleq i(t) - i_{\text{ss}}, \tag{2.8}$$

we can rewrite the dynamic equation (2.5), as

$$m\frac{d^2\hat{x}}{dt^2} = mg - K_{\text{f}}\left(\frac{\hat{\imath} + i_{\text{ss}}}{\hat{x} + x_{\text{ss}}}\right)^2. \tag{2.9}$$

Now, linearizing (2.9) about $(\hat{x} = 0, \hat{\imath} = 0)$, yields [3]

$$\frac{d^2\hat{x}}{dt^2} = \frac{1}{m}\left[\frac{\partial}{\partial\hat{x}}\left(mg - K_{\text{f}}\frac{(\hat{\imath} + i_{\text{ss}})^2}{(\hat{x} + x_{\text{ss}})^2}\right)\Bigg|_{(\hat{x}=0,\hat{\imath}=0)}\hat{x} + \frac{\partial}{\partial\hat{\imath}}\left(mg - K_{\text{f}}\frac{(\hat{\imath} + i_{\text{ss}})^2}{(\hat{x} + x_{\text{ss}})^2}\right)\Bigg|_{(\hat{x}=0,\hat{\imath}=0)}\hat{\imath}\right], \tag{2.10}$$

or, equivalently,

$$\frac{d^2\hat{x}}{dt^2} = \frac{2K_{\text{f}}i_{\text{ss}}^2}{x_{\text{ss}}^3 m}\hat{x} - \frac{2K_{\text{f}}i_{\text{ss}}}{x_{\text{ss}}^2 m}\hat{\imath}. \tag{2.11}$$

Finally, taking the Laplace transform of (2.11), we obtain

$$G_m(s) \triangleq \frac{\hat{X}(s)}{\hat{I}(s)} = -\frac{a}{s^2 - b}, \tag{2.12}$$

where $\hat{X}(s) \triangleq \mathcal{L}[\hat{x}(t)]$, $\hat{I}(s) \triangleq \mathcal{L}[\hat{\imath}(t)]$, and

$$a \triangleq \frac{2K_{\text{f}}i_{\text{ss}}}{x_{\text{ss}}^2 m}, \qquad b \triangleq \frac{2K_{\text{f}}i_{\text{ss}}^2}{x_{\text{ss}}^3 m}. \tag{2.13}$$

The numerical values of the electrical and mechanical subsystem parameters for the laboratory maglev model are provided in Table 1. In addition, the variables a and b in (2.13) are computed with $x_{\text{ss}} = 7$mm and $i_{\text{ss}} = 1$ Amp.

3. Objective

i) PI control of the electrical subsystem to track a desired current.

ii) PID control of the mechanical subsystem to track a desired ball position.

Physical quantity	Symbol	Numerical value	Units
Coil inductance	L	0.4125	Henry
Coil resistance	R_ℓ	10	Ohm
Current sensor resistance	R_s	1	Ohm
Force constant	K_f	32654	$\frac{\text{mN-mm}^2}{\text{Amp}^2}$
Gravitational constant	g	9810	$\frac{\text{mm}}{\text{sec}^2}$
Ball mass	m	0.068	Kg

Table 1: Numerical Values for Physical Parameters of The Maglev System

4. Equipment List

i) PC with Q4 data acquisition card and terminal board

ii) Software environment: Windows, MATLAB, Simulink, RTW, and QuaRC

iii) Magnetic levitation apparatus with a steel ball

iv) Universal power module: UPM-2405

v) Set of leads

5. Experimental Procedure

i) Using the set of leads, universal power module, magnetic levitation apparatus, and the terminal board of the Q4 data acquisition card, complete the wiring diagram shown in Figure 3.

ii) Start MATLAB. In the MATLAB window, choose "C:\ControlLab\Experiment4" from the Current Directory window. This directory path choice will change the directory from the default MATLAB directory to the directory where all files needed to perform Experiment 4 are stored.

iii) You can now perform various steps of the magnetic levitation control experiment. However, before proceeding, you **must** request your laboratory teaching assistant to check your electrical connections.

iv) From the **File** menu of the MATLAB window, select the option **Open** to load the Simulink block-diagram "Experiment4_A.mdl." shown in Figure 4 to your desktop. This will load

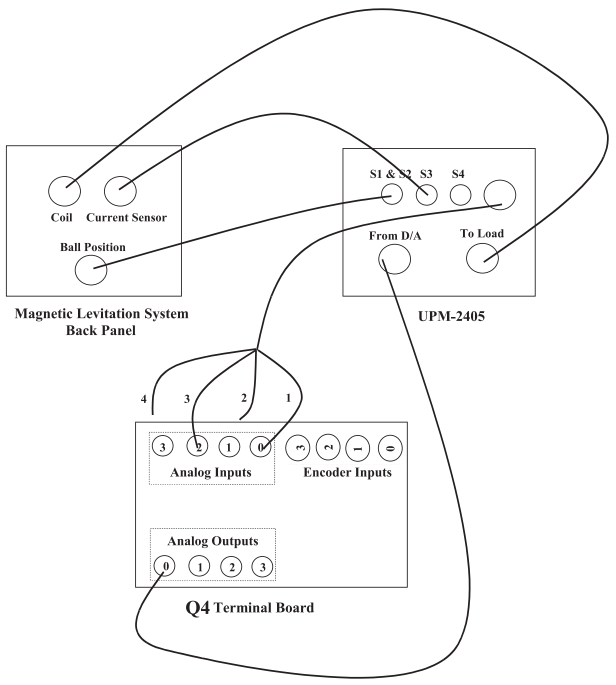

Figure 3: Wiring Diagram for The Magnetic Levitation Experiment

the Simulink diagram file for calibrating the ball sensor voltage when the ball is resting on the black post. The voltage measured on S1 should be about 0 Volts. A digital display window will also appear on your Simulink diagram.

a) In the MATLAB window, click the black **Start** arrow button to acquire the voltage measured on S1 (position sensor).

b) Adjust the offset **potentiometer** on the Maglev to obtain 0 Volts.

c) In the MATLAB window, click the black **Stop** square button when you finish calibrating the sensor off-set.

v) Close the currently opened Simulink diagram. From the **File** menu of the MATLAB window, select the option **Open** to load the Simulink block-diagram "Experiment4_B.mdl" shown in Figure 5 to your desktop. This program applies 1.5 Amperes to the coil which causes the ball to jump up to the magnet and stay there. The voltage measured on S1 should be **between 4.75 and 4.95** Volts.

a) In the MATLAB window, click the black **Start** arrow button to acquire the

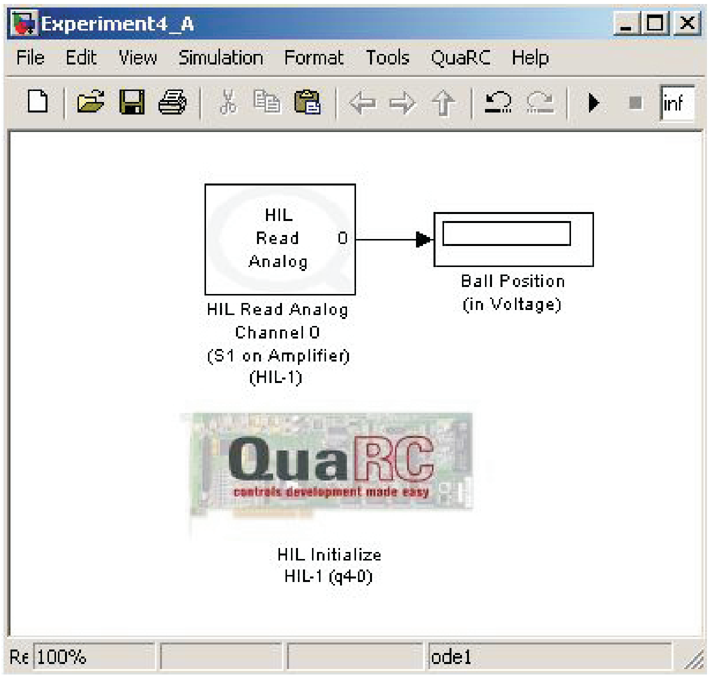
Experiment4_A
File Edit View Simulation Format Tools QuaRC Help
inf
HIL Read Analog 0
Ball Position (in Voltage)
HIL Read Analog Channel 0 (S1 on Amplifier) (HIL-1)
QuaRC controls development made easy
HIL Initialize HIL-1 (q4-0)
Re 100%
ode1

Figure 4: Simulink Block-Diagram for Ball Position Sensor Offset Calibration

voltage measured on S1 (position sensor).

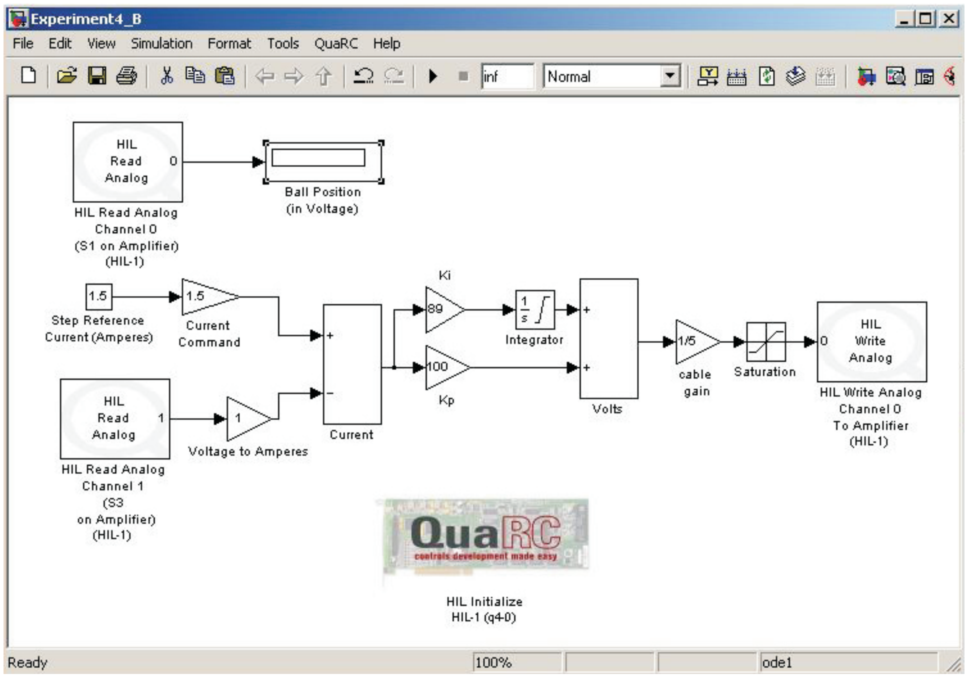
Experiment4_B
File Edit View Simulation Format Tools QuaRC Help
inf
Normal
HIL Read Analog 0
Ball Position (in Voltage)
HIL Read Analog Channel 0 (S1 on Amplifier) (HIL-1)
1.5
Step Reference Current (Amperes)
1.5
Current Command
Ki
89
1/s
Integrator
100
Kp
+
-
Current
+
+
Volts
1/5
cable gain
Saturation
0
HIL Write Analog
HIL Write Analog Channel 0 To Amplifier (HIL-1)
HIL Read Analog 1
1
Voltage to Amperes
HIL Read Analog Channel 1 (S3 on Amplifier) (HIL-1)
QuaRC controls development made easy
HIL Initialize HIL-1 (q4-0)
Ready
100%
ode1

Figure 5: Simulink Block-Diagram for Ball Position Gain Calibration

b) Adjust the gain **potentiometer** on the Maglev to obtain anywhere between **4.75 to 4.95** Volts on the position sensor.

c) In the MATLAB window, click the black **Stop** square button when you finish calibrating the sensor gain.

vi) Close the currently opened Simulink diagram. From the **File** menu of the MATLAB window, select the option **Open** to load the Simulink block-diagram "Experiment4_C.mdl" shown in Figure 6 to your desktop. A plot window will also appear on your desktop. The various Simulink subblocks used in Figure 6 are given in detail in Figures 7–11.

a) In Figure 6, under the subblock labeled **Current Control** (Figure 10), the gains K_p and K_i must be designed and supplied by you. In particular, design a PI feedback controller so that the two poles of the close-loop electrical subsystem are -270 and -0.8 respectively. The feedback diagram of the electrical subsystem with the PI controller is shown in Figure 12, where $A \triangleq R_\ell + R_\mathrm{s}$. The characteristic equation of the closed-loop system in Figure 12 can be used for the purpose of finding K_p and K_i such that the desired poles are achieved.

b) In Figure 6, under the subblock labeled **Mechanical control** (Figure 11), the gains K_p, K_i, and K_d must also be designed and supplied by you. In particular, design a PID feedback controller so that the ball position step response exhibits a peak overshoot $M_\mathrm{p} \leq 5\%$ with settling time $T_\mathrm{s} \leq 0.19$ seconds. The close-loop system is a third order system; hence you must set the third pole to the left of the dominant complex-conjugate pole-pair. The feedback diagram of the mechanical subsystem with the PID controller is shown in Figure 13. The characteristic equation of the closed-loop system in Figure 13 can be used for the purpose of finding K_p, K_i, and K_d such that the desired performance specifications are achieved. Note that in Figure 9, a feedforward controller based on (2.6) is also included to account for the i_ss term in (2.6).

c) Before proceeding, you **must** request your laboratory teaching assistant to approve your gain values. In the MATLAB window, click the black **Start** arrow button to acquire the transient and steady-state position step response of the ball.

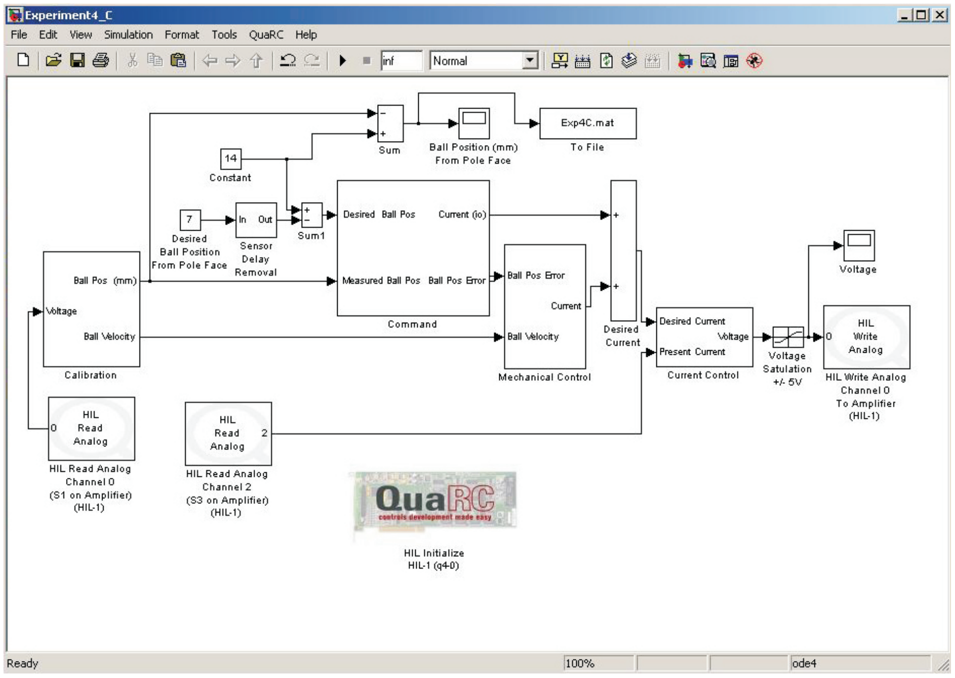

Figure 6: Simulink Block-Diagram for Magnetic Levitation System PID Controller

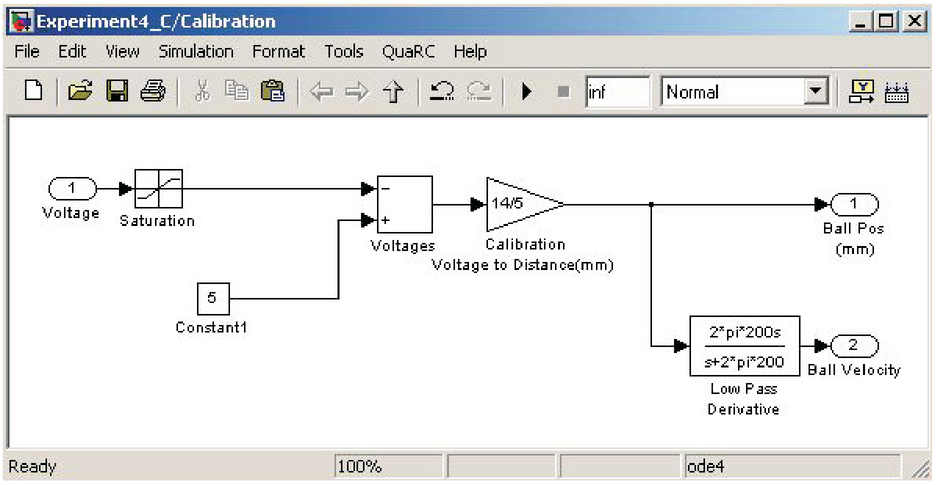

Figure 7: Calibration Subblock

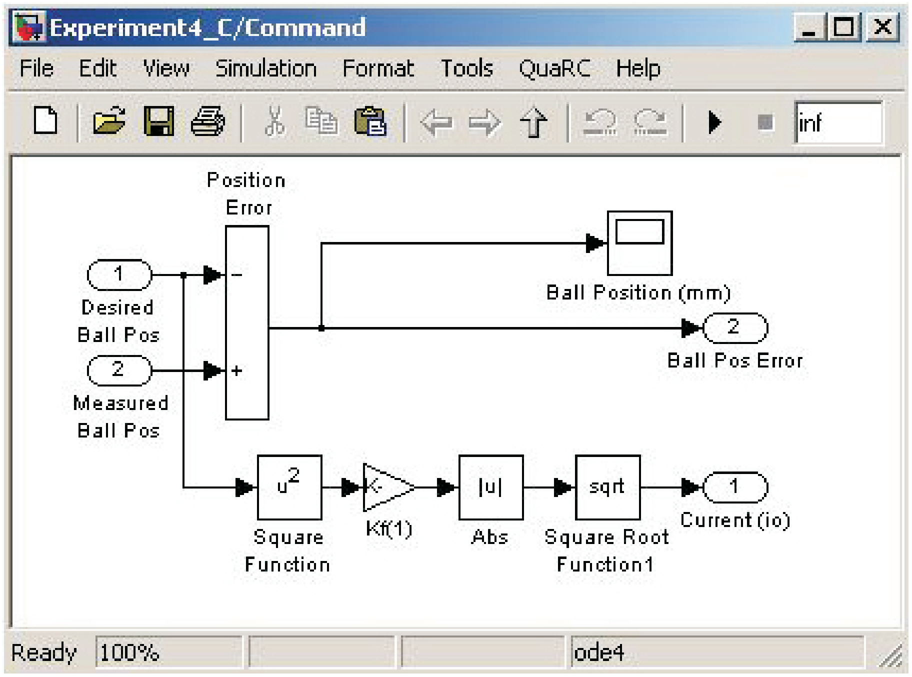

Figure 8: Command Subblock

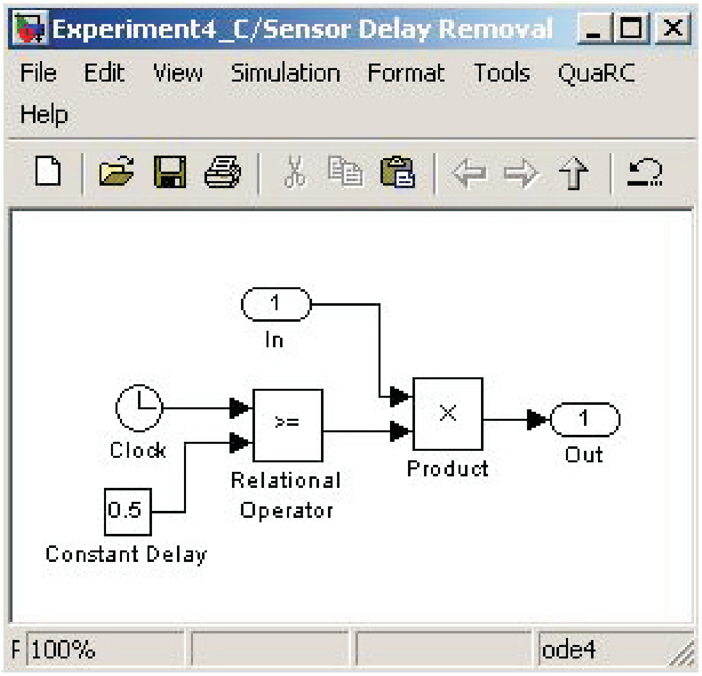
Experiment4_C/Sensor Delay Removal
File Edit View Simulation Format Tools QuaRC Help
1
In
Clock
>=
Relational Operator
0.5
Constant Delay
Product
1
Out
100% ode4

Figure 9: Sensor Delay Removal Subblock

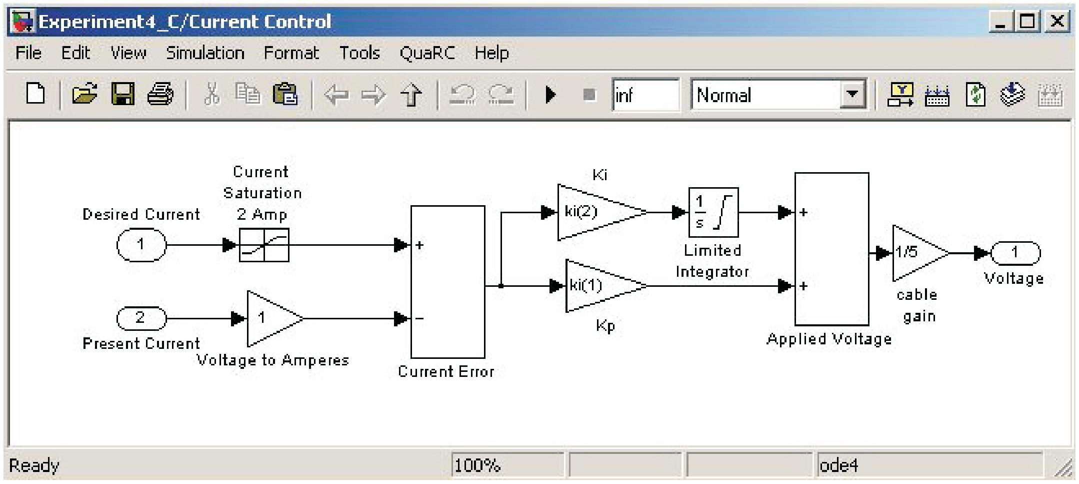

Figure 10: Current Control Subblock

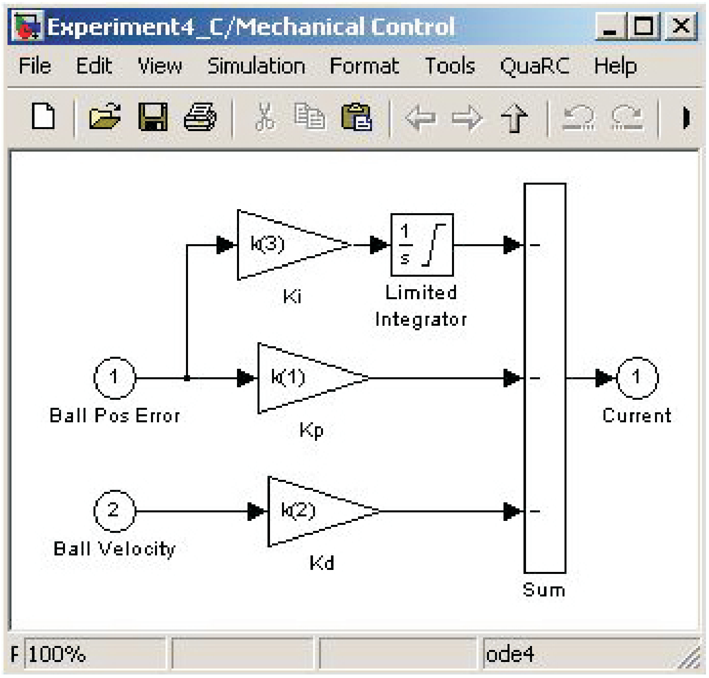

Figure 11: Mechanical Control Subblock

6. Analysis

i) What is the significance of steps *iv*) and *v*) of Section 5 where we adjust the offset and gain potentiometers, respectively, to achieve the desired voltage from the position sensor?

ii) Evaluate the actual overshoot and setting time of the ball position step response and

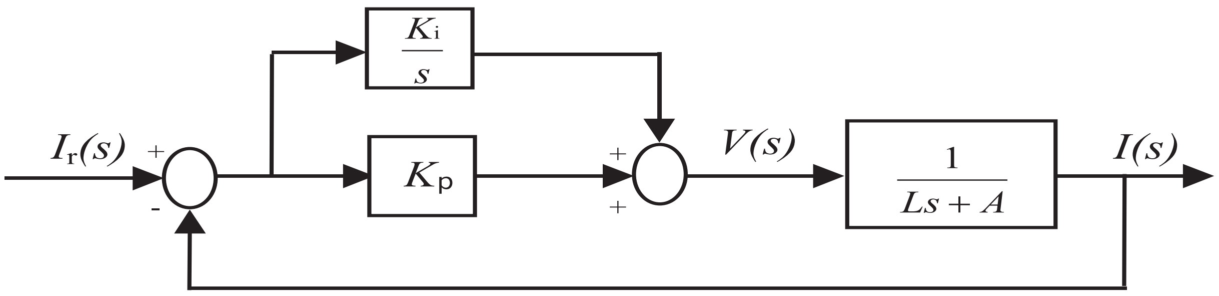

Figure 12: Closed-Loop Feedback Interconnection for PI Control of Electrical Subsystem

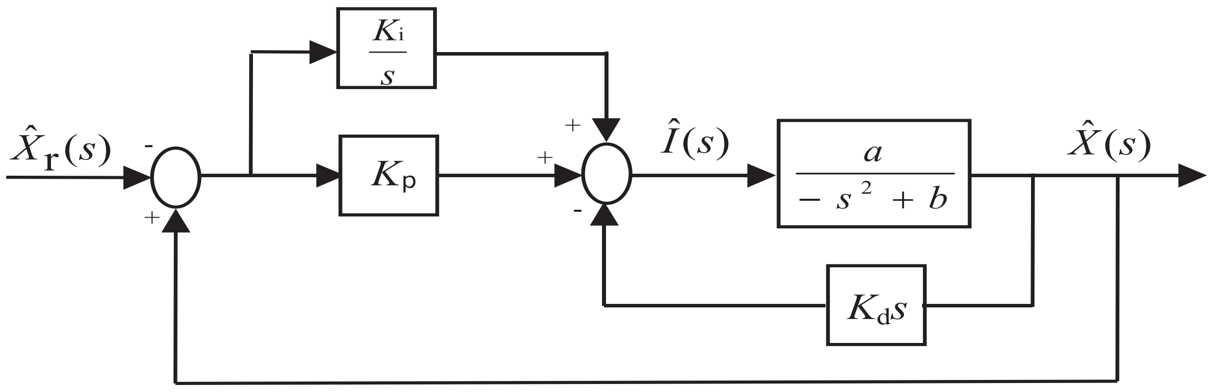

Figure 13: Closed-Loop Feedback Interconnection for PID Control of Mechanical Subsystem

compare with the specified overshoot and setting time. Comment.

iii) How will the electrical subsystem (See Figure 12) respond if gains K_p and K_i are selected to set the two poles of the electrical subsystem at -1 and -0.8?

iv) Can we experimentally set the real root of the closed-loop mechanical subsystem very far from the imaginary axis, in the left-half plane?

References

1. Special Issue on Magnetic Bearing Control, *IEEE Trans. Control System Technology*, 1996., Vol. 4, No. 5.

2. R. C. Dorf and R. H. Bishop, *Modern Control Systems*, Addison Wesley, Menlo Park, CA, 1998, 8th Ed., pp. 108.

3. B. Kuo, *Automatic Control Systems*, Prentice-Hall, Englewood Cliffs, NJ, 1995, 7th Ed., pp. 188–189.

Experiment 5: Modeling and Linear Quadratic Control of a Rotary Inverted Pendulum

Concepts emphasized: Dynamic modeling, linearization, state variables, and LQR design.

1. Introduction

The problem of balancing a broomstick in a vertical upright position on a person's hand (see Figure 1) is well known to the feedback control community [1, 4]. For any human, a physical demonstration of the broomstick-balancing act constitutes a challenging task requiring intelligent, coordinated hand movement based on visual feedback. The instability associated with the equilibrium point ($\alpha = 0$, $\dot{\alpha} = 0$), corresponding to the broomstick vertical upright position, leads to the challenge inherent in the problem.

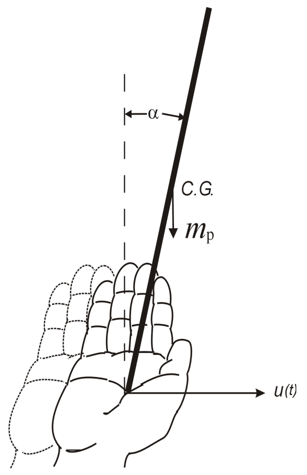

Figure 1: The Broomstick Balancing Problem [1]

A one-dimensional (1–D) electro-mechanical analogue of the broomstick-balancing problem is the classical inverted-pendulum-on-cart (IPC) problem [1] (see Figure 2). In the IPC problem, the cart is moved rectilinearly to keep the pendulum vertical upright. The IPC problem is intimately related to the problem of balancing a missile immediately after launch [1, 4].

The dynamics of IPC are inherently nonlinear. In addition, similar to the broomstick-balancing problem, the equilibrium point ($\alpha = 0$, $\dot{\alpha} = 0$) for the inverted pendulum is unstable. The feedback control design problem for IPC has been extensively studied and a variety of control designs have

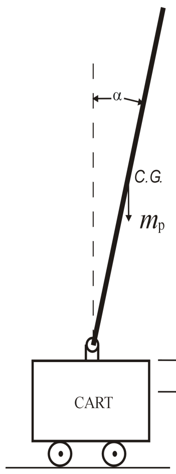

Figure 2: Inverted Pendulum on Cart Problem

been proposed in the literature for this interesting problem. In this laboratory exercise, we will consider a variant of the IPC problem, *viz.*, the 1–D rotary inverted pendulum (RIP) problem.

The laboratory RIP-model consists of a rigid link (pendulum) rotating in the vertical plane. The rigid link is attached to a pivot arm which is mounted on the load shaft of the SRV-02 DC-motor. The pivot arm can be rotated in the horizontal plane by the SRV-02 DC-motor. The SRV-02 DC-motor is instrumented with an encoder and a tachometer. In addition, an encoder is mounted on the pivot arm to measure the pendulum angle. The principal objective of this experiment is to balance the pendulum in the vertical upright position and to position the pivot arm. Since the plant has two degrees of freedom but only one actuator, the system is underactuated and exhibits significant nonlinear behavior for large pendulum excursion.

In this laboratory exercise, we will develop the governing differential equations of motion for the RIP-system using Lagrange's method [3]. Next, we will linearize the nonlinear RIP-model dynamics in the neighborhood of interest and develop a state-space model for the system. In addition, we will briefly outline the Linear Quadratic Regulator (LQR) design methodology [5]. For step command tracking, we will unify the integral control scheme with the LQR control design technique [2]. Finally, we will design, implement, and evaluate the performance of an LQ tracking control law on the laboratory RIP test-bed.

2. Background

Mechanical system modeling: A schematic representation of the rotary inverted pendulum is given in Figure 3 where l_p denotes the pendulum helf-length and m_p denotes the pendulum mass. Assume that the pendulum rod is rigid and massless. Let α be the angle of the rod from the vertical axis z. The pivot arm OA has length r. The total effective mass moment of inertia reflected at the output shaft of the SRV-02 DC-motor apparatus is called the base mass moment of inertia and is denoted by J_b. Note that J_b includes moment of inertia of DC-motor, tachometer, various gears, and pivot arm; all reflected at the center of rotation O. The SRV-02 DC-motor applies a torque τ on the pivot arm OA.

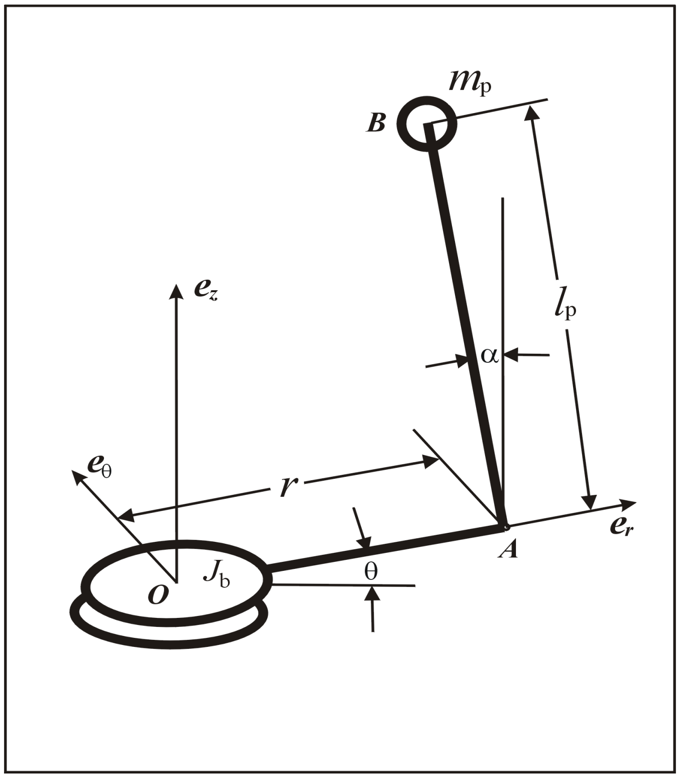

Figure 3: Simplified Model of Rotary Inverted Pendulum

Next, note that the position vector OB in the cylindrical coordinate frame e_r–e_θ–e_z is given by

$$\overrightarrow{OB} = r\mathrm{e}_r + l_\mathrm{p}\sin\alpha\,\mathrm{e}_\theta + l_\mathrm{p}\cos\alpha\,\mathrm{e}_z. \tag{2.1}$$

Furthermore, note that the coordinate frame e_r–e_θ–e_z has angular velocity $\dot{\theta}\mathrm{e}_z$. Next, computing

$\frac{\mathrm{d}}{\mathrm{d}t}\overrightarrow{OB}$, the velocity of mass m_p (or point B) is given by

$$\overrightarrow{v} \;=\; -l_\mathrm{p}\dot{\theta}\sin\alpha\,\mathrm{e}_r + (r\dot{\theta} + l_\mathrm{p}\dot{\alpha}\cos\alpha)\mathrm{e}_\theta - l_\mathrm{p}\dot{\alpha}\sin\alpha\,\mathrm{e}_z. \tag{2.2}$$

Now, it follows from (2.2) and the notation $v \overset{\triangle}{=} |\overrightarrow{v}|$ that

$$v^2 \;=\; (l_\mathrm{p}\dot{\theta}\sin\alpha)^2 + (r\dot{\theta} + l_\mathrm{p}\dot{\alpha}\cos\alpha)^2 + (l_\mathrm{p}\dot{\alpha}\sin\alpha)^2. \tag{2.3}$$

Next, note that the total kinetic energy of the RIP-system is the sum of kinetic energies of the pendulum mass m_p and the base inertia J_b, which are given by

$$T_\mathrm{p} \;=\; \frac{1}{2}m_\mathrm{p}v^2, \tag{2.4}$$

and

$$T_\mathrm{b} \;=\; \frac{1}{2}J_\mathrm{b}\dot{\theta}^2, \tag{2.5}$$

respectively. Thus, the total kinetic energy of the RIP-system is

$$T \;=\; \frac{1}{2}(m_\mathrm{p}v^2 + J_\mathrm{b}\dot{\theta}^2). \tag{2.6}$$

Furthermore, the potential energy of the RIP-system is given by

$$U \;=\; m_\mathrm{p}gl_\mathrm{p}\cos\alpha, \tag{2.7}$$

where g is the gravitational acceleration. Finally, note that computing the work done by the external torque τ (applied by the SRV-02 DC-motor)

$$\delta W \;=\; \tau\delta\theta,$$

the generalized forces are identified to be

$$Q_\theta \;=\; \tau, \qquad Q_\alpha = 0. \tag{2.8}$$

Now, we use Lagrange's equations [3] for θ and α coordinates given by

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial T}{\partial\dot{\theta}}\right) - \frac{\partial T}{\partial\theta} + \frac{\partial U}{\partial\theta} \;=\; Q_\theta, \tag{2.9}$$

and

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial T}{\partial\dot{\alpha}}\right) - \frac{\partial T}{\partial\alpha} + \frac{\partial U}{\partial\alpha} \;=\; Q_\alpha, \tag{2.10}$$

to obtain

$$(m_\mathrm{p}r^2 + J_\mathrm{b} + m_\mathrm{p}l_\mathrm{p}^2 \sin^2\alpha)\ddot{\theta} + (m_\mathrm{p}rl_\mathrm{p}\cos\alpha)\ddot{\alpha} - (m_\mathrm{p}rl_\mathrm{p}\sin\alpha)\dot{\alpha}^2 + (2m_\mathrm{p}l_\mathrm{p}^2\sin\alpha\,\cos\alpha)\dot{\alpha}\,\dot{\theta} \;=\; \tau, \tag{2.11}$$

and

$$m_\mathrm{p}l_\mathrm{p}^2\ddot{\alpha} + (m_\mathrm{p}rl_\mathrm{p}\cos\alpha)\ddot{\theta} - (m_\mathrm{p}l_\mathrm{p}^2\sin\alpha\,\cos\alpha)\dot{\theta}^2 - m_\mathrm{p}gl_\mathrm{p}\sin\alpha \;=\; 0, \tag{2.12}$$

respectively.

Next, we simplify the dynamic model (2.11), (2.12) by linearizing it in the vicinity of the equilibrium point $(\alpha = 0, \dot{\alpha} = 0)$, which corresponds to the pendulum maintaining a vertical upright position. Thus, in (2.11), (2.12), we replace $\sin\alpha \approx \alpha$ and $\cos\alpha \approx 1$ and neglect the higher-order terms in the variables α, $\dot{\alpha}$, etc. This leads to the linearized RIP-system dynamics

$$(m_\mathrm{p}r^2 + J_\mathrm{b})\ddot{\theta} + m_\mathrm{p}rl_\mathrm{p}\ddot{\alpha} \;=\; \tau, \tag{2.13}$$

$$m_\mathrm{p}l_\mathrm{p}^2\ddot{\alpha} + m_\mathrm{p}rl_\mathrm{p}\ddot{\theta} - m_\mathrm{p}gl_\mathrm{p}\alpha \;=\; 0. \tag{2.14}$$

After simple algebraic manipulation of (2.13), (2.14), we obtain the following linear, state-space representation [5] of the RIP-system

$$\begin{bmatrix} \dot{\theta} \\ \dot{\alpha} \\ \ddot{\theta} \\ \ddot{\alpha} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & -\frac{m_\mathrm{p}rg}{J_\mathrm{b}} & 0 & 0 \\ 0 & \frac{J_\mathrm{b}+m_\mathrm{p}r^2}{l_\mathrm{p}J_\mathrm{b}}g & 0 & 0 \end{bmatrix} \begin{bmatrix} \theta \\ \alpha \\ \dot{\theta} \\ \dot{\alpha} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \frac{1}{J_\mathrm{b}} \\ -\frac{r}{l_\mathrm{p}J_\mathrm{b}} \end{bmatrix} \tau. \tag{2.15}$$

DC-motor dynamics: Recall from Experiment 3 that, neglecting the armature inductance L_a, equations (2.2) and (2.3) of Experiment 3 yield

$$\begin{aligned} V_\mathrm{a} \;&=\; i_\mathrm{a}R_\mathrm{a} + K_\mathrm{b}\omega_\mathrm{m} \\ &=\; i_\mathrm{a}R_\mathrm{a} + K_\mathrm{b}K_\mathrm{g}\omega_\ell, \end{aligned} \tag{2.16}$$

since $\omega_\mathrm{m} = K_\mathrm{g}\omega_\ell$ where $\omega_\ell \triangleq \dot{\theta}$ is the load (i.e., the pivot arm) angular velocity. Now, it follows from (2.16) that

$$i_\mathrm{a} = \frac{V_\mathrm{a}}{R_\mathrm{a}} - \frac{K_\mathrm{b}K_\mathrm{g}}{R_\mathrm{a}}\omega_\ell. \tag{2.17}$$

Next, using $\tau = K_\mathrm{g}T_\mathrm{m}$ and equation (2.1) of Experiment 3, it follows that

$$\tau \;=\; K_\mathrm{g}K_\mathrm{T}i_\mathrm{a}. \tag{2.18}$$

Hence, using (2.17) in (2.18), and noting that in the SI-units the numerical values of $K_{\rm T}$ and $K_{\rm b}$ are identical [4], we obtain

$$\tau = \frac{K_{\rm b}K_{\rm g}}{R_{\rm a}}V_{\rm a} - \frac{K_{\rm b}^2K_{\rm g}^2}{R_{\rm a}}\omega_\ell. \qquad (2.19)$$

The numerical values of the mechanical and electrical subsystem parameters for the laboratory RIP-model are provided in Table 1 below.

Physical quantity	Symbol	Numerical value	Units
Pivot arm length	r	8.75×0.0254	meter
Base mass moment of inertia	$J_{\rm b}$	0.005	Kg-meter^2
Pendulum length	$2l_{\rm p}$	13.125×0.0254	meter
Pendulum mass	$m_{\rm p}$	0.126	Kg
Gravitational constant	g	9.8	meter/sec^2
DC-motor armature resistance	$R_{\rm a}$	2.6	Ohm
Motor constant	$K_{\rm T}, K_{\rm b}$	0.00767	N-m/Amp, Volt-sec/rad
Gear ratio	$K_{\rm g}$	14×5	

Table 1: Numerical Values for Physical Parameters of The RIP-System

Finally, substituting (2.19) into (2.15), rearranging terms, and using the numerical parameter values given in Table 1, we obtain the RIP-system model given by

$$\begin{bmatrix} \dot\theta \\ \dot\alpha \\ \ddot\theta \\ \ddot\alpha \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & -55.0710 & -22.2484 & 0 \\ 0 & 132.2206 & 29.6645 & 0 \end{bmatrix} \begin{bmatrix} \theta \\ \alpha \\ \dot\theta \\ \dot\alpha \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 41.4385 \\ -55.2514 \end{bmatrix} V_{\rm a}. \qquad (2.20)$$

3. LQR-Based Controller Design

Linear quadratic regulator theory: The LQR theory is a powerful method for the control of linear systems in the state-space domain. The LQR technique generates controllers with guaranteed closed-loop stability robustness property even in the face of certain gain and phase variation at the plant input/output. In addition, the LQR-based controllers provide reliable closed-loop system performance despite the presence of stochastic plant disturbance. The LQ control design framework is applicable to the class of stabilizable linear systems.

Next, we briefly summarize the LQR theory. Given an $n^{\rm th}$-order stabilizable system

$$\dot x(t) = Ax(t) + Bu(t), \qquad t \geq 0, \qquad x(0) = x_0, \qquad (3.1)$$

where $x(t) \in \mathbb{R}^n$ is the state vector and $u(t) \in \mathbb{R}^m$ is the input vector, determine the matrix gain $K \in \mathbb{R}^{m\times n}$ such that the static, full-state feedback control law

$$u(t) \;=\; -Kx(t), \tag{3.2}$$

satisfies the following criteria:

i) the closed-loop system (3.1) and (3.2) is asymptotically stable and

ii) the quadratic performance functional

$$J(K) \;\triangleq\; \int_0^\infty [x^T(t)R_1x(t) + u^T(t)R_2u(t)]dt, \tag{3.3}$$

where R_1 is a nonnegative-definite matrix that penalizes the departure of system states from the equilibrium and R_2 is a positive-definite matrix that penalizes the control input, is minimized.

The solution of the LQR problem can be obtained via a Lagrange multiplier-based optimization technique and is given by

$$K \;=\; {R_2}^{-1}B^TP, \tag{3.4}$$

where $P \in \mathbb{R}^{n\times n}$ is a nonnegative-definite matrix satisfying the matrix Riccati equation

$$0 \;=\; A^TP + PA + R_1 - PB{R_2}^{-1}B^TP. \tag{3.5}$$

Note that it follows from (3.2) that the LQR-based control design requires the availability of all state variables for feedback purpose. The state variables for the laboratory RIP-system model are identified from (2.20) to be θ, α, $\dot{\theta}$ and $\dot{\alpha}$. For our laboratory RIP-model, the pivot arm angle θ and angular velocity $\dot{\theta}$ are measured by an encoder and a tachometer, respectively. The pendulum angular position α is measured by another encoder. The pendulum angular velocity $\dot{\alpha}$ is not measured by any physical sensor, instead, we numerically compute $\dot{\alpha}$ by implementing a low-pass differentiator, e.g. $\frac{100s}{s+100}$, as part of the overall control scheme.

In order to design an LQR controller for the RIP-system, we identify the plant dynamics A and input matrix B from (2.20). In addition, we choose the weighting matrices R_1 and R_2 to penalize the state and control variables, respectively, as

$$R_1 = \begin{bmatrix} 0.25 & 0 & 0 & 0 \\ 0 & 4 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \qquad R_2 = 0.2. \tag{3.6}$$

Note that in (3.6), R_1 places a significantly higher penalty on the pendulum angle α excursions ($R_1(2,2) = 4$) than the pivot arm angle θ excursions ($R_1(1,1) = 0.25$). In addition, the pendulum angular velocity $\dot{\alpha}$ is penalized ($R_1(4,4) = 1$) whereas the pivot arm angular velocity $\dot{\theta}$ is not penalized at all ($R_1(3,3) = 0$). This R_1 prevents large departure of pendulum angle α from the equilibrium and the tendency of the pendulum to fall down; thus maintaining the pendulum equilibrium ($\alpha = 0, \dot{\alpha} = 0$). The control penalty R_2 given in (3.6) is determined by trial and error. A larger value for R_2 will lead to smaller control effort and larger excursions of θ and α whereas a smaller value of R_2 will lead to larger control effort which may saturate the actuator.

Next, an LQR controller for the given data is designed by using the MATLAB command **lqr** to solve the matrix Riccati equation (3.5) and to compute the controller gain (3.4). In particular, executing $K = \mathrm{lqr}(A, B, R_1, R_2)$, in the MATLAB command window (with the input variables in the MATLAB memory) we obtain the feedback regulator gain

$$\begin{aligned} K &= \begin{bmatrix} -1.1180 & -19.7995 & -1.6190 & -3.2299 \end{bmatrix} \text{ V/rad}, \\ &= \begin{bmatrix} -0.0195 & -0.3460 & -0.0283 & -0.0565 \end{bmatrix} \text{ V/deg}. \end{aligned} \tag{3.7}$$

Finally, the controlled voltage to be applied to the SRV-02 DC-motor is given by

$$V_{\mathrm{a}}(t) = -0.0195\theta(t) - 0.3460\alpha(t) - 0.0283\dot{\theta}(t) - 0.0565\dot{\alpha}(t). \tag{3.8}$$

LQR-based tracking controller design: The LQR-based control law (3.8) renders the origin of the RIP-system asymptotically stable, i.e., $\lim\limits_{t\to\infty} x(t) \to 0$ with (3.8). Thus, although the above controller maintains the pendulum in the vertical upright position, it does not allow one to position the pivot arm arbitrarily. An LQR-based controller can be designed to position the pivot arm at a nonzero angle by shifting the origin of the state variable θ to the desired set point of the pivot arm angular position. However, external disturbances and nonlinear effects, e.g., gravity, gear backlash, etc., present in the RIP-model may lead to poor closed-loop system performance for LQR-based set point controllers. Next, recall from the classical control theory that the integral control action yields zero steady-state error for constant command input tracking even in the presence of exogenous disturbances. Thus, in this experiment, to design a tracking controller which positions the pivot arm as well as maintains the pendulum vertical upright, we unify the integral control scheme with the LQR design methodology.

Hence, consider the feedback diagram shown in Figure 4 where the integrator $\frac{1}{s}$ has been introduced to enable the pivot arm to track constant command angle. Note, that the integral state $x_{\rm I}$ satisfies

$$\dot{x}_{\rm I} = e, \tag{3.9}$$

where

$$e \overset{\triangle}{=} \theta - r. \tag{3.10}$$

Note that in (3.10) r is the desired, constant pivot arm angular position. In addition, it follows from (3.10) that

$$\dot{e} = \dot{\theta}. \tag{3.11}$$

Now, replacing the $\dot{\theta}$ equation in the vector differential equation (2.20) by (3.11) and augmenting the resulting vector differential equation with (3.9), we obtain the state-space model

$$\begin{bmatrix} \dot{e} \\ \dot{\alpha} \\ \ddot{\theta} \\ \ddot{\alpha} \\ \dot{x}_{\rm I} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & -55.0710 & -22.2484 & 0 & 0 \\ 0 & 132.2206 & 29.6645 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} e \\ \alpha \\ \dot{\theta} \\ \dot{\alpha} \\ x_{\rm I} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 41.4385 \\ -55.2514 \\ 0 \end{bmatrix} V_{\rm a}. \tag{3.12}$$

Next, we select the following design variables for the LQR-based tracker design

$$R_{1\rm a} = \begin{bmatrix} 0.25 & 0 & 0 & 0 & 0 \\ 0 & 4 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.04 \end{bmatrix}, \qquad R_{2\rm a} = 0.05. \tag{3.13}$$

Now, using the augmented system dynamic matrix $A_{\rm a}$ and input matrix $B_{\rm a}$ from (3.12) and the weighting matrices $R_{1\rm a}$ and $R_{2\rm a}$ from (3.13), we can design the augmented feedback control gain $K_{\rm a}$ by executing the MATLAB command $K_{\rm a} = \text{lqr}(A_{\rm a}, B_{\rm a}, R_{1\rm a}, R_{2\rm a})$. Finally, the feedback gains K and $K_{\rm I}$ implemented in Figure 4 are obtained from $\begin{bmatrix} K & K_{\rm I} \end{bmatrix} = K_{\rm a}$.

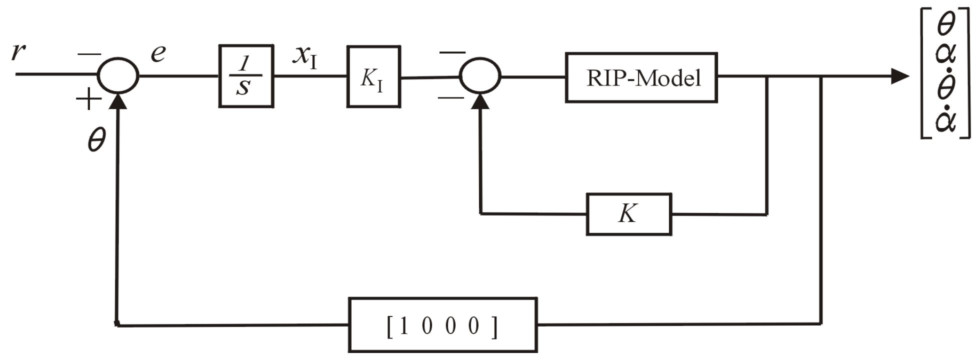
r
−
+
e
θ
1/s
x_I
K_I
−
−
RIP-Model
K
[1 0 0 0]
$\begin{bmatrix}\theta\\ \alpha\\ \dot\theta\\ \dot\alpha\end{bmatrix}$

Figure 4: Integral Control of Rotary Inverted Pendulum

4. Objective

i) Develop and linearize the governing differential equations of the RIP-system.

ii) Design and implement a stabilizing LQR-based controller for the laboratory RIP-model.

iii) Design and implement an LQR-based tracker for the laboratory RIP-model.

5. Equipment List

i) PC with Q4 data acquisition card and terminal board

ii) Software environment: Windows, MATLAB, Simulink, RTW, and QuaRC

iii) SRV-02 geared DC-motor apparatus with an optical encoder and tachometer; pivot arm attachment with an encoder; and pendulum

iv) Universal power module: UPM-1503

v) Set of leads

6. Experimental Procedure

i) Using the set of leads, universal power module UPM 1503, SRV-02 DC-motor apparatus, and the terminal board of the Q4 data acquisition card, complete the wiring diagram shown in Figure 5.

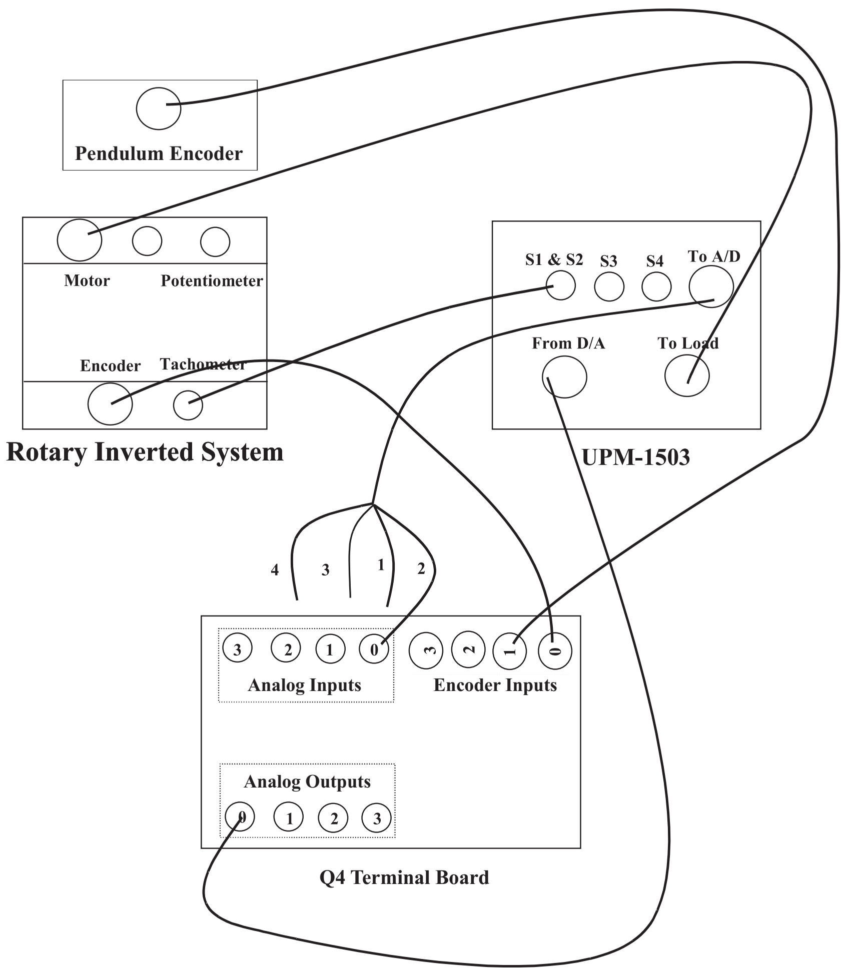

Figure 5: the Wiring Diagram for Rotary inverted Pendulum

ii) Start MATLAB. In the MATLAB window, choose "C:\ControlLab\Experiment5" from the Current Directory window. This directory path choice will change the directory from the default MATLAB directory to the directory where all files needed to perform Experiment 5 are stored.

iii) You can now perform various steps of the rotary inverted pendulum experiment. However, before proceeding, you **must** request your laboratory teaching assistant to check your electrical connections.

iv) At the MATLAB command prompt, execute the script **Experiment5a.m**. This will assign the numerical values of the physical parameters of the RIP-model, compute the linear state-space model (2.20), assign the penalty matrices R_1 and R_2 of (3.6), and solve the LQR problem to generate the controller gain (3.7). Note that for this part of the experiment the integral control gain K_{I} is set to zero.

v) From the **File** menu of the MATLAB window, select the option **Open** to load the Simulink block-diagram "Experiment5.mdl." as shown in Figure 6 to your desktop. This will load the model file for conducting the RIP stabilization experiment.

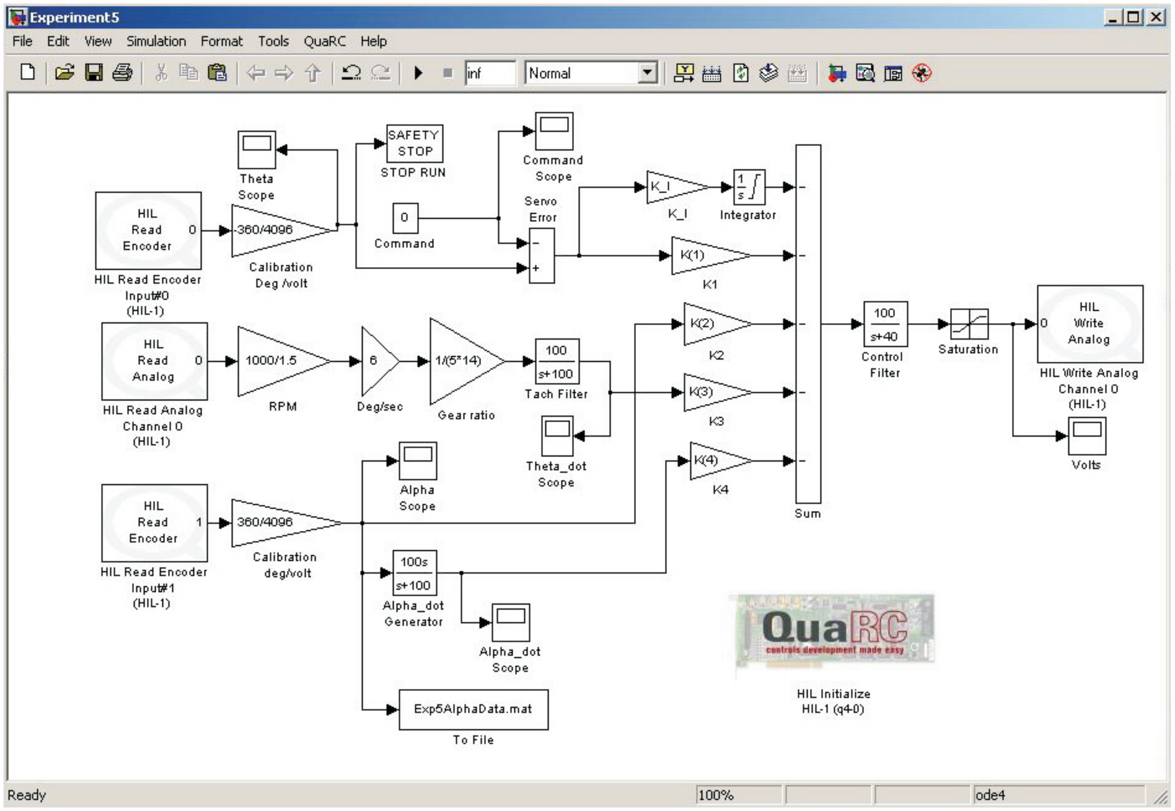

Figure 6: Simulink Block-Diagram for LQR-Based RIP Stabilization

a) In order to properly start the controller, move the pivot arm to the zero position and hold the pendulum in the vertical upright position when you click the black **Start** arrow button.

b) Record the time response of the pendulum angle **alpha** and the pivot arm angular position **theta**.

c) Apply a slight tap to the pendulum so that it falls around 2.5 degrees which causes the arm to move toward the falling direction. Record the time response of pendulum angle α and pivot arm angular position θ.

d) Note that the LQR controller (3.7) implemented in this part of the experiment does not include the integral control action. To evaluate the pivot arm tracking performance of controller (3.8), change the value of the Constant block in the Simulink block-diagram of Figure 6 and record the time response of pendulum angle α and pivot arm angular position θ.

vi) To design the LQR-based tracker outlined in Section 3, execute the script **Experiment5b.m** at the MATLAB command prompt. This will produce the augmented system dynamics and input matrices, the augmented state and control penalty matrices, and the controller gain K_a along with its partitions K and K_I. Next, repeat steps *v.a*)–*v.d*) given above.

7. Analysis/Assignment

i) In Section 2, we developed the nonlinear RIP-system dynamic model (2.11) and (2.12) by approximating the pendulum as a point mass concentrated at the pendulum center of gravity. In addition, in Section 2, the point mass m_p is assumed to be attached to a massless rigid bar of length l_p. The preceding assumptions essentially ignore the rotational effects of pendulum mass moment of inertia on system dynamics. Develop a complete nonlinear RIP-system model which accounts for the rotational effects of the pendulum mass moment of inertia.

ii) Analyze the scripts “Experiment5a.m” and “Experiment5b.m.”

iii) Analyze and comment on your experimental results. In addition, contrast the performance of the stabilizing control law with the tracking control law.

iv) Analyze the effects of the Control Filter and Tach Filter blocks in the Simulink block-diagram of Figure 6.

v) Identify the nonlinearities present in the laboratory RIP-system and discuss their effects on the system response.

References

1. R. C. Dorf and R. H. Bishop, *Modern Control Systems*, Addison Wesley, Menlo Park, CA, 1998, 8th Ed., pp. 136.

2. G. F. Franklin, J. D. Powell, and A. Emami-Naeini, *Feedback Control of Dynamic Systems*, Addison Wesley, Menlo Park, CA, 1994, 3rd Ed., pp. 552.

3. D. T. Greenwood, *Principles of Dynamics*, Prentice-Hall, Englewood Cliffs, NJ, 1988, 2nd Ed., pp. 265–268.

4. B. Kuo, *Automatic Control Systems*, Prentice-Hall, Englewood Cliffs, NJ, 1995, 7th Ed., pp. 179 and pp. 221.

5. K. Ogata, *Modern Control Engineering*, Prentice-Hall, Upper Saddle River, NJ, 1997, 3rd Ed., pp. 70–72 and pp. 920–923.

Experiment 6: Level Control of a Coupled Water Tank

Concepts emphasized: Dynamic modeling, time-domain analysis, and proportional-plus-integral control.

1. Introduction

Industrial applications of liquid level control abound, e.g., in food processing, beverage, dairy, filtration, effluent treatment, and nuclear power generation plants; pharmaceutical industries; water purification systems; industrial chemical processing and spray coating; boilers; and automatic liquid dispensing and replenishment devices. The typical actuators used in liquid level control systems include pumps, motorized valves, on-off valves, etc. In addition, level sensors such as displacement float, capacitance probe, pressure sensor [1], etc. provide liquid level measurement for feedback control purpose. In this laboratory exercise, the students model, calibrate, and control a two-tank level control system. In particular, this experiment exposes the students to the fundamental modeling principle of fluid mass balance, pressure sensor calibration, and a feedback control design methodology for a state-coupled, two-tank level control system.

2. Background

System Modeling: The schematic drawing in Figure 1 represents the model of a two degree-of-freedom (DOF) state-coupled, water tank system. This system consists of two tanks with orifices and level sensors at the bottom of each tank, a pump, and a water basin. The two tanks have same diameters and can be fitted with different diameter outflow orifices. In this laboratory setup, the pump provides infeed to Tank 1 and the outflow of Tank 1 becomes infeed to Tank 2. The outflow of Tank 2 is emptied into the water basin. The following conditions with regard to the system dynamic model are used to describe the level of water in Tanks 1 and 2.

i) The water levels in Tanks 1 and 2 are measured by two pressure sensors;

ii) the level of water in Tank 1 is always less than 30cm;

iii) the desired level of water in Tank 2 is always greater than 0cm and less than 20cm;

iv) the voltage applied at the input terminals of the pump is between 0 and 22 Volts.

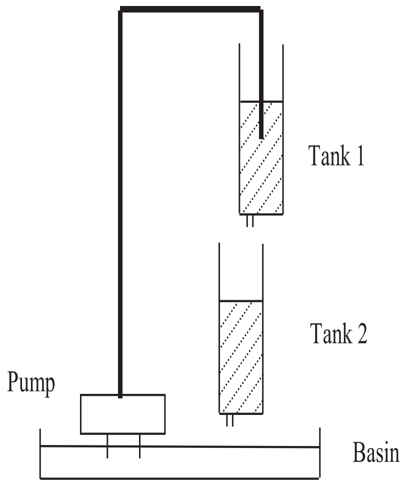

Figure 1: State-Coupled Two-Tank System

Based on the above assumptions, the dynamic equations for the liquid level in the two tanks are derived as follows. Note that for each tank the time rate of change of liquid level is given by

$$\dot{L}_i(t) \;=\; \frac{1}{A_i}\left(F_i^{\mathrm{in}}(t) - F_i^{\mathrm{out}}(t)\right) \quad \frac{\mathrm{cm}}{\mathrm{sec}}, \quad i = 1, 2, \tag{2.1}$$

where L_i, A_i, F_i^{in}, and F_i^{out} are the liquid level, cross-sectional area, inflow rate, and outflow rate, respectively, for the i^{th} tank. Next, note that the inflow rate to Tank 1 is given by

$$F_1^{\mathrm{in}}(t) \;=\; K_{\mathrm{p}} V_{\mathrm{p}} \quad \frac{\mathrm{cm}^3}{\mathrm{sec}}, \tag{2.2}$$

where K_{p} is the pump constant $(\frac{\mathrm{cm}^3}{\text{Volts-sec}})$ and V_{p} is the voltage applied to the pump. In addition, using Bernoulli's law for flow through small orifices, the outflow velocity from the orifice at the bottom of each tank is

$$v_i^{\mathrm{out}}(t) \;=\; \sqrt{2gL_i} \quad \frac{\mathrm{cm}}{\mathrm{sec}}, \quad i = 1, 2. \tag{2.3}$$

Then, the outflow rate for each tank is given by

$$F_i^{\mathrm{out}}(t) \;=\; a_i\sqrt{2gL_i} \quad \frac{\mathrm{cm}^3}{\mathrm{sec}}, \quad i = 1, 2, \tag{2.4}$$

where g is the gravitational acceleration and a_i denotes the cross-sectional area of the outflow orifice at the bottom of the i^{th} tank. Finally, note that for the two-tank level control system shown in Figure 1

$$F_2^{\mathrm{in}}(t) \;=\; F_1^{\mathrm{out}}(t). \tag{2.5}$$

Thus, using (2.1)–(2.5), we obtain the dynamic equations for the liquid level in the two tanks as

$$\dot{L}_1(t) = -\frac{a_1}{A_1}\sqrt{2gL_1(t)} + \frac{K_\mathrm{p}}{A_1}V_\mathrm{p}(t), \tag{2.6}$$

$$\dot{L}_2(t) = \frac{a_1}{A_2}\sqrt{2gL_1(t)} - \frac{a_2}{A_2}\sqrt{2gL_2(t)}. \tag{2.7}$$

Remark 2.1. Note that using (2.6), we can compute the steady-state pump voltage $V_{\mathrm{p}_\mathrm{ss}}$ that produces the desired steady-state constant level L_{1_ss} in Tank 1. Specifically, setting $\dot{L}_1(t) = 0$ in (2.6) yields

$$V_{\mathrm{p}_\mathrm{ss}} = a_1\frac{\sqrt{2gL_{1_\mathrm{ss}}}}{K_\mathrm{p}}. \tag{2.8}$$

In a similar manner, we can compute the steady-state level L_{1_ss} in Tank 1 that produces the desired steady-state constant level L_{2_ss} in Tank 2. Specifically, setting $\dot{L}_2(t) = 0$ in (2.7) yields

$$L_{1_\mathrm{ss}} = \left(\frac{a_2}{a_1}\right)^2 L_{2_\mathrm{ss}}. \tag{2.9}$$

Now, theoretically one can use (2.8) and (2.9) to regulate the water level in Tank 2. However, external disturbances, system parameter uncertainty/variation, etc., necessitate a feedback controller to improve the level control system performance.

Next, defining a set of shifted variables

$$\ell_1(t) \triangleq L_1(t) - L_{1_\mathrm{ss}}, \tag{2.10}$$

$$\ell_2(t) \triangleq L_2(t) - L_{2_\mathrm{ss}}, \tag{2.11}$$

$$u(t) = V_\mathrm{p}(t) - V_{\mathrm{p}_\mathrm{ss}}, \tag{2.12}$$

we can rewrite the dynamic equations (2.6) and (2.7) as

$$\dot{\ell}_1(t) = -\frac{a_1}{A_1}\sqrt{2g(\ell_1(t) + L_{1_\mathrm{ss}})} + \frac{K_\mathrm{p}}{A_1}\left(u(t) + V_{\mathrm{p}_\mathrm{ss}}\right), \tag{2.13}$$

$$\dot{\ell}_2(t) = \frac{a_1}{A_2}\sqrt{2g(\ell_1(t) + L_{1_\mathrm{ss}})} - \frac{a_2}{A_2}\sqrt{2g(\ell_2(t) + L_{2_\mathrm{ss}})}. \tag{2.14}$$

Finally, linearizing (2.13), (2.14), about $(\ell_1 = 0, \ell_2 = 0, u = 0)$, we obtain

$$\dot{\ell}_1(t) = \alpha_1\ell_1(t) + \beta_1 u(t), \tag{2.15}$$

$$\dot{\ell}_2(t) = \alpha_2\ell_2(t) + \beta_2\ell_1(t), \tag{2.16}$$

where

$$\alpha_1 \triangleq -\frac{a_1}{A_1}\sqrt{\frac{g}{2L_{1_{\rm ss}}}}, \qquad \beta_1 \triangleq \frac{K_{\rm p}}{A_1},$$
$$\alpha_2 \triangleq -\frac{a_2}{A_2}\sqrt{\frac{g}{2L_{2_{\rm ss}}}}, \qquad \beta_2 \triangleq \frac{a_1}{A_2}\sqrt{\frac{g}{2L_{1_{\rm ss}}}}. \tag{2.17}$$

Next, we address the level control problem for Tank 2 (i.e., set-point tracking of $L_2(t)$) via a subsystem decomposition of (2.15) and (2.16). In particular, we consider the level control for L_2 via the subsystem dynamics (2.16) with ℓ_2 and ℓ_1 as the subsystem output and input, respectively. The level control problem for the Tank 2 subsystem necessitates the control of level L_1 in Tank 1. The problem of controlling L_1 is addressed via the subsystem dynamics (2.15) with ℓ_1 and u as the subsystem output and input, respectively.

Now, we develop the transfer function models for the subsystem dynamics (2.15) and (2.16). Thus, taking the Laplace transform of (2.15) and arranging terms, we obtain

$$G_1(s) \triangleq \frac{\ell_1(s)}{u(s)} = \frac{\beta_1}{s-\alpha_1}, \tag{2.18}$$

where $\ell_1(s) \triangleq \mathcal{L}[\ell_1(t)]$ and $u(s) \triangleq \mathcal{L}[u(t)]$ and $\mathcal{L}$ is the Laplace operator. Similarly, taking the Laplace transform of (2.16) and arranging terms, we obtain

$$G_2(s) \triangleq \frac{\ell_2(s)}{\ell_1(s)} = \frac{\beta_2}{s-\alpha_2}, \tag{2.19}$$

where $\ell_2(s) \triangleq \mathcal{L}[\ell_2(t)]$.

The numerical values of the parameters for the laboratory two-tank water level control system are provided in Table 1 below. Note that the variables α_i and β_i, for $i = 1, 2$, in (2.17) are computed with $L_{1_{\rm ss}} = L_{2_{\rm ss}} = 12$cm.

Physical quantity	Symbol	Numerical value	Units
Tank 1, 2 diameters	D_1, D_2	4.425	cm
Tank 1, 2 orifice diameters	d_1, d_2	0.47625	cm
Pump constant	$K_{\rm p}$	4.6	$\frac{\text{cm}^3}{\text{Volts-sec}}$
Gravitational constant	g	980	$\frac{\text{cm}}{\text{sec}^2}$

Table 1: Numerical Values for Physical Parameters of Two-Tank Level Control System

3. Objective

Proportional-plus-integral (PI) control of the state-coupled, two-tank system to track a desired level of water in Tank 2.

4. Equipment List

i) PC with Q4 data acquisition card and terminal board

ii) Software environment: Windows, MATLAB, Simulink, RTW, and QuaRC

iii) Water Tank apparatus with a water basin

iv) Universal power module: UPM-2405

v) Set of leads

5. Experimental Procedure

i) Using the set of leads, universal power module, water tank apparatus, and the terminal board of the Q4 data acquisition card, complete the wiring diagram shown in Figure 2.

ii) Start MATLAB. In the MATLAB window, choose "C:\ControlLab\Experiment6" from the Current Directory window. This directory path choice will change the directory from the default MATLAB directory to the directory where all files needed to perform Experiment 6 are stored.

iii) You can now perform various steps for the level control of coupled water tanks. However, before proceeding, you **must** request your laboratory teaching assistant to check your electrical connections.

iv) From the **File** menu of the MATLAB window, select the option **Open** to load the Simulink block-diagram "Experiment6_A.mdl" shown in Figure 3 to your desktop. This will load the files for calibrating the pressure sensor voltage when there is no water in Tanks 1 and 2. The voltage measured on S1 and S2 should be 0 Volts.

a) In the MATLAB window, click the black **Start** arrow button to acquire the voltages measured on S1 (the level of water in Tank 1) and S2 (the level of water in Tank 2).

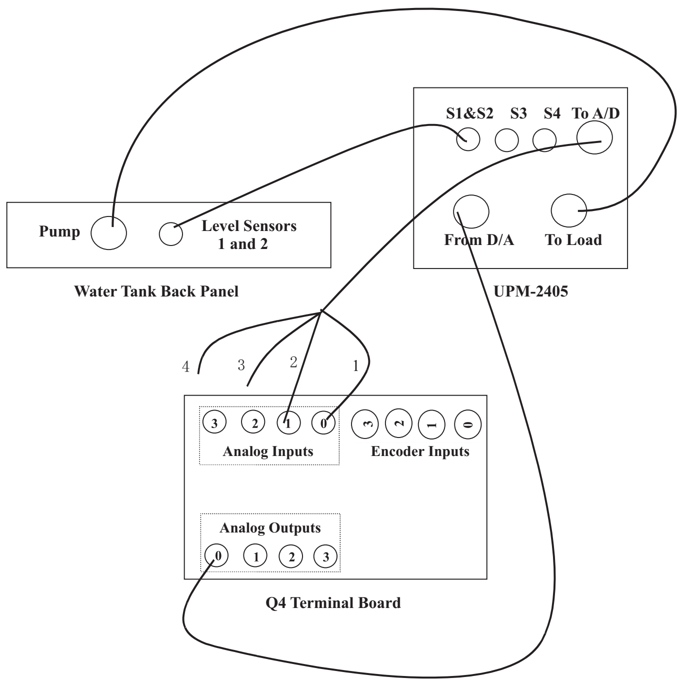

Figure 2: Wiring Diagram for Two-Tank Level Control

b) Adjust the offset **potentiometers** 1 and 2 on the water tank apparatus back panel to obtain 0 Volts.

c) In the MATLAB window, click the black **Stop** square button when you finish calibrating the sensor off-set.

v) Fill water into Tank 1 upto the 25cm level. The voltage measured on S1 should now be about **4.1** Volts.

a) In the MATLAB window, click the black **Start** arrow button to acquire the voltage measured on S1 (pressure sensor).

b) Adjust the gain **potentiometer** 1 on the water tank apparatus back panel to obtain any where between **4.0 to 4.2** Volts on S1 (pressure sensor).

c) In the MATLAB window, click the black **Stop** square button when you finish calibrating the sensor gain.

vi) Repeat (*v*) for Tank 2.

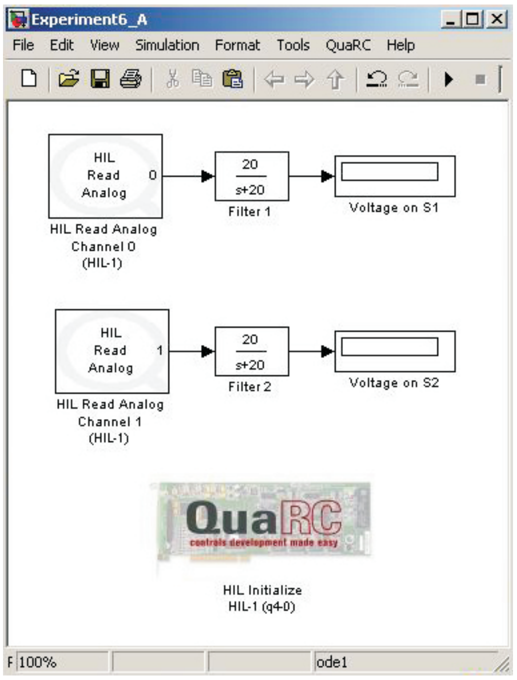
Experiment6_A
File Edit View Simulation Format Tools QuaRC Help
HIL Read Analog 0
HIL Read Analog Channel 0 (HIL-1)
20 / s+20
Filter 1
Voltage on S1
HIL Read Analog 1
HIL Read Analog Channel 1 (HIL-1)
20 / s+20
Filter 2
Voltage on S2
QuaRC controls development made easy
HIL Initialize HIL-1 (q4-0)
100%
ode1

Figure 3: Simulink Block-Diagram for Water Pressure Sensor Offset and Gain Calibration

vii) Close the currently opened Simulink diagram. From the **File** menu of the MATLAB window, select the option **Open** to load the Simulink block-diagram “Experiment6_B.mdl” shown in Figure 4 to your desktop. A plot window will also appear on your desktop. The various Simulink subblocks used in Figure 4 are given in detail in Figures 5 and 6.

a) At the MATLAB command prompt, execute the script **Experiment6.m**. This will assign the numerical values of the physical parameters of the two-tank level control system.

b) In Figure 4, under the subblock labeled **Tank 1 Controller** (Figure 5), the gains $k_{\mathrm{p}1}$ and $k_{\mathrm{i}1}$ must be designed and supplied by you. In particular, design a PI controller so that the closed-loop Tank 1 subsystem response exhibits a peak overshoot less than 1.5% and settling time less than 10 seconds. Note that $G_1(s)$ given by (2.18) denotes the open-loop transfer function for the Tank 1 subsystem. Furthermore, note that in Figure 4, a feedforward controller based on (2.12) is also implemented for the Tank 1 subsystem to account for the V_{pss} term in (2.12).

c) In Figure 4, under the subblock labeled **Tank 2 Controller** (Figure 6), the gains

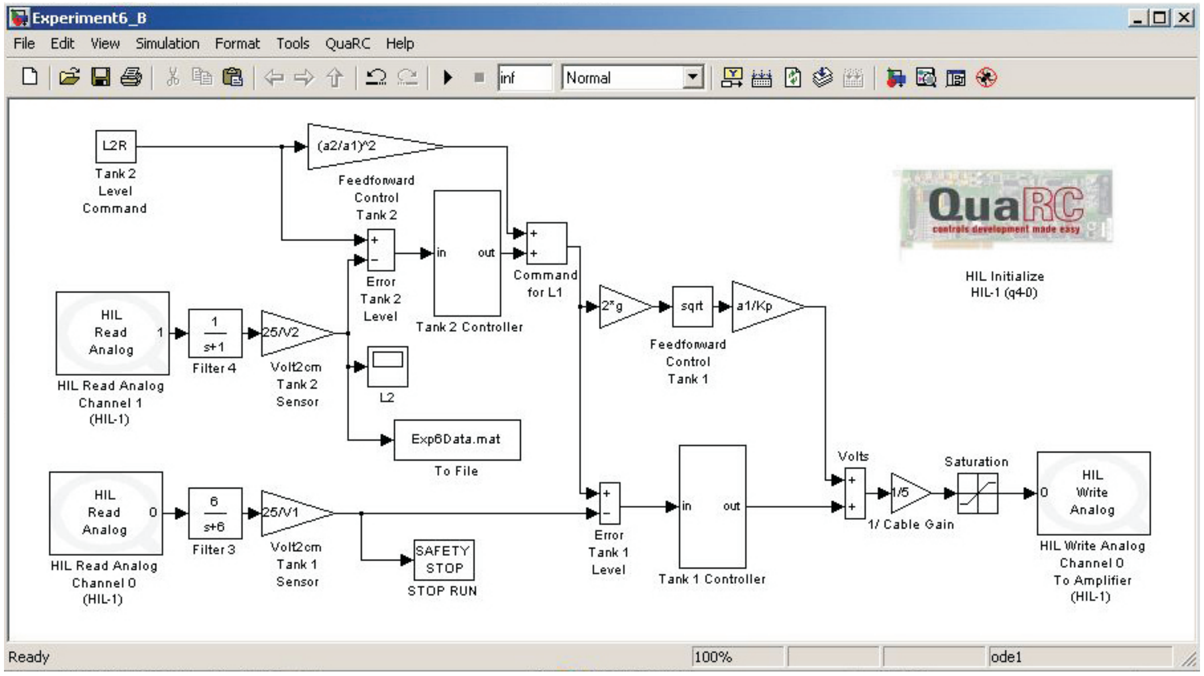

Figure 4: Simulink Block-Diagram for Two-Tank System PI Control

$k_{\mathrm{p}2}$ and $k_{\mathrm{i}2}$ must also be designed and supplied by you. In particular, design a PI controller so that the closed-loop Tank 2 subsystem response exhibits a peak overshoot less than 3.5% and settling time less than 20 seconds. Note that $G_2(s)$ given by (2.19) denotes the open-loop transfer function for the Tank 2 subsystem. Furthermore, note that in Figure 4, a feedforward controller based on (2.10) is implemented for the Tank 2 subsystem to account for the $L_{1_{\mathrm{ss}}}$ term in (2.10).

d) Before proceeding, you **must** request your laboratory teaching assistant to approve your gain values. In the MATLAB window, click the black **Start** arrow button to acquire the transient and steady-state step response of the level of water in Tank 2.

6. Analysis

i) Analyze the script "Experiment6.m" and the simulink control diagram "Experiment6_B.mdl."

ii) Build a nonlinear simulation model (using Simulink) for the two-tank level control system.

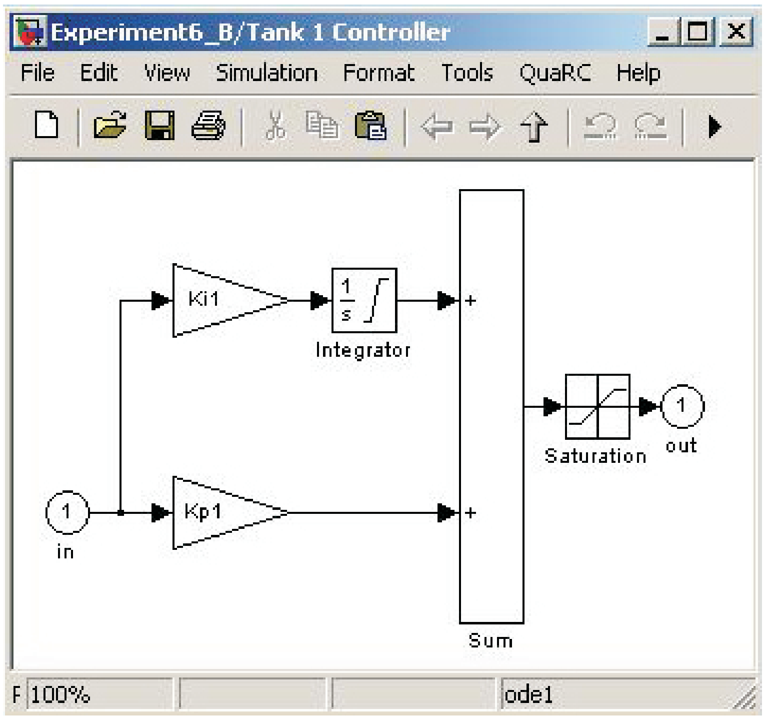

Figure 5: Tank 1 Controller Subblock

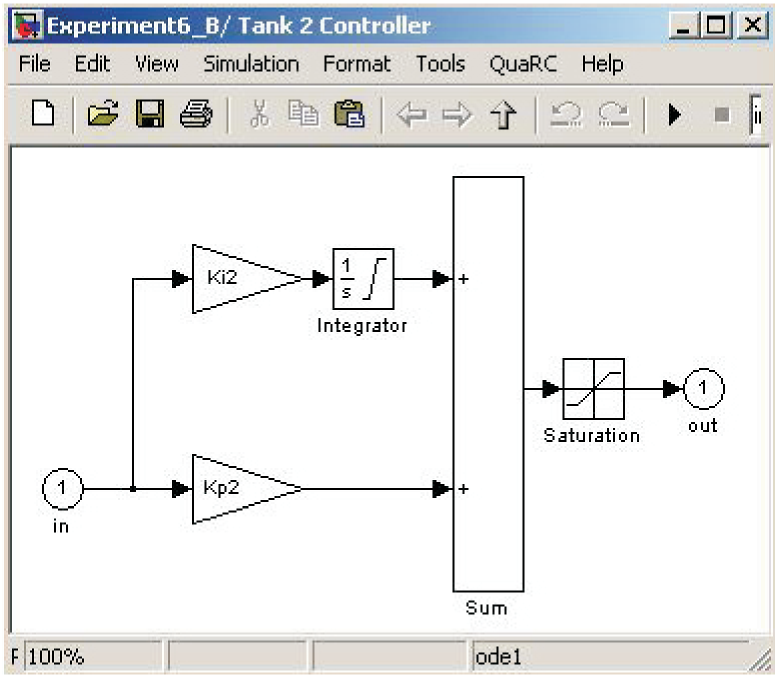

Figure 6: Tank 2 Controller Subblock

Note that as in the laboratory setup, for the simulation model the input voltage to the pump must be limited to 22 Volts. Obtain the open-loop response of the system. In addition, obtain the closed-loop response of the simulation model. How does the simulated system response compare with the experimental response?

iii) Obtain the closed-loop response for the simulation model of the two-tank level control system with *a)* only the PI controller and *b)* only the feedforward controller.

iv) Analyze and comment on your experimental results. Specifically, analyze the experimental time response of water levels in Tanks 1 and 2. Does the system response meet the performance specifications? Explain.

v) Obtain the experimental response of the two-tank system to disturbances. Note that addition of water into Tank 1 and/or Tank 2 from any source other than the pump constitutes an exogenous disturbance.

References

1. R. N. Bateson, *Introduction to Control System Technology*, Prentice-Hall, Upper Saddle River, NJ, 1999, 6th Ed., pp. 304–307.