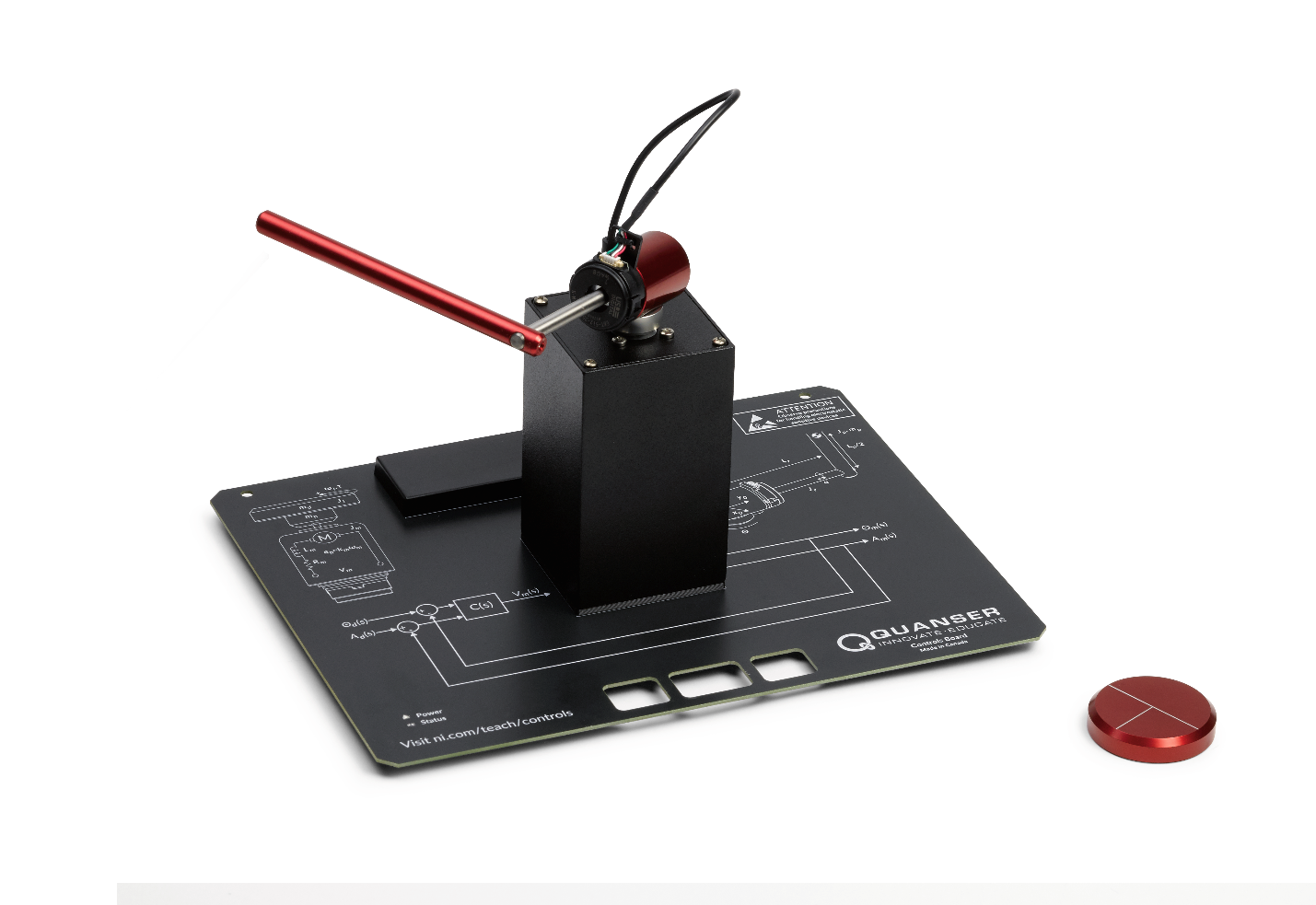


Lab Manual: Control Systems Design and Analysis

Using the Quanser Controls Board for NI ELVIS III



© 2018 Quanser Inc., All Rights Reserved

Printed in Markham, Ontario.

This document and the software described in it are provided subject to a license agreement. Neither the software nor this document may be used or copied except as specified under the terms of that license agreement. Quanser Inc. grants the following rights: a) The right to reproduce the work, to incorporate the work into one or more collections, and to reproduce the work as incorporated in the collections, b) to create and reproduce adaptations provided reasonable steps are taken to clearly identify the changes that were made to the original work, c) to distribute and publically perform the work including as incorporated in collections, and d) to distribute and publicly perform adaptations. The above rights may be exercised in all media and formats whether now known or hereafter devised. These rights are granted subject to and limited by the following restrictions: a) You may not exercise any of the rights granted to You in above in any manner that is primarily intended for or directed toward commercial advantage or private monetary compensation, and b) You must keep intact all copyright notices for the Work and provide the name Quanser Inc. for attribution. These restrictions may not be waved without express prior written permission of Quanser Inc.

LabVIEW and National Instruments are trademarks of National Instruments.

All other trademarks or product names are the property of their respective owners.

**Additional Disclaimers:** The reader assumes all risk of use of this resource and of all information, theories, and programs contained or described in it. This resource may contain technical inaccuracies, typographical errors, other errors and omissions, and out-of-date information. Neither the author nor the publisher assumes any responsibility or liability for any errors or omissions of any kind, to update any information, or for any infringement of any patent or other intellectual property right.

Neither the author nor the publisher makes any warranties of any kind, including without limitation any warranty as to the sufficiency of the resource or of any information, theories, or programs contained or described in it, and any warranty that use of any information, theories, or programs contained or described in the resource will not infringe any patent or other intellectual property right. THIS RESOURCE IS PROVIDED “AS IS.” ALL WARRANTIES, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, ANY AND ALL IMPLIEDWARRANTIES OFMERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NON-INFRINGEMENT OF INTELLECTUAL PROPERTY RIGHTS, ARE DISCLAIMED.

No right or license is granted by publisher or author under any patent or other intellectual property right, expressly, or by implication or estoppel.

IN NO EVENT SHALL THE PUBLISHER OR THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL, COVER, ECONOMIC, OR CONSEQUENTIAL DAMAGES ARISING OUT OF THIS RESOURCE OR ANY INFORMATION, THEORIES, OR PROGRAMS CONTAINED OR DESCRIBED IN IT, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES, AND EVEN IF CAUSED OR CONTRIBUTED TO BY THE NEGLIGENCE OF THE PUBLISHER, THE AUTHOR, OR OTHERS. Applicable law may not allow the exclusion or limitation of incidental or consequential damages, so the above limitation or exclusion may not apply to you.

Table of Contents

[Introduction 6](#_Toc514420211)

[Learning Objectives 6](#_Toc514420212)

[Prerequisites 6](#_Toc514420213)

[Completed Courses 6](#_Toc514420214)

[Hardware, Software, and Tool Knowledge 7](#_Toc514420215)

[Organization of the Lab Manual 7](#_Toc514420216)

[Lab 1: DC Motor Modeling 7](#_Toc514420217)

[Lab 2: Speed Control 7](#_Toc514420218)

[Lab 3: Position Control 7](#_Toc514420219)

[Lab 4: Stability 7](#_Toc514420220)

[Lab 5: Inverted Pendulum Control 8](#_Toc514420221)

[Lab 6: Digital Control 8](#_Toc514420222)

[Lab Tools and Technology 9](#_Toc514420223)

[Platform: NI ELVIS III 9](#_Toc514420224)

[Hardware: Quanser Controls Board 10](#_Toc514420225)

[Software: LabVIEW 11](#_Toc514420226)

[Lab 1: DC Motor Modeling 12](#_Toc514420227)

[Learning Objectives 13](#_Toc514420228)

[Required Tools and Technology 14](#_Toc514420229)

[Expected Deliverables 15](#_Toc514420230)

[Section 1: First Principles Modeling 16](#_Toc514420231)

[1.1 Theory and Background 16](#_Toc514420232)

[1.2 Implement 19](#_Toc514420233)

[1.3 Analyze 20](#_Toc514420234)

[Section 2: Experimental Modeling 22](#_Toc514420235)

[2.1 Theory and Background 22](#_Toc514420236)

[2.2 Implement 25](#_Toc514420237)

[2.3 Analyze 26](#_Toc514420238)

[Lab 2: DC Motor Speed Control 27](#_Toc514420239)

[Learning Objectives 28](#_Toc514420240)

[Required Tools and Technology 29](#_Toc514420241)

[Expected Deliverables 30](#_Toc514420242)

[Section 1: Qualitative PI Control Design 31](#_Toc514420243)

[1.1 Theory and Background 31](#_Toc514420244)

[1.2 Implement 32](#_Toc514420245)

[1.3 Analyze 33](#_Toc514420246)

[Section 2: Qualitative PI Control Design 34](#_Toc514420247)

[2.1 Theory and Background 34](#_Toc514420248)

[2.2 Implement 37](#_Toc514420249)

[2.3 Analyze 38](#_Toc514420250)

[Lab 3: DC Motor Position Control 39](#_Toc514420251)

[Learning Objectives 40](#_Toc514420252)

[Required Tools and Technology 41](#_Toc514420253)

[Expected Deliverables 42](#_Toc514420254)

[Section 1: Quantitative PD Control Design 43](#_Toc514420255)

[1.1 Theory and Background 43](#_Toc514420256)

[1.2 Implement 45](#_Toc514420257)

[1.3 Analyze 47](#_Toc514420258)

[Lab 4: Inverted Pendulum Control 48](#_Toc514420259)

[Learning Objectives 49](#_Toc514420260)

[Required Tools and Technology 50](#_Toc514420261)

[Expected Deliverables 51](#_Toc514420262)

[Section 1: Inverted Pendulum Control 52](#_Toc514420263)

[1.1 Theory and Background 52](#_Toc514420264)

[1.2 Implement 59](#_Toc514420265)

[1.3 Analyze 63](#_Toc514420266)

[Section 2: Optimal Control of an Inverted Pendulum 65](#_Toc514420267)

[2.1 Theory and Background 65](#_Toc514420268)

[2.2 Implement 66](#_Toc514420269)

[2.3 Analyze 70](#_Toc514420270)

[Section 3: Swing-Up Hybrid Control 71](#_Toc514420271)

[3.1 Theory and Background 71](#_Toc514420272)

[3.2 Implement 74](#_Toc514420273)

[3.3 Analyze 76](#_Toc514420274)

# Introduction

As automation and connected devices move from industry to commercial products and the home, an understanding of the design and implementation of control systems on hardware is essential. The courseware progression that accompanies the Quanser Controls Board begins with a grounding in the basics of modeling and control. Topics then transition into more complex strategies including optimal control, hybrid control, and digital control. The skills and hands-on experiences gained using the Controls Board are directly applicable to the challenges engineers face creating the complex systems that dominate the world today.

## Learning Objectives

After completing the labs and projects in this manual, you should have the ability to complete the following actions.

1. Model a first-order system both experimentally and theoretically
2. Create a control system to meet a set of desired specifications
3. Determine the stability of a system
4. Create a controller to control an unstable system
5. Create an optimal controller to govern the behavior of a complex coupled system
6. Control a digital system with a limited sampling rate

### Prerequisites

This lab manual was designed for students who have completed the following courses and have a working knowledge of the following hardware, software, and tools.

### Completed Courses

1. Differential Equations or equivalent
2. Linear Algebra or equivalent
3. Dynamic Systems or equivalent

### Hardware, Software, and Tool Knowledge

1. Basic experience using the LabVIEW graphical programming language. Learn what you need to know [here](http://www.ni.com/academic/students/learn-labview/).

## Organization of the Lab Manual

The lab manual for the Quanser Controls Board is divided into a collection of laboratory sessions that focus on the fundamental aspects of modern control design. Each session consists of a background section that can be used to orient and refresh student knowledge on the topic, as well as to serve as a reference for the concepts addressed. The second section presents a procedure to implement and validate the concepts on hardware, to relate the theoretical foundation to an applied context. The final section focuses on assessment and specific observations to ensure that the learning objectives of the exercise are achieved.

### Lab 1: DC Motor Modeling

This section covers modeling of a DC motor including first principles theoretical modeling, experimental modeling, and modeling of frequency domain characteristics.

### Lab 2: Speed Control

This section covers speed control of a DC motor. The sequence begins with basic proportional control, and then moves on to qualitative PI control, and quantitative control design to specifications. The session concludes with a presentation of Lead control design.

### Lab 3: Position Control

This section covers the design and implementation of a DC motor position controller. The sequence concludes with an investigation of steady state error characteristics.

### Lab 4: Stability

This section covers the basics of stability analysis including BIBO stability, Nyquist stability analysis, and the Routh Hurwitz coefficient test.

### Lab 5: Inverted Pendulum Control

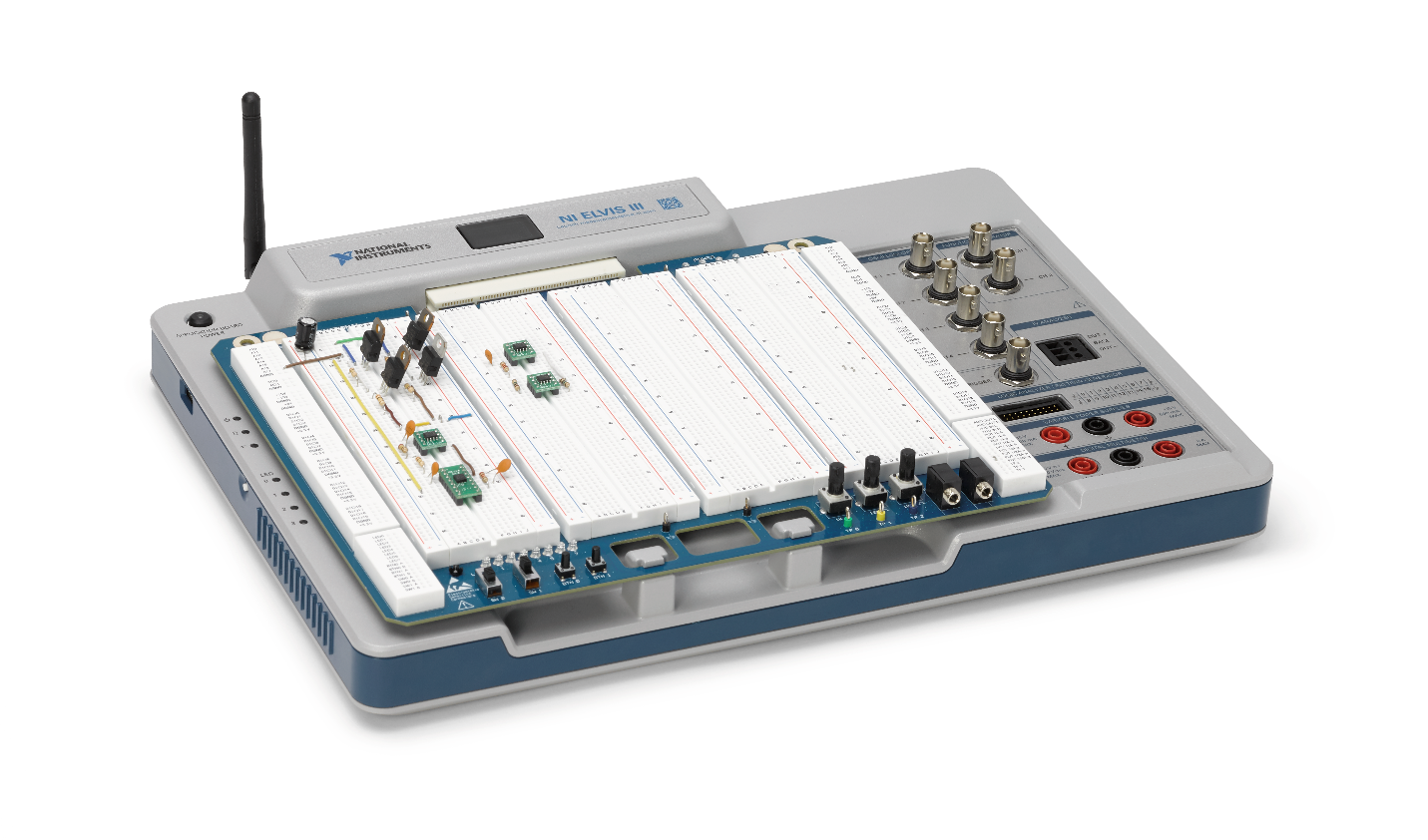
This section covers the control of an inverted Furuta pendulum. The progression begins with basic experimental modeling, before moving into PID control, optimal control, and hybrid control.

### Lab 6: Digital Control

This section covers the basics of digital control of a DC motor. The sequence begins with an introduction to digital control concepts and quantization, before moving into matched pole-zero discretization, and then finally digital control design.

## Lab Tools and Technology

### Platform: NI ELVIS III

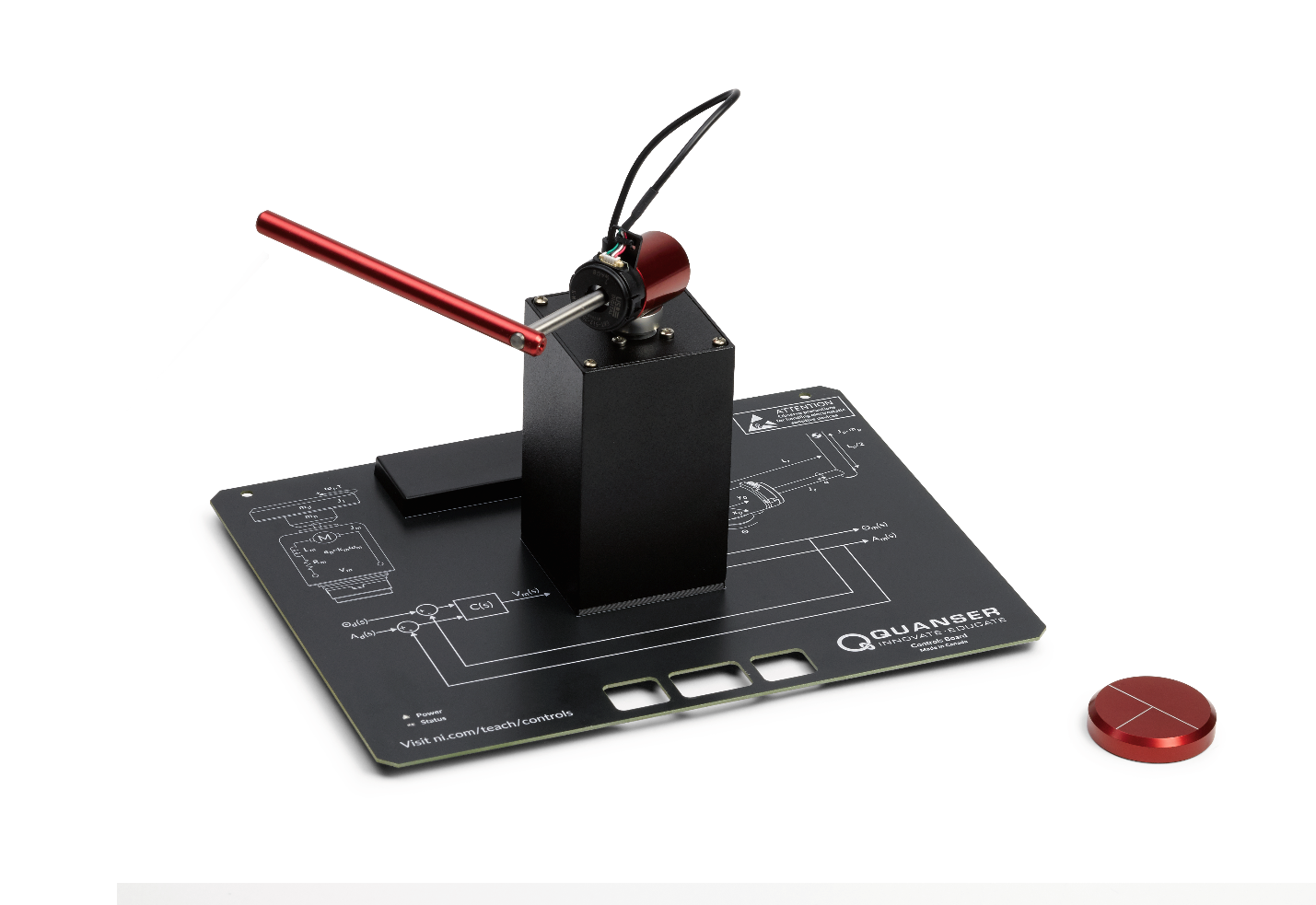


The NI Educational Laboratory Virtual Instrumentation Suite (NI ELVIS) is an engineering laboratory solution for project-based learning that combines instrumentation and embedded design with a web-driven experience to create an active learning environment in the lab and studio and flipped classrooms, delivering a greater understanding of engineering fundamentals and system design. NI ELVIS addresses engineering curriculum by integrating project-based learning, teamwork, and design with course-specific application boards and labs developed by experts from education and industry. NI ELVIS, as a programmable platform, gives educators the ability to scale to future multidisciplinary applications driving student employability.

**

Learn more at [http://www.ni.com/en-us/support/model.ni-elvis-iii.html](http://www.ni.com/en-us/support/model.ni-elvis-iii.html%20)

### Hardware: Quanser Controls Board



Designed exclusively for NI ELVIS platform, the application board is designed to accurately match dynamic models based on both physical principles and experimental tests. The servo has a highly linear motor response allowing for accurate modeling and control design. With integrated encoder feedback for both the motor and the optional pendulum accessory the servo can be easily configured for control tasks ranging from motor speed and position control up to and including inverted pendulum swing-up and balance control.



Learn more at <http://www.ni.com/en-us/support/model.quanser-controls-board-for-ni-elvis-iii.html>

### Software: LabVIEW



LabVIEW is systems engineering software for applications that require test, measurement, and control with rapid access to hardware and data insights. A widely-used, industry-standard tool for engineering system design, LabVIEW offers a graphical programming approach that helps visualize every aspect of an application. This visualization makes it easy to design engineering systems, convey concepts, and help students focus their time on the theory rather than get bogged down in the low-level implementation.

**

Learn more at <http://www.ni.com/en-us/shop/labview/labview-details.html>

# Lab 1: DC Motor Modeling



Figure 0-1: One of the many application of DC motor modeling and control

The design and implementation of a control system almost always begins with the creation of a model of the plant that is to be controlled. This is because in industry it is rare to be able to test, tune, and characterize the performance of a controller on a plant without risk of damage or injury. This is also because, as illustrated in the example in Figure 5, a controller often needs to be designed and tuned enough for basic operation before final tuning and validation are even possible. That being said, the creation of a theoretical model of a complex system can be prohibitively difficult, resulting in the creation of various approaches to experimental modeling to characterize complex systems. The ultimate goal of modeling for control design is to be able to create a simulation of a plant and various controllers in order to gain confidence that an approach will work before implementation on hardware.

## Learning Objectives

After completing this lab, you should be able to complete the following activities.

1. Create an electromechanical model of a DC motor using first principles.
2. Inspect the step response of a DC motor to characterize a transfer function.
3. Use Bode plotting to experimentally determine the model of a DC motor.

## Required Tools and Technology

|  |  |
| --- | --- |
| Platform: NI ELVIS III | * View User Manual   http://www.ni.com/en-us/support/model.ni-elvis-iii.html |
| Hardware: Quanser Controls Board | * View User Manual   http://www.ni.com/en-us/support/model.quanser-controls-board-for-ni-elvis-iii.html |
| Software: LabVIEW Version 18.0 or Later  Toolkits and Modules:   * LabVIEW Real-Time Module * NI ELVIS III Toolkit * LabVIEW Control Design & Simulation | * Before downloading and installing software, refer to your professor or lab manager for information on your lab’s software licenses and infrastructure * Download & Install for NI ELVIS III   <http://www.ni.com/academic/download>   * View Tutorials   http://www.ni.com/academic/students/learn-labview/ |

## Expected Deliverables

In this lab, you will collect the following deliverables:

* LabVIEW first principles model of the DC motor
* Calculated equivalent moment of inertia acting on the motor shaft
* Response of the first principles model and hardware
* Transfer function representing the first order model
* Experimental step-response of the DC motor
* Transfer function of the experimental model of the DC motor

Your instructor may expect you complete a lab report. Refer to your instructor for specific requirements or templates.

## Section 1: First Principles Modeling

### 1.1 Theory and Background

A servomotor is typically characterized by a brushed DC motor, and a sensor for measuring angular rotation. The Quanser Controls Board consists of a brushed DC motor, and high-resolution optical encoder. The model of the servomotor consists of an electrical model, and mechanical model of the dynamics of the motor mechanism. The motor armature circuit schematic is shown in Figure 1-1, and the electrical and mechanical parameters are given in Table 1-1. The DC motor shaft is connected to a load hub, which is a metal disk used to mount the inertia disk or rotary pendulum, and has a moment of inertia of *Jh*. For this modeling exercise, a disk load is attached to the load hub with a moment of inertia of *Jd*.

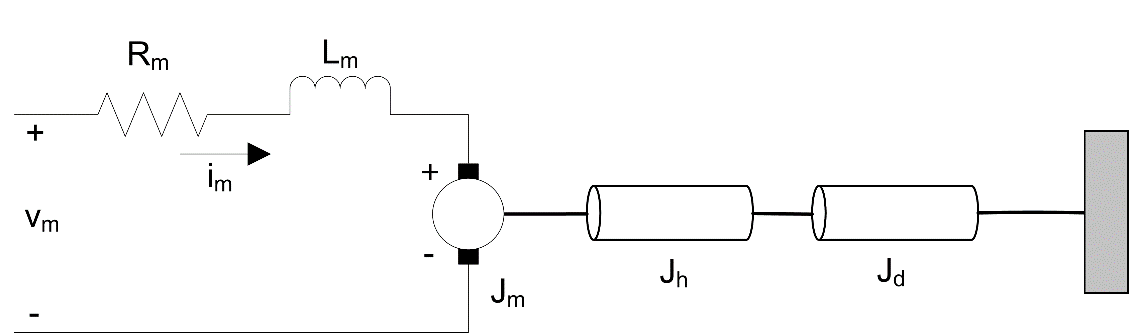


Figure 1-1: Quanser Controls Board DC motor and load

The back-emf (electromotive) voltage *eb* depends on the speed of the motor shaft, *ωm*, and the back-emf constant of the motor, *km*. It opposes the current flow. The back emf voltage characteristics are:

Equation 1-1

Table 1-1: Quanser Controls Board system parameters

|  |  |  |
| --- | --- | --- |
| Symbol | Description | Value |
| DC Motor | | |
| Rm | Terminal resistance | 8.4 Ω |
| Kt | Torque constant | 0.042 N·m/A |
| Km | Motor back-emf constant | 0.042 V/(rad/s) |
| Jm | Rotor inertia | 4.0 x 10-6 kg·m2 |
| Lm | Rotor inductance | 1.16 mH |
| Load Hub | | |
| mh | Load hub mass | 0.0106 kg |
| rh | Load hub radius | 0.0111 m |
| Disk Load | | |
| md | Load disk mass | 0.053 kg |
| rd | Load disk radius | 0.0248 m |

Using Kirchoff’s Voltage Law, you can write the following equation:

Equation 1-2

.

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

Equation 1-3

.

Solving for *im(t)*, the motor current can be found as

Equation 1-4

The motor shaft equation is expressed as

Equation 1-5

where *Jeq* is the total moment of inertia acting on the motor shaft and *τm* is the applied torque from the DC motor. Based on the current applied, the torque is

Equation 1-6

The moment of inertia of a disk about its pivot, with mass *m* and radius *r* is

Equation 1-7

### 1.2 Implement

1. Open the project **Quanser Controls Board.lvproj**, and open **DC Motor First Principles.vi**. Browse to the block diagram, shown below:

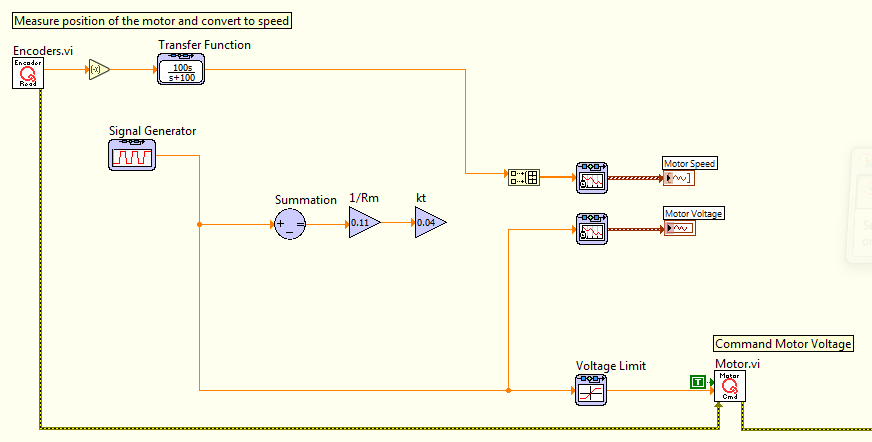


Figure 1-2: Incomplete block diagram representing the DC motor model

1. Using the equations in *Section 1.1*, assemble a simple block diagram to model the system. You will need a few Gain nodes, a Subtract node, and an Integrator node (to go from acceleration to speed). Part of the solution is shown in Figure 1-2.
2. The motor shaft of the Quanser Controls Board is attached to a load hub and a disk load. Based on the parameters given in Table 1-1, calculate the equivalent moment of inertia that is acting on the motor shaft.
3. Design the DC motor model using Control & Simulation blocks as described above. Save a screen capture of your model.
4. Run the VI with your model. The waveform chart response should be similar to Figure 1-3.

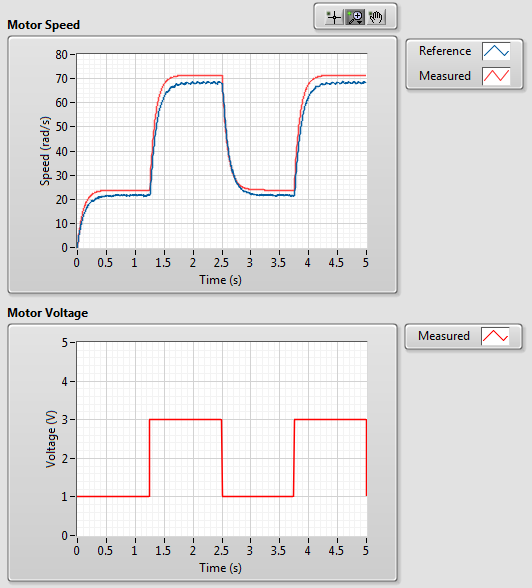


Figure 1-3: Measured and simulated response of the Quanser Controls Board DC motor

1. Save a screen capture of your waveform charts.
2. Formulate the differential equation for *ωm* using Equation 1-3 through Equation 1-6. Compare your result with the transfer function obtained from the experimental modeling experiments. Hint: Obtain the Voltage *Vm (s)* to Speed *Ωm (s)* transfer function by applying a Laplace Transform to the derived differential equation.
3. Click on the **Stop** button to stop the VI.

### 1.3 Analyze

1-1 What is the equivalent moment of inertia acting on the motor shaft that you calculated in Step 2?

1-2 Attach the screen capture you saved in Step 3.

1-3 Does your model represent the Quanser Controls Board DC motor when validated in Step 4? Attach the screen capture you saved in Step 4.

1-4 What is the differential equation for *ωm*, formulated in Step 4?

## Section 2: Experimental Modeling

### 2.1 Theory and Background

The bump test is a simple experimental test that is used to create a model based on the step response of a stable system. A step input is given to the system and its response is recorded. As an example, consider a system given by the following transfer function:

Equation 2-1

The step response shown in Figure 2-1 is generated using this transfer function with *K* = 5 rad/V − s and *τ* = 0.05 s.

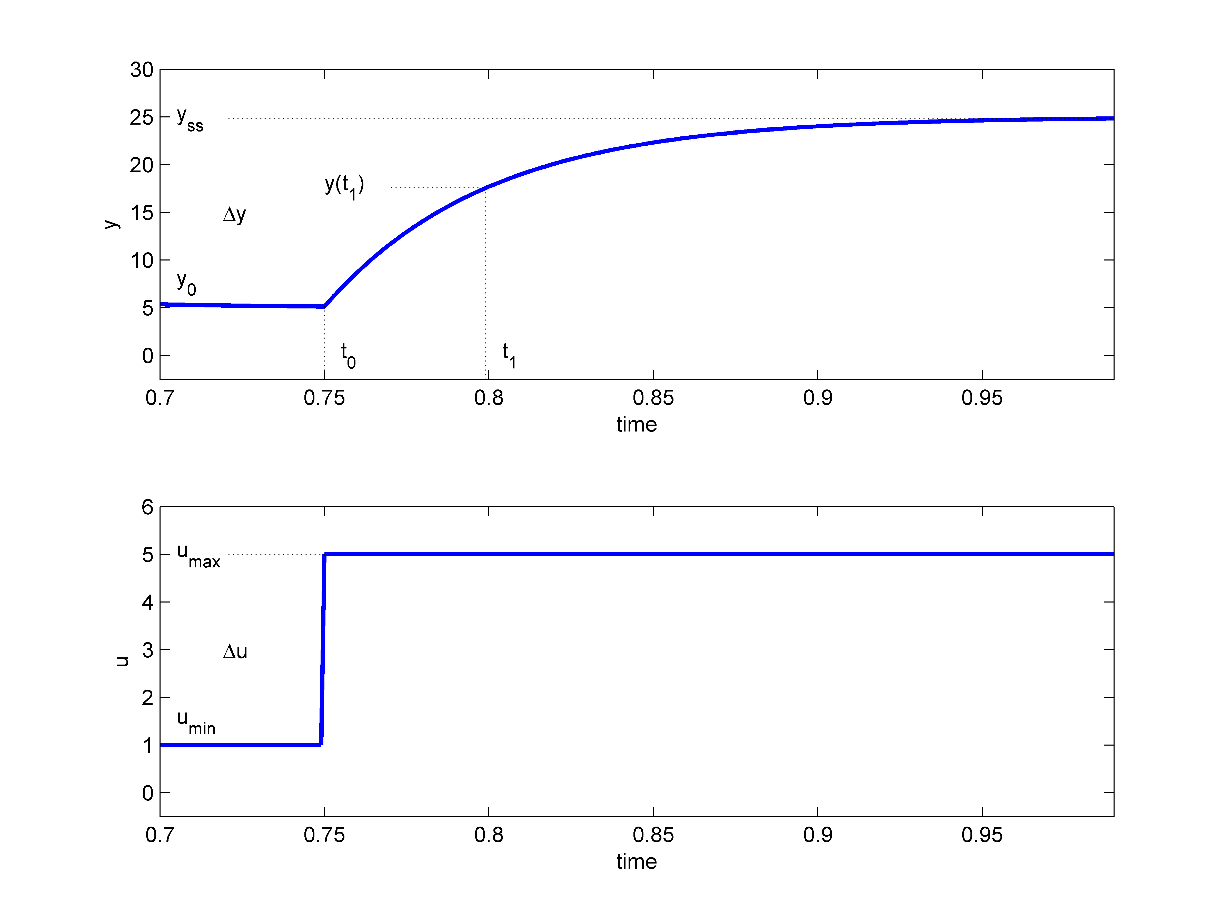


Figure 2-1: Input and output signal used in the bump test method

The step input begins at time *t*0. The input signal has a minimum value of *u*min and a maximum value of *u*max. The resulting output signal is initially at *y*0. Once the step is applied, the output tries to follow it and eventually settles at its steady-state value *y*ss. From the output and input signals, the steady-state gain is

Equation 2-2

where *∆y = yss − y*0 and *∆u = u*max *− u*min. The time constant of a system τ is defined as the time it takes the system to respond to the application of a step input to reach 1 − 1/e ≈ 63.2% of its steady-state value. For example, for the response in Figure 2-1

where

Equation 2-3

Then, you can read the time *t1* that corresponds to *y(t1)* from the response data in Figure 2-1. From this, the model time constant can be found as

Equation 2-4

When applied to the Quanser Controls Board DC motor, the s-domain representation of a step input voltage with a time delay *t*0 is given by

Equation 2-5

where *Av* is the amplitude of the step and *t*0 is the step time (i.e. the delay). The voltage-to-speed transfer function is

Equation 2-6

where *K* is the model steady-state gain, *τ* is the model time constant, *Ωm (s) = L [ωm (t)]* is the load gear rate, and *Vm (s) = L [vm (t)]* is the applied motor voltage.

If we substitute input Equation 2-5 into the system transfer function Equation 2-6, we get

Equation 2-7

We can then find the Quanser Controls Board DC motor speed step response in the time domain *ωm (t)* by taking inverse Laplace of this equation

Equation 2-8

noting the initial conditions *ωm (0-) = ωm (t*0*)*.

### 2.2 Implement

1. Open the project **Quanser Controls Board.lvproj**, and open **DC Motor Experimental.vi**.

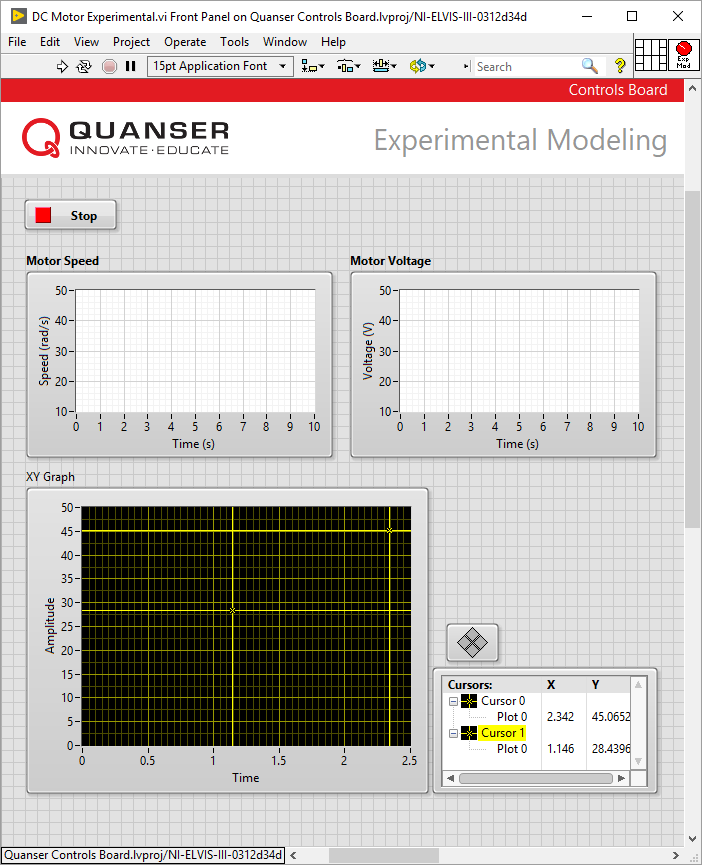


Figure 2-2: Application for experimental modeling of the Quanser Controls Board DC Motor

1. Run the VI to apply a 2 V step to the servo. The response should be similar to that shown in Figure 2-3.

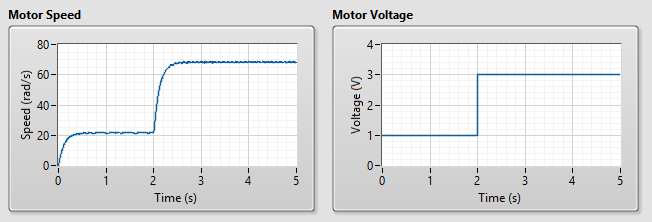


Figure 2-3: Quanser Controls Board DC Motor bump test response

1. Plot the motor speed response and the input voltage. For example, you can right click any of the waveform charts and select **Export >> Export Simplified Image** to save the measured load/disk speed and motor voltage to a bitmap image file and attach that to your report.
2. Find the steady-state gain using the measured step response. Hint: Use the Cursor palette in the XY Graph to measure points off the plot.
3. Find the time constant from the obtained response.
4. Check and record if your derived model parameters *K* and *τ* are correct.
5. Modify the VI by adding a *Transfer Function* block using the derived model parameters to plot the simulated and measured responses simultaneously. Run the VI. Save a figure displaying both the measured and simulated response in one plot, as well as in the input voltage.
6. Click on the **Stop** button to stop the VI.

### 2.3 Analyze

2-1 Attach the plot that you saved in Step 3.

2-2 What is the steady-state gain that you found in Step 4?

2-3 What is the time constant that you found in Step 5?

2-4 Attach the figure that you saved in Step 7 showing the commanded voltage, and both the measured and simulated responses.

2-5 Based on the observed response in Step 7, did you derive the model parameters *K* and *τ* correctly?

# Lab 2: DC Motor Speed Control



Figure 0-1: Vehicle cruise control is only one of the many applications of DC motor speed control

One of the most common tasks that automation, robotics, and industrial engineers are called upon to perform when creating industrial systems is to control the speed of a DC motor. From automation in manufacturing, to automotive systems, autonomous systems, and even aerospace DC motors are used to actuate critical systems, and their speed needs to be controlled to perform within specific design criteria. As an introduction to control systems, the control of a DC motor also serves as an excellent starting point because they are relatively easy to model and control. The DC motor system on the Quanser Controls Board was designed to make that experience even easier by: A) tuning the dynamics of the motor to match a theoretical linear model accurately, and B) creating an interface to the hardware that makes sending commands to the amplifier system and reading the sensors quick and easy. Despite the ease of use, the skills gained in these exercises are directly applicable to a multitude of exciting emerging application areas in engineering.

## Learning Objectives

After completing this lab, you should be able to complete the following activities.

1. Tune PI control parameters for speed control of a DC motor.
2. Design PI control parameters to meet speed control specifications.
3. Use lead compensator design to control the speed of a DC motor.

## Required Tools and Technology

|  |  |
| --- | --- |
| Platform: NI ELVIS III | * View User Manual   http://www.ni.com/en-us/support/model.ni-elvis-iii.html |
| Hardware: Quanser Controls Board | * View User Manual   http://www.ni.com/en-us/support/model.quanser-controls-board-for-ni-elvis-iii.html |
| Software: LabVIEW Version 18.0 or Later  Toolkits and Modules:   * LabVIEW Real-Time Module * NI ELVIS III Toolkit * LabVIEW Control Design & Simulation | * Before downloading and installing software, refer to your professor or lab manager for information on your lab’s software licenses and infrastructure * Download & Install for NI ELVIS III   <http://www.ni.com/academic/download>   * View Tutorials   http://www.ni.com/academic/students/learn-labview/ |

## Expected Deliverables

In this lab, you will collect the following deliverables:

* Evidence of the behavior of the DC motor with various control gains
* Step responses of the system to various sets of control gains
* Peak time and percent overshoot specifications
* Calculated control gains to meet the specifications
* Measured speed response to the designed control gains
* Measures peak time and percent overshoot
* Evidence of the effect of the design parameters on the control response

Your instructor may expect you complete a lab report. Refer to your instructor for specific requirements or templates.

## Section 1: Qualitative PI Control Design

### 1.1 Theory and Background

The speed of the Quanser Controls Board DC motor is controlled using a proportional-integral control system. The block diagram of the closed-loop system is shown in Figure 1-1:

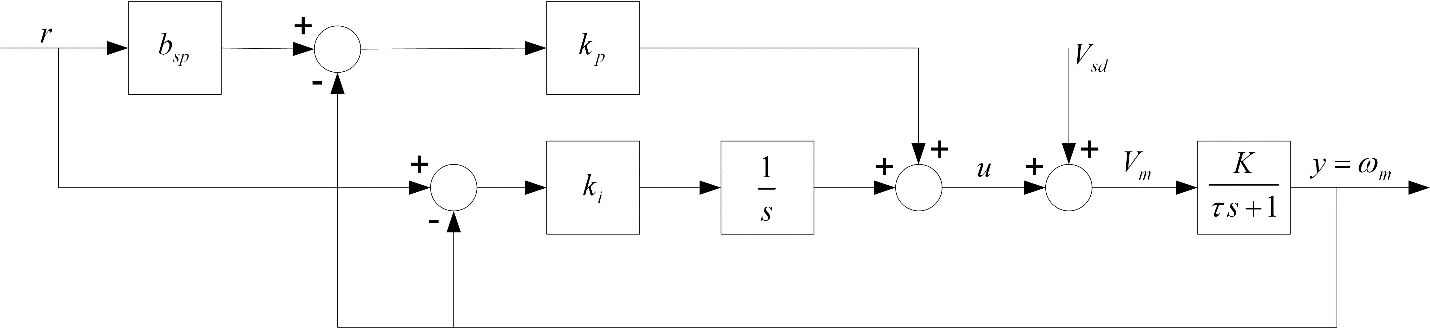


Figure 1-1: Closed loop PI control system block diagram

The transfer function representing the DC motor speed-voltage relation with steady-state gain *K* and time constant *τ* is

Equation 1-1

and will be used to design the PI controller. The input-output relation in the time-domain for a PI controller with set-point weighting is

Equation 1-2

where *kp* is the proportional gain, *ki*is the integral gain, and *bsp* is the set-point weight. The closed loop transfer function from the speed reference *r* to the angular motor speed output *ωm* is

Equation 1-3

### 1.2 Implement

1. Open the project **Quanser Controls Board.lvproj**, and then open **DC Motor Speed Control.vi** listed under the NI ELVIS III.
2. Run the VI. The DC motor should begin rotating and the scopes should look similar to Figure 1-2

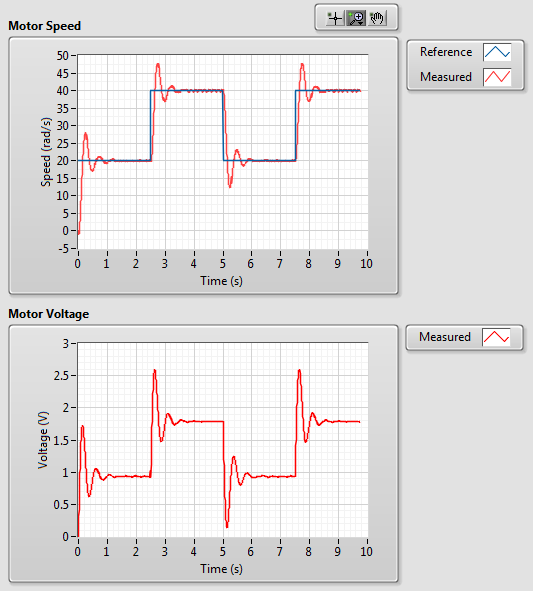


Figure 1-2: Default response of the PI control application

1. In the Signal Generator section of the front panel set:

* Amplitude (rad/s) to **10**
* Frequency (Hz) to **0.2**
* Offset (rad/s) to **80**

1. In the Control Parameters section set:

* kp (V s/rad) to **0.050**
* ki (V/rad) to **1.00**
* bsp to **0.00**

1. Examine the behavior of the measured speed, shown in red, with respect to the reference speed, shown in blue, in the Speed (rad/s) scope.
2. Increment and decrement *kp* in steps of 0.005 V s/rad.
3. Observe the changes in the measured signal with respect to the reference signal in response to the updated proportional gains.
4. Set *kp* to 0 V s/rad and *ki* to 0 V/rad. The motor should stop spinning.
5. Increment the integral gain, *ki*, in steps of 0.05 V/rad. Vary the integral gain between 0.05 V/rad and 1.00 V/rad.
6. Examine the response of the measured speed in the Speed (rad/s) scope.
7. Click on the *Stop* button to stop the VI.

### 1.3 Analyze

1-1 Explain the behavior of the measured speed with respect to the reference speed in Step 5.

1-2 Explain the observed performance differences in Step 7.

1-3 Describe and explain the response of the measured speed in Step 10 when the integral gain is set low compared to when *ki* is high.

## Section 2: Qualitative PI Control Design

### 2.1 Theory and Background

The standard second-order transfer function has the form

Equation 2-1

where ωn is the natural frequency and ζ is the damping ratio. This leads to the standard desired closed-loop characteristic polynomial of

Equation 2-2

where ω0 is the undamped closed loop frequency and ζ is the damping ratio. The denominator of the transfer function shown in Equation 1-3 in Section 1.1 is the characteristic equation of the Quanser Controls Board DC motor system, and matches the desired characteristic equation in Equation 2-2 with the following gains:

Equation 2-3

and

Equation 2-4

Large values of ω0 give large values of controller gain. The damping ratio, ζ, and the set-point weight parameter, *bsp*, can be used to adjust the speed and overshoot of the response to reference values.

There is no tachometer sensor present on the Quanser Controls Board to measure the speed. Instead, the derivative of the encoder signal is calculated using a first-order differentiating filter.

#### Second-order Step Response

The properties of the response of a second-order system as defined by Equation 2-1 depend on the values of the parameters ωn and ζ.

Consider a second-order system as shown in Equation 2-1 subjected to a step input given by

Equation 2-5

with a step amplitude of *R0* = 1.5. The system response to this input is shown in Figure 2-1, where the red trace is the output response *y(t)* and the blue trace is the step input *r(t).*

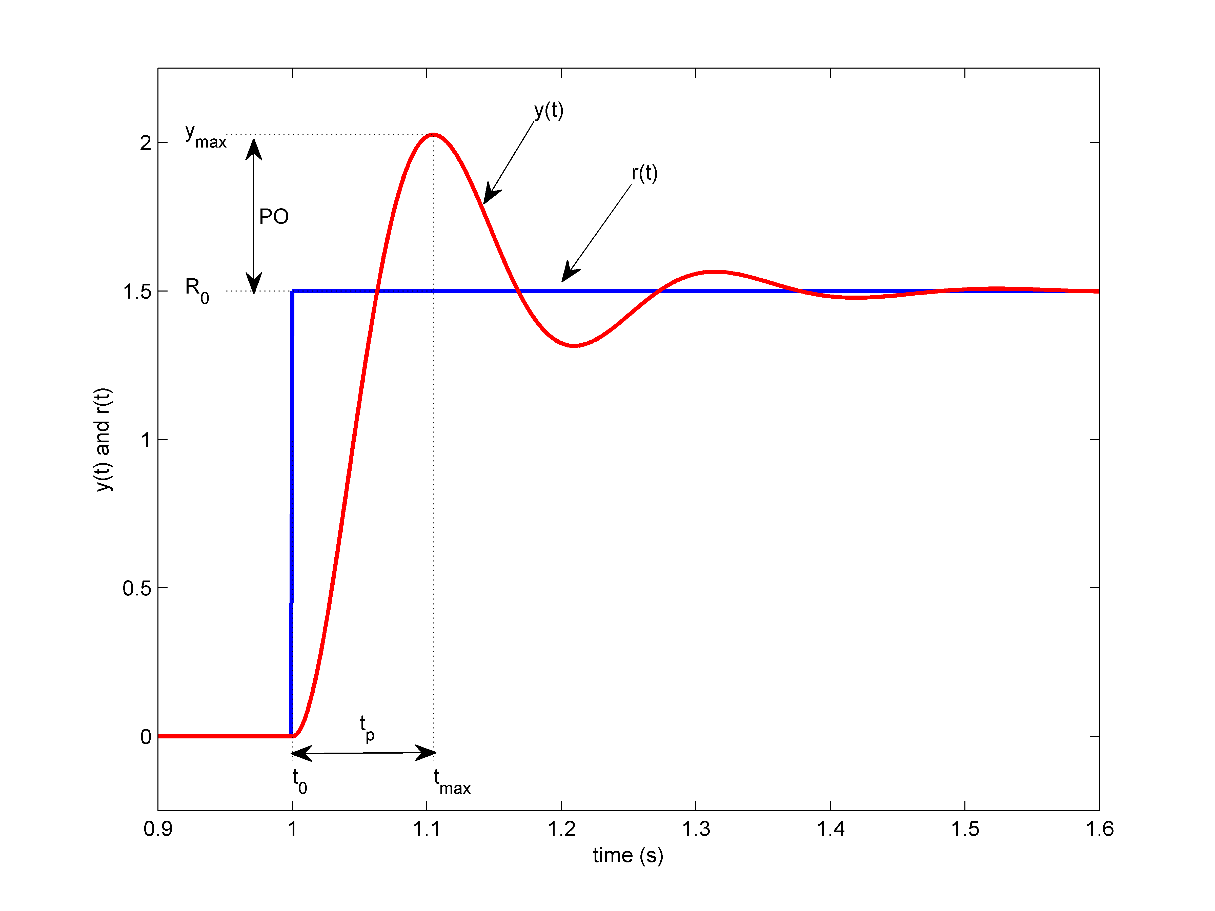


Figure 2-1: Default response of the PI control application

#### Peak Time and Overshoot

The maximum value of the response is denoted by the variable *ymax* and it occurs at a time *tmax*. For a response similar to Figure 2-1, the percent overshoot is found using

Equation 2-6

In a second-order system, the amount of overshoot depends solely on the damping ratio parameter and it can be calculated using the equation

Equation 2-7

From the initial step time *t0* the time it takes for the response to reach its maximum value is

Equation 2-8

This is called the peak time of the system. It depends on both the damping ratio and natural frequency of the system. It can be derived analytically as

Equation 2-9

Generally speaking, the damping ratio affects the shape of the response while the natural frequency affects the speed of the response.

### 2.2 Implement

#### Quantitative PI Control Design

1. Calculate the expected peak time, *tp,* and percent overshoot, *PO*, given the following design specifications:

* ζ = **0.75**,
* ω0 = **16** rad/s.

Optional: You can also design a VI that simulates the Quanser Controls Board first-order model with PI control and have it calculate the peak time and overshoot.

1. Calculate the proportional and integral control gains *kp* and *ki*, respectively, according to the design specifications for the model parameters *K* = 22.6 rad/(V s) and *τ* = 0.12 s.
2. Open the project **Quanser Controls Board.lvproj**, open **DC Motor Speed Control.vi**.
3. Run the VI. The DC motor should begin rotating and the scopes should look similar to Figure 1-2.
4. In the *Signal Generator* section of the front panel set:

* Amplitude (rad/s) to **10**
* Frequency (Hz) to **0.2**
* Offset (V) to **80**

1. In the *Control Parameters* section, enter the control gains that you found in Step 2, and ensure that the *bsp* is set to zero.
2. When you have collected two sample cycles, stop the application by clicking on the *Stop* button.
3. Capture the measured speed response. Make sure you include both the Speed (rad/s) and the control signal Voltage (V) scopes.
4. Measure the peak time and percentage overshoot of the observed response. If the specifications have not been satisfied, adjust the proportional gain *kp* and integral gain *ki* to meet the specifications and capture your system response plots.
5. Click on the **Stop** button to stop the VI.

#### Set-point Weight Analysis

1. Run **DC Motor Speed Control.vi** as outlined in the previous section.
2. In the *Signal Generator* section of the front panel set:

* Amplitude (rad/s) to **10**
* Frequency (Hz) to **0.2**
* Offset (V) to **80**

1. In the *Control Parameters* section set:

* kp (V s/rad) to **0.10**
* ki (V/rad) to **1.50**
* bsp to **0.00**

1. Increment the set-point weight parameter, *bsp*, in steps of 0.05 between 0 and 1.
2. Examine the effect that raising *bsp* has on the shape of the measured speed signal in the *Speed (rad/s)* scope.
3. Click on the **Stop** button to stop the VI.

### 2.3 Analyze

2-1 What is the expected peak time and overshoot that you calculated in Step 1?

2-2 What are the proportional and integral control gains that you calculated in Step 2?

2-3 Attach the measured speed response that you captured in Step 8.

2-4 What peak time and percentage overshoot did you measure in Step 9?

2-5 Were the specifications satisfied? If not, what gains did you use to achieve the specifications in Step 9?

2-6 What effect does increasing the specification *ζ* have on the measured speed response? How about on the control gains? Hint: Start by examining Equation 2-7.

2-7 What effect does increasing the specification *ω0* have on the measured speed response and the generated control gains? Hint: Start by examining Equation 2-9.

2-8 Explain what the set-point weight parameter is doing in Step 15.

# Lab 3: DC Motor Position Control

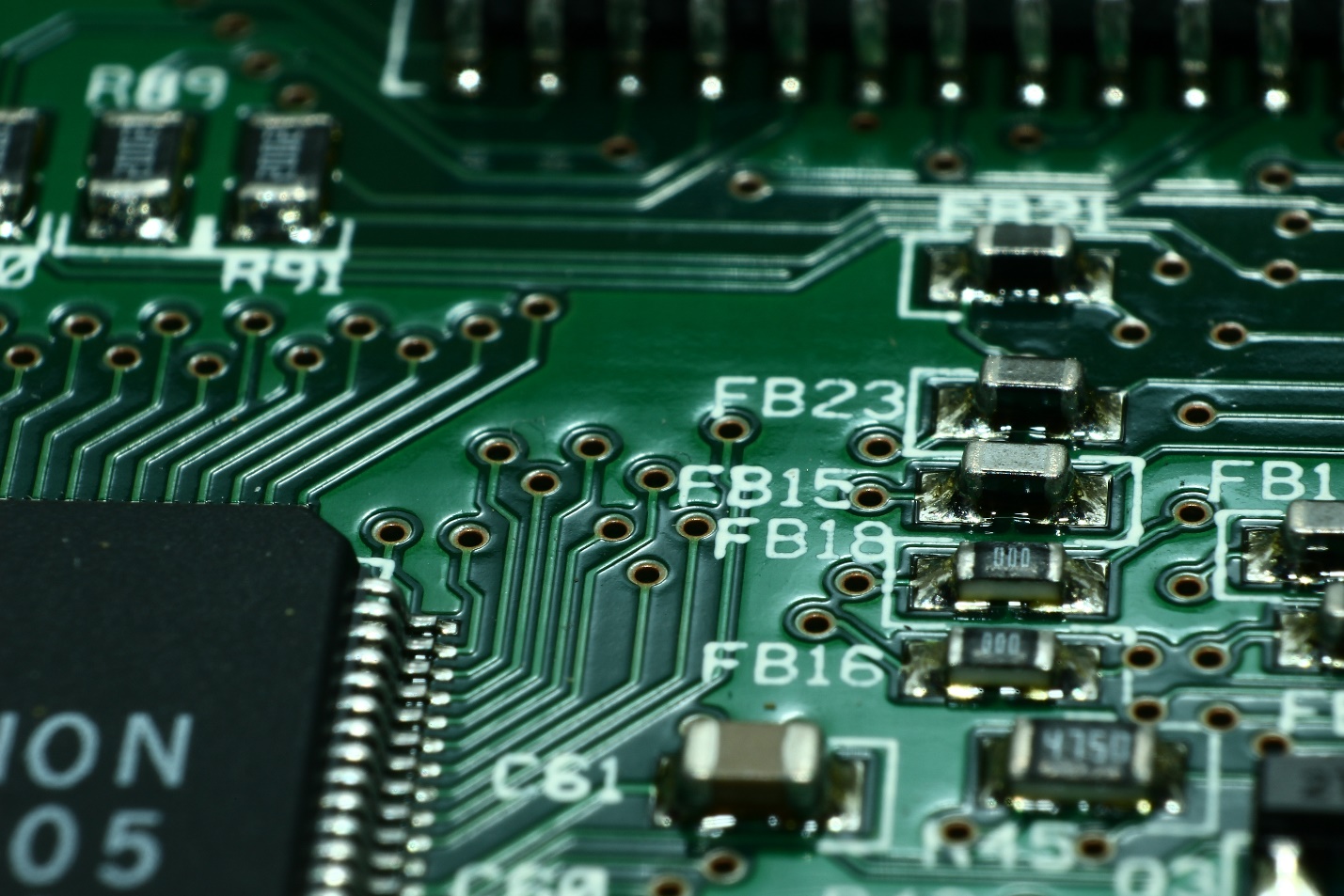


Figure 0-1: Pick and place machines for PCB manufacturing are a great example of position control

For a wide variety of applications from robotics, to manufacturing and industrial automation the accurate position control of DC motors is an essential skill. The hobby market, including RC vehicles, is also dominated by servo motors that are implemented as a combination of a brushed DC motor, sensor, and closed-loop position controller. As with many other applications of control systems, there are several methods that can be employed to control the position of a DC motor. We will briefly cover the most common, PID control, followed by a more specialized application of a compensator.

## Learning Objectives

After completing this lab, you should be able to complete the following activities.

1. Design PD control parameters to meet position control specifications.
2. Analyze the response of a position controller.
3. Assess the steady-state error of a position controller.

## Required Tools and Technology

|  |  |
| --- | --- |
| Platform: NI ELVIS III | * View User Manual   http://www.ni.com/en-us/support/model.ni-elvis-iii.html |
| Hardware: Quanser Controls Board | * View User Manual   http://www.ni.com/en-us/support/model.quanser-controls-board-for-ni-elvis-iii.html |
| Software: LabVIEW Version 18.0 or Later  Toolkits and Modules:   * LabVIEW Real-Time Module * NI ELVIS III Toolkit * LabVIEW Control Design & Simulation | * Before downloading and installing software, refer to your professor or lab manager for information on your lab’s software licenses and infrastructure * Download & Install for NI ELVIS III   <http://www.ni.com/academic/download>   * View Tutorials   http://www.ni.com/academic/students/learn-labview/ |

## Expected Deliverables

In this lab, you will collect the following deliverables:

* Calculated control gains to meet the specifications
* Measured position response to the designed control gains
* Measures peak time and percent overshoot
* Evidence of the effect of the design parameters on the control response

Your instructor may expect you complete a lab report. Refer to your instructor for specific requirements or templates.

## Section 1: Quantitative PD Control Design

### 1.1 Theory and Background

Control of motor position is a natural way to introduce the benefits of derivative action. In this experiment a proportional-integral-derivative controller is designed according to specifications. The closed-loop PID control block diagram is shown in Figure 1-1.

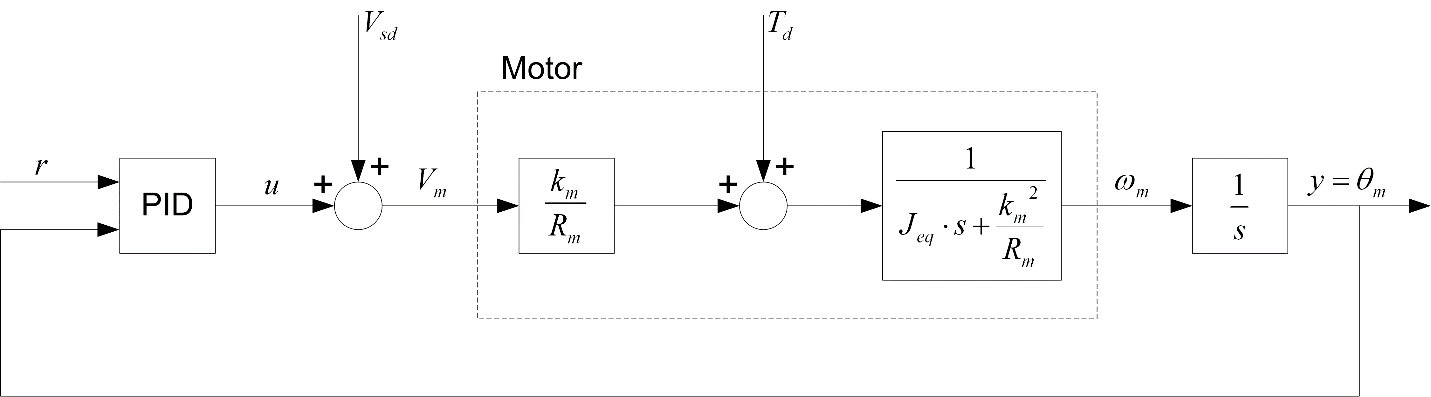


Figure 1-1: DC motor PID closed-loop block diagram

The two-degree of freedom PID transfer function inside the PID block in Figure 1-1 is

Equation 1-1

where *kp* is the position proportional control gain, *kd* is the derivative control gain, *ki* is the integral control gain, *bsp* is the set-point weight on the reference position *r(t)*, and *bsd* is the set-point weight on the velocity reference of *r(t).*

The dotted box labeled Motor in Figure 1-1 is the motor model in terms of the back-emf motor constant km, the electrical motor armature resistance *Rm*, and the equivalent moment of inertia of the motor pivot *Jeq*. The direct disturbance applied to the inertial wheel is represented by the disturbance torque variable Td and the simulated disturbance voltage is denoted by the variable *Vsd*.

#### PD Control Design

The integral term will not be used to control the servo position. The behavior of the controlling the motor position is first analyzed using a PD control. By setting *ki* = 0 in the PID control Equation 1-1 and taking its Laplace transform, the PD transfer function is

Equation 1-2

The Quanser Controls Board DC motor voltage-to-position transfer function is

Equation 1-3

where *K* = 21.9 rad/(V s) is the model steady-state gain, *τ* = 0.15 s is the model time constant, *ϴm(s)* = *L[θm(t)]* is the motor / disk position, and *Vm(s)* = *L[vm(t)]* is the applied motor voltage. If desired, you can conduct an experiment to find more precise model as outlined in the experimental modeling laboratory.

Using the PD control Equation 1-2, the closed-loop transfer function of the motor position controller is

Equation 1-4

The position controller used in this case will be implemented with *bsd* = 0, which yields a variation of the standard PD controller.

#### Rate Feedback Position Control

To improve the performance of the controller, a variation of the classic PD control will be used: rate feedback control illustrated in Figure 1-2. Unlike the standard PD, only the negative velocity is fed back (i.e. not the velocity of the error) and a low-pass filter will be used in-line with the derivative term to suppress measurement noise. The combination of a first order low-pass filter and the derivative term results in a high-pass filter *H(s)* which will be used instead of a direct derivative.

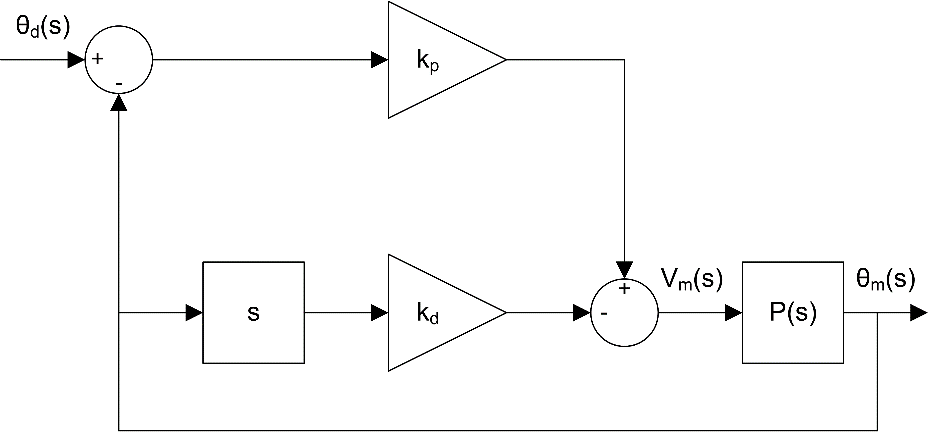


Figure 1-2: Block diagram of rate feedback control

When the velocity reference set-point weight is set to zero as outlined in the previous section, the resultant PD control transfer function is

Equation 1-5

This is a second-order transfer function. Recall the standard second-order transfer function

Equation 1-6

### 1.2 Implement

1. Find the proportional and derivative gains required for the DC motor closed-loop transfer function given in Equation 1-5 to match the standard second-order system in Equation 1-6. Your gain equations will be a function of *ωn* and *ζ*.
2. For the response to have a peak time of 0.15 s and a percentage overshoot of 2.5 %, the natural frequency and damping ratio needed are *ωn* = 32.3 rad/s and *ζ* = 0.76. Using the model parameters given above in Section 1.1 (or those you found previously through a modeling lab), calculate the control gains needed to satisfy these requirements.
3. Open the project **Quanser Controls Board.lvproj**, and then open **DC Motor Position Control.vi** listed under the NI ELVIS III.

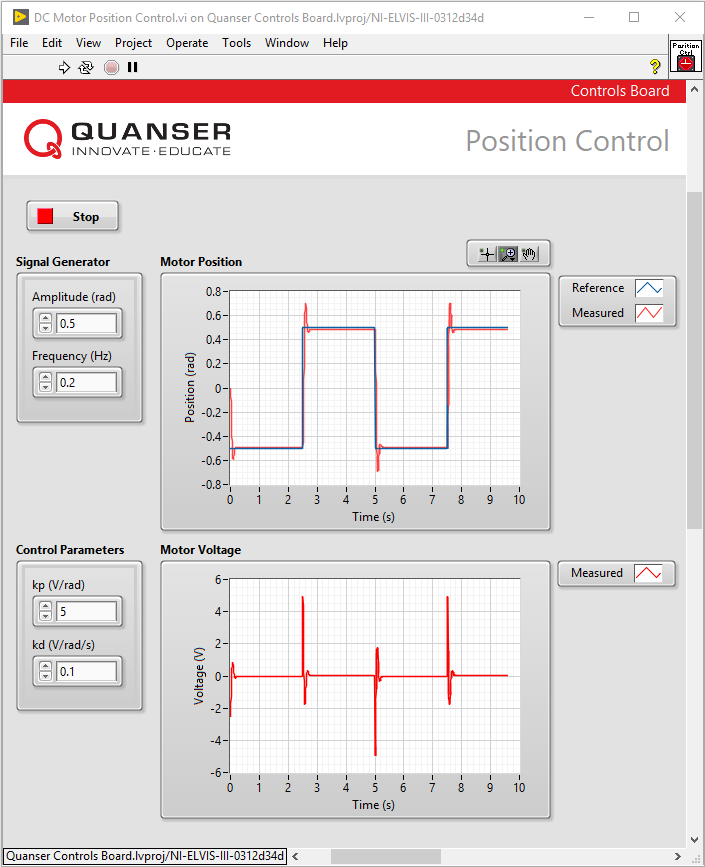


Figure1-2: Application for Qualitative PD speed control

1. Run the VI. The DC motor should begin rotating back and forth.
2. In the Signal Generator section of the front panel set:

* Amplitude (rad) to **0.50**
* Frequency (Hz) to **0.40**

1. In the Control Parameterssection, enter the control gains that you found in Step 2.
2. Capture the position response found in the **Motor Position** scope and control signal used in the **Motor Voltage** scope.
3. Measure the peak time and percentage overshoot of the measured position response.
4. If your response does not match the above overshoot and peak time specifications, try tuning your control gains until they are satisfied. Save the resulting response.
5. Stop the VI by clicking on the **Stop** button.

### 1.3 Analyze

1-1 What expressions for the proportional and derivative gains did you find in Step 1?

1-2 What control gains did you calculate would satisfy the performance requirements in Step 2?

1-3 Attach the response that you saved in Step 7.

1-4 What peak time and percentage overshoot did you measure in Step 8? Does the response that you measured meet the overshoot and peak time specifications given in Step 2 without saturating the motor (going beyond ±10 V)?

1-5 If your measured response did not meet the specifications, attach your tuned response, and comment on how you modified your controller to arrive at those results.

1-6 What effect does changing the two specifications, *ωn* and *ζ*, have on the generated control gains and measured response?

# Lab 4: Inverted Pendulum Control



Figure 0-1: A landing rocket can be viewed as a similar control challenge to an inverted pendulum

Balancing an inverted pendulum may seem like a purely academic challenge, but the control algorithm that is used is analogous to a wide variety of problems from Segway scooters to rockets. More broadly, the stabilization of unstable systems is a ubiquitous problem that requires much the same approach to controller design and development as the approaches covered in this lab. Complex system often require more complex controllers to achieve a desired level of performance, but ultimately the approaches presented in this lab represent an important collection of skills for any modern control systems engineer.

## Learning Objectives

After completing this lab, you should be able to complete the following activities.

1. Design a controller to balance an inverted pendulum using pole placement.
2. Design an optimized controller for an inverted pendulum.
3. Implement a hybrid swing-up controller for energy-based automatic inversion.

## Required Tools and Technology

|  |  |
| --- | --- |
| Platform: NI ELVIS III | * View User Manual   http://www.ni.com/en-us/support/model.ni-elvis-iii.html   * View Tutorials   <youtube link> |
| Hardware: Quanser Controls Board | * View User Manual   http://www.ni.com/en-us/support/model.quanser-controls-board-for-ni-elvis-iii.html |
| Software: LabVIEW Version 18.0 or Later  Toolkits and Modules:   * LabVIEW Real-Time Module * NI ELVIS III Toolkit * LabVIEW Control Design & Simulation | * Before downloading and installing software, refer to your professor or lab manager for information on your lab’s software licenses and infrastructure * Download & Install for NI ELVIS III   <http://www.ni.com/academic/download>   * View Tutorials   http://www.ni.com/academic/students/learn-labview/ |

## Expected Deliverables

In this lab, you will collect the following deliverables:

* Calculated control gains to meet the specifications
* Simulated response to the designed control gains
* Measured response on hardware including performance analysis
* Optimal controller design for the inverted pendulum
* Understanding of the effect of the design parameters on the controller
* Energy of the pendulum when inverted
* Control gains required to inverted the pendulum automatically

Your instructor may expect you complete a lab report. Refer to your instructor for specific requirements or templates.

## Section 1: Inverted Pendulum Control

### 1.1 Theory and Background

The design of a balance controller for an inverted pendulum begins with the definition of a linear state-space model. The standard linear state-space equations are

Equation 1-1

and

Equation 1-2

where *x* is the state, *u* is the control input, *A*, *B*, *C*, and *D* are state-space matrices. For the rotary pendulum system, the state and output are defined as

Equation 1-3

]

and

Equation 1-4

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

Equation 1-5

and

Equation 1-6

The velocities of the servo and pendulum angles can be computed in the digital controller using the encoder reading by taking the derivative and filtering the result though a low-pass filter.

The linear state-space model that represents the Quanser Controls Board inverted pendulum system is therefore

Equation 1-7

Your model may slightly differ based on the specific model parameters of your particular Quanser Controls Board, but this model should be representative of a general example.

#### Stability

The stability of a system can be determined from its poles:

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

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

Equation 1-8

det

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

#### Controllability

If the control input *u* of a system can take each state variable, *xi* where *i* = 1 … *n*, from an initial state to a final state then the system is controllable, otherwise it is uncontrollable.

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

Equation 1-9

equals the number of states in the system,

rank *.*

#### Companion Matrix

If (*A, B*) are controllable and *B* is n x 1, then *A* is similar to a companion matrix. Let the characteristic equation of *A* be

Equation 1-10

*.*

Then the companion matrices of *A* and *B* are

Equation 1-11

and

Equation 1-12

Define

where *T* is the controllability matrix defined in Equation 1-9 and

Then

and

#### Pole Placement

If (*A*,*B*) are controllable, then pole placement can be used to design the controller. Given the control law, the state-space in Equation 1-1 becomes

To illustrate how to design gain *K*, consider the following system

Equation 1-13

and

Equation 1-14

Note that *A* and *B* are already in the companion form. We want the closed-loop poles to be at *[-1 -2 -3]*. The desired characteristic equation is therefore

Equation 1-15

For the gain, apply control and get

The characteristic equation of is

Equation 1-16

Equating the coefficients between Equation 1-15 and the desired polynomial in Equation 1-15

|  |  |
| --- | --- |
|  |  |

Solving for the gains, we find that a gain of is required to move the poles to their desired location. We can generalize the procedure to design a gain *K* for a controllable *(A,B)* system as follows:

**Step 1** Find the companion matrices and. Compute.

**Step 2** Compute to assign the poles of to the desired locations. Applying the control law to the general system given in Equation 1-11,

Equation 1-17

**Step 3** Find to get the feedback gain for the original system *(A, B)*.

**Remark**It is important to do the conversion. Remember that *(A, B)* represents the actual system while the companion matrices and do not.

Desired Poles

The rotary inverted pendulum system has four poles. As depicted in Figure 1-1, poles *p1* and *p2* are the complex conjugate dominant poles and are chosen to satisfy the natural frequency, *ωn*, and damping ratio, *ζ*, specifications. If we let the conjugate poles be

Equation 1-18

and

Equation 1-19

Where and is the damped natural frequency. The remaining closed-loop poles,, and, are placed along the real-axis to the left of the dominant poles, as shown in Figure 1-1.

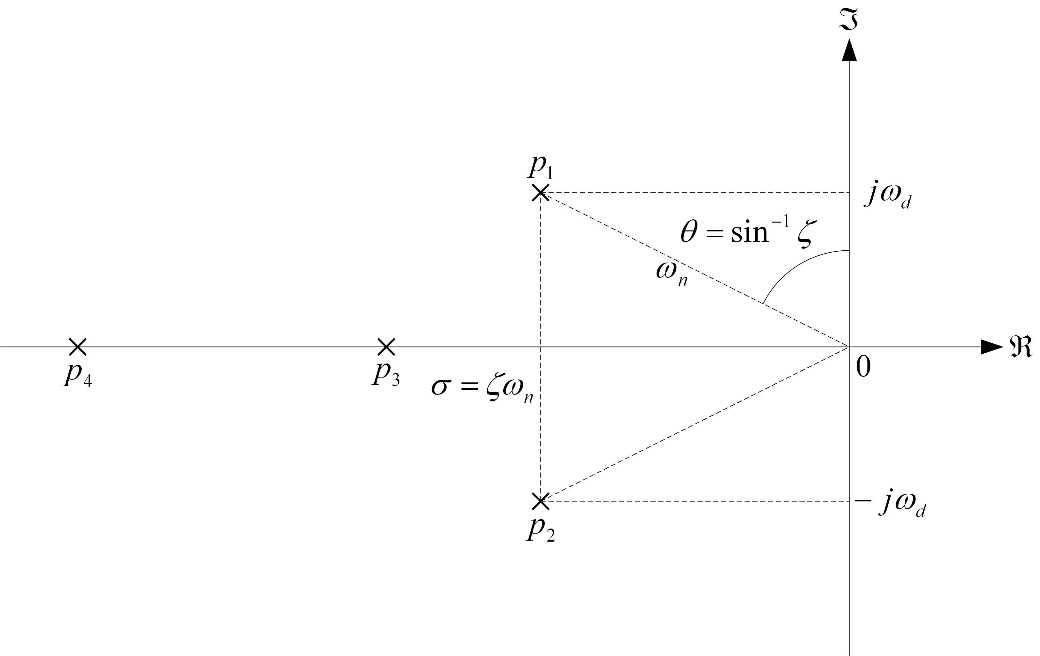


Figure 1-1: Desired closed-loop pole locations

#### Feedback Control

The feedback control loop that balances the rotary pendulum is illustrated in Figure 1-2. The reference state is defined

where is the desired rotary arm angle. The controller is

Note that if then, which is the control algorithm used in the pole-placement algorithm.

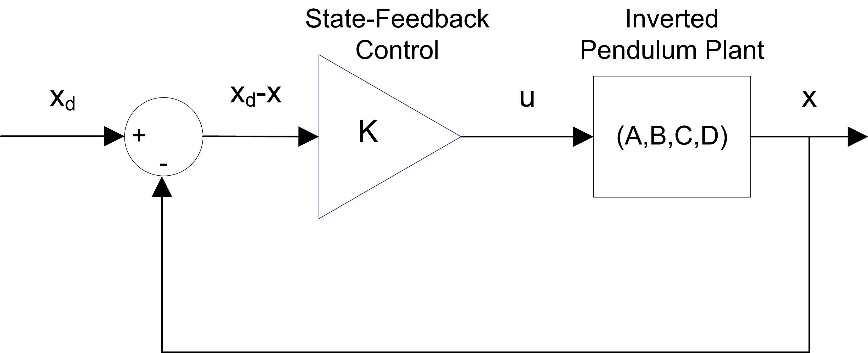


Figure1-2: State-feedback control loop

### 1.2 Implement

#### Control Design

1. The open-loop poles of the inverted pendulum are located at -16.17, 16.17, -0.005, and 0. Using the open-loop poles, find the characteristic equation of *A*.
2. Find the corresponding companion matrices and.
3. Find the location of the two dominant poles, *p*1 and *p*2, based on the following specifications

* ζ = **0.7**.
* ωn = **4** rad/s.

1. Give the desired characteristic equation if the other poles are placed at *p*3 = -30 and *p*4 = -40.
2. When applying the control to the companion form, it changes to. Find the gain that assigns the poles to their new desired location.
3. Open the project **Quanser Controls Board.lvproj**, and open **Balance Control Deisgn.vi**.

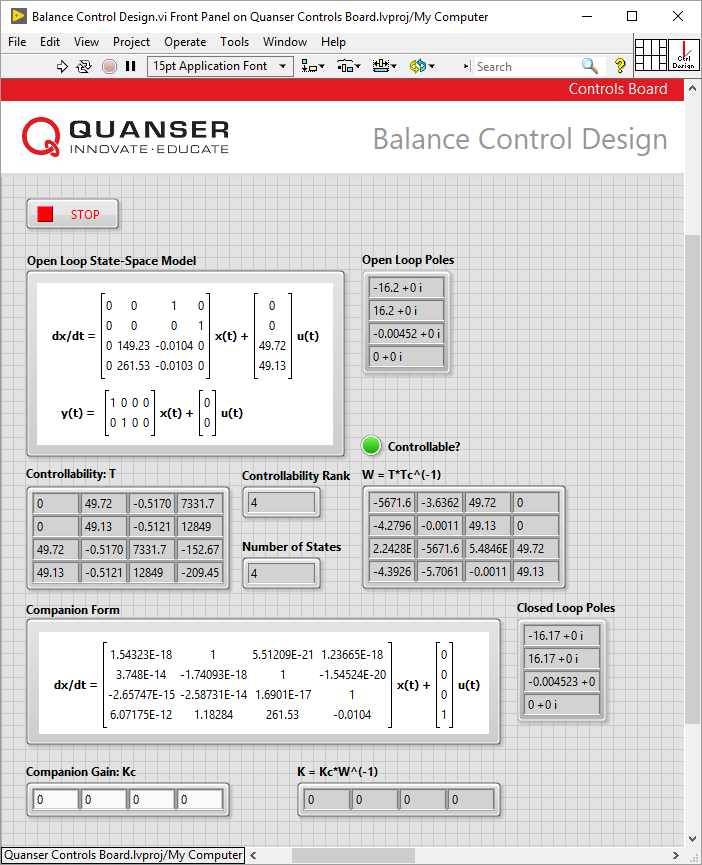
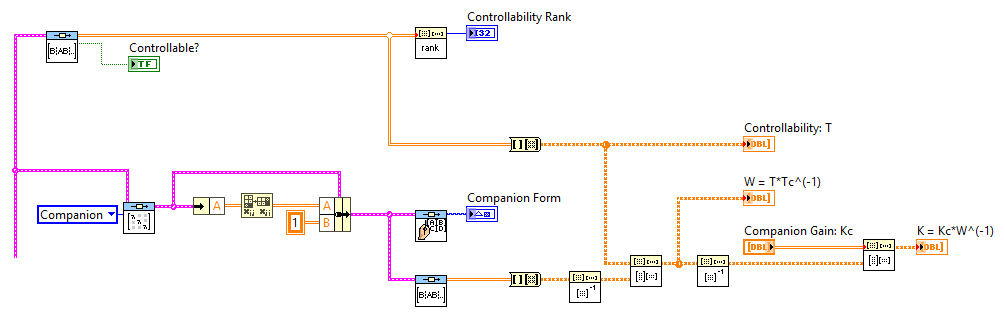


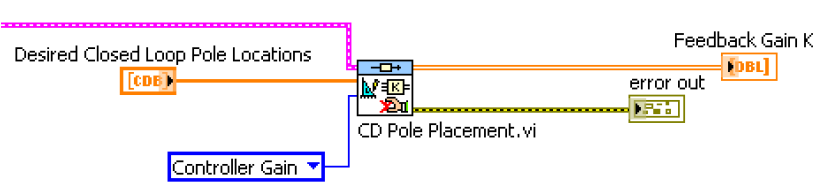
Figure1-2: Application for Qualitative PI speed control

1. Run the VI. The model should match the model given in Equation 1-7.
2. The companion matrices and are automatically found (denoted as *Ac* and *Bc*). In order to determine the appropriate gain *K*, the transformation matrix must be found. Open the block diagram, and locate the code segment used to calculate the controllability matrix *T*, the companion controllability matrix *Tc*, and the inverse of *Tc* shown below. Inspect the code segment in order to understand the functionality of the algorithm.



1. Enter the companion gainthat you found in Step 5 into the **Companion Gain: Kc** input on the front panel. Record the resultant feedback gain *K*.
2. Record the closed-loop poles of the system when using the gain *K*.

**Note:** The code that is used to determine the required matrices and feedback gain above can be replaced by a single operation shown below. This approach is a much simpler approach to implementing pole placement when using platforms that support common control algorithms.



1. Stop the VI by clicking on the **Stop** button.

#### Simulation

1. Open the project **Quanser Controls Board.lvproj**, and open **Balance Control Simulation.vi**.

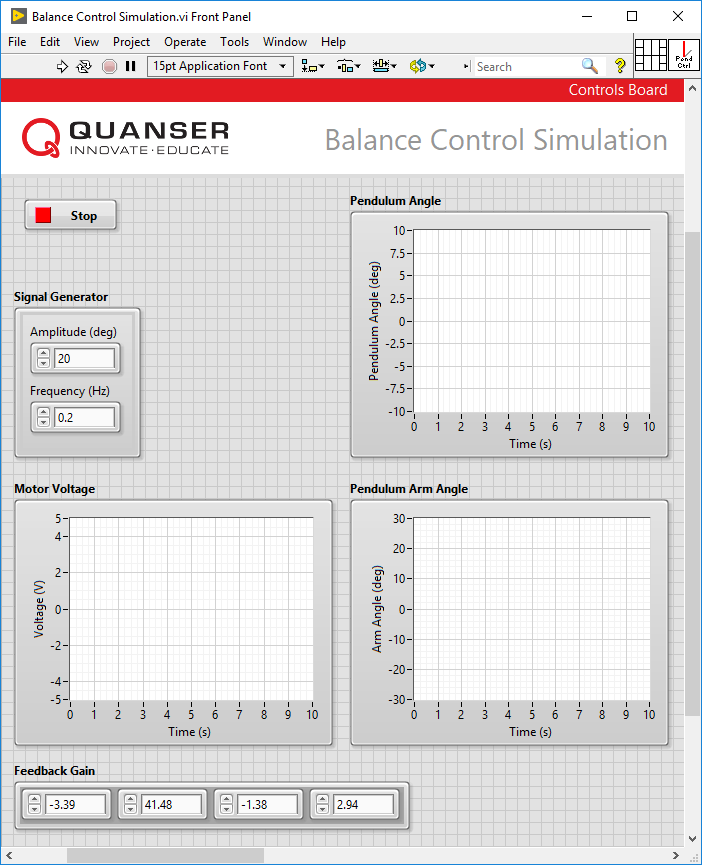


Figure1-3: Application for Inverted Pendulum Control Simulation

1. Enter the gain that you found in the previous section into the **Feedback Gain** input.
2. Record the simulated response of your designed gain.
3. Given the additional implementation specifications:

* Maximum pendulum deflection < **15**°.
* Maximum control effort < **5** V.

Measure the simulated response and determine if the additional specifications listed are met.

1. Stop the VI by clicking on the **Stop** button.

#### Inverted Pendulum Control

1. Open the project **Quanser Controls Board.lvproj**, and open **Balance Control.vi**.
2. Enter the feedback gain, *K*, from the previous section.
3. Ensure the pendulum is in the hanging down position and is motionless. Run the VI and manually bring up the pendulum to its upright, vertical position. You should feel the voltage kick-in when it is within the range where the balance control engages.
4. Once it is balanced, introduce a ±20 degree rotary arm command by setting **Amplitude (deg)** to 20 in the VI. The response should look similar to your simulation.
5. Record the measured rotary pendulum responses.
6. Measure the pendulum deflection and voltage used. Are the specifications given in Step 4 of the previous section satisfied for the implementation?
7. Stop the VI by clicking on the **Stop** button.

### 1.3 Analyze

1-1 Based on the location of the open-loop poles, is the system stable, marginally stable, or unstable? Does that make sense?

1-2 What is the characteristic equation of *A*?

1-3 What are the corresponding companion matrices and ?

1-4 What are the locations of the two dominant poles, *p1* and *p2*?

1-5 What is the gain that assigns the poles to their new desired location?

1-6 Based on the number of states and rank of the controllability matrix shown in the Control Design VI, is the system controllable?

1-7 Describe the functionality of the code that is used to determine the controllability matrix *T*, the companion controllability matrix, the inverse of *Tc*, and the *W* matrix*.*

**Hint:** Use *Context Help* (ctrl+H) to get a description of each block.

1-8 Enter the companion gain you found in the pre-lab into the **Companion Gain (Kc)** input box on the VI front panel. Run the VI again to calculate the feedback gain *K* and record its value.

1-9 Record the closed-loop poles of the system when using the gain *K*. Have the poles been placed in their desired locations?

1-10 What is the simulated response of your designed gain?

1-11 What are the specifications of the simulated response? Are the additional specifications listed met?

1-12 What response was recorded in Step 20?

1-13 What is the pendulum deflection and voltage used in the response? Are the specifications satisfied?

## Section 2: Optimal Control of an Inverted Pendulum

### 2.1 Theory and Background

Linear Quadratic Regulator (LQR) theory is a technique that is ideally suited for finding the optimal parameters of the pendulum balance controller. Given that the equations of motion of the system can be described in the form

where *A* and *B* are the state and input system matrices, respectively, the LQR algorithm computes a control law *u* such that the performance criterion or cost function

Equation 2-1

is minimized. The design matrices *Q* and *R* hold the penalties on the deviations of state variables from their set-point and the control actions, respectively. When an element of *Q* is increased, therefore, the cost function increases the penalty associated with any deviations from the desired set-point of that state variable, and thus the specific control gain will be larger. When the values of the *R* matrix are increased, a larger penalty is applied to the aggressiveness of the control action, and the control gains are uniformly decreased.

In our case the state vector *x* is defined

Equation 2-2

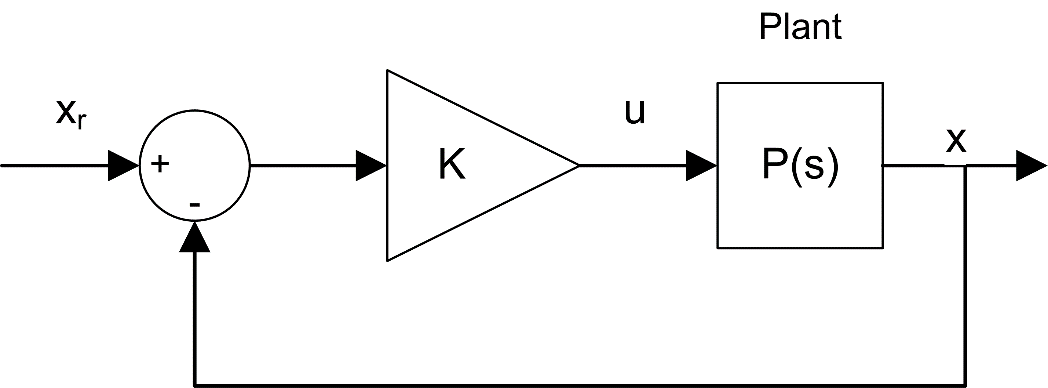


Figure 2-1: Block diagram of balance state-feedback control for rotary pendulum

Since there is only one control variable, *R* is a scalar. The reference signal *x*ref is set to, and the control strategy used to minimize the cost function *J* is thus given by

Equation 2-3

This control law is state-feedback control, and is illustrated in Figure 2-1. It is equivalent to PD control.

### 2.2 Implement

LQR design theory has built in support in LabVIEW™ using the Control Design & Simulation module. Given a model of the system in state-space form (with system matrices *A* and *B*) and the weighting matrices *Q* and *R*, the LQR function in the Control Design Toolkit automatically minimizes the cost function Equation 2-1 and computes the optimal feedback control gain automatically.

#### LQR Control Design

In this experiment, the state-space model is already available. Therefore, the effect of changing the *Q* weighting matrix while *R* is fixed to 1 on the cost function *J* will be explored.

1. Open the project **Quanser Controls Board.lvproj**, and open **Optimal Control Design.vi**.

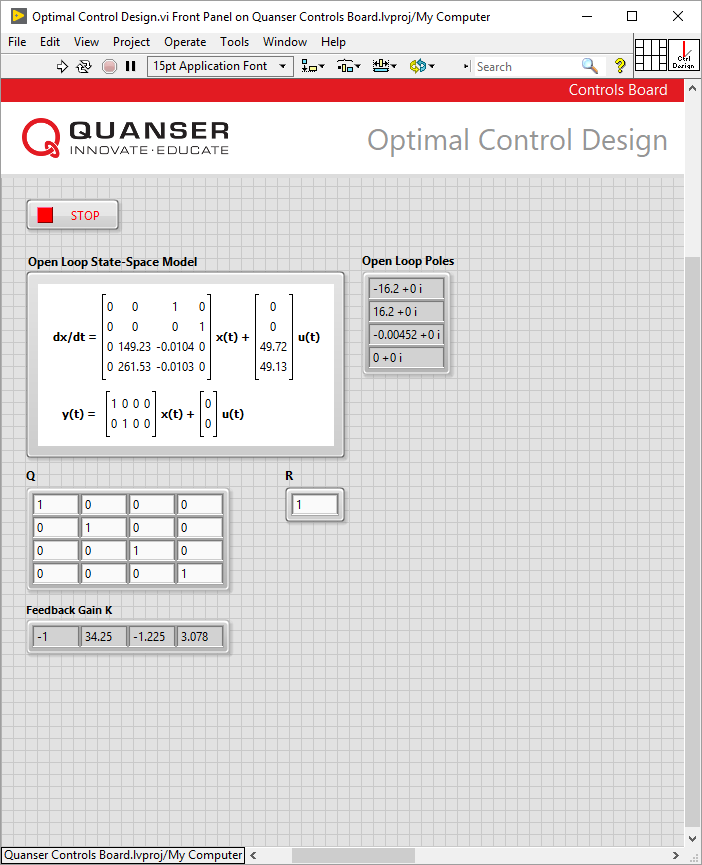


Figure 2-2: VI used to design the balance controller using LQR

1. Ensure that the weighting matrices are set to

and *.*

1. Record the gain *K* that is generated by the VI.
2. Stop the VI, and repeat the process using the following weighting matrices

and *.*

1. Record the new gain.
2. Stop the VI by clicking on the **Stop** button.

#### LQR Balance Control

1. Once again open the **Balance Control.vi** shown in Figure 2-3.

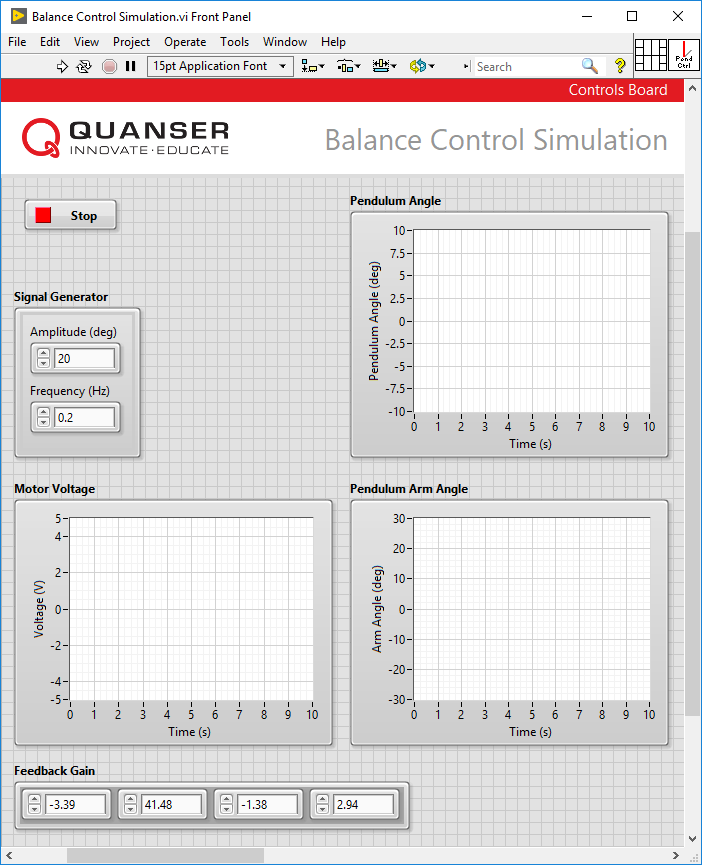


Figure 2-3: VI used to balance the pendulum using LQR

1. Make sure that the feedback gains are set to those recorded in Step 3 of the LQR Control Design section.
2. Run the VI.
3. Manually rotate the pendulum in the upright position until the controller engages.
4. Once the pendulum is balanced, set the **Amplitude (deg)** control to 30 to make the arm angle go between ±30°, and set the **Frequency (Hz)** control to 0.1.
5. The scopes should look similar to those shown in Figure 2-4. Record the response of the rotary arm, pendulum, and controller voltage.

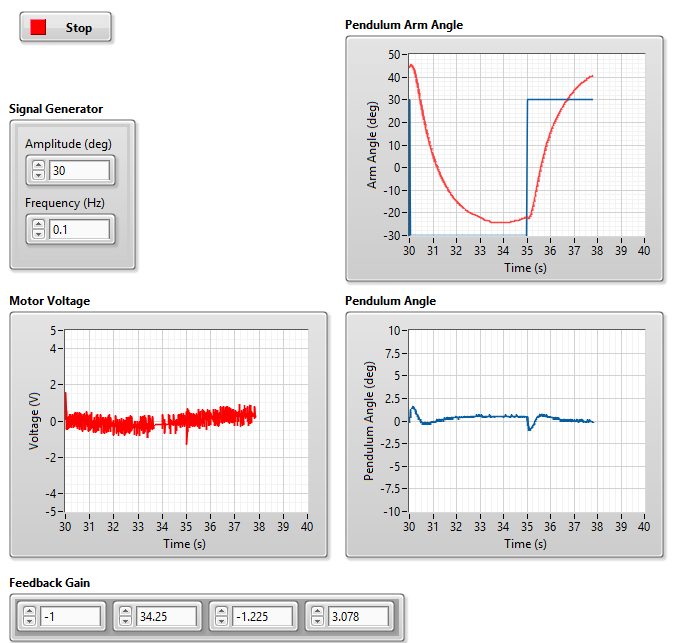


Figure 2-4: Rotary pendulum response

1. Stop the VI. Update the feedback gains with the values in Step 5 of the LQR Control Design section and repeat your analysis.
2. Finally, using **Optimal Control Design.vi** adjust the value of *R* to 0.8 and record a new set of feedback gains. Update the feedback gains in the balance control VI with these new gains, and observe and record the resultant gain and response.
3. Use **Optimal Control Design.vi** to generate a set of new feedback gains to meet the following maximum control specifications:
   * Pendulum deflection: ±5°
   * Overshot: 20%
   * Peak time: 0.8 s

Describe your experimental procedure to find the necessary control gain.  
Note: Assume that the value of *R* can remain set to **1** during tuning.

1. List the resulting LQR *Q* matrix and control gain *K* used to yield the desired results. Record the responses using this new control gain and briefly outline how the response changed.
2. Stop the VI by clicking on the **Stop** button.

### 2.3 Analyze

2-1 What gain was generated in Step 3?

2-2 What gain was generated in Step 5?

2-3 How does changing *q*11 affect the generated control gain? Based on the description of LQR in the background section, is this what you expected?

2-4 What is the response of the system in Step 5?

2-5 Examine and describe the change in the system in Step 13.

2-6 What is the effect of reducing the value of *R* on the response of the system in Step 8?

2-7 Adjust the diagonal elements of *Q* matrix according to the specifications in Step 15. Describe your experimental procedure to find the necessary control gain.

2-8 List the resulting LQR *Q* matrix and control gain *K* used to yield the desired results. Attach the responses using this new control gain and briefly outline how the response changed.

## Section 3: Swing-Up Hybrid Control

### 3.1 Theory and Background

#### Energy Control

In theory, if the arm angle of the pendulum system is kept constant and the pendulum is given an initial perturbation, the pendulum will keep on swinging with constant amplitude. The idea of energy control is based on the preservation of energy in ideal systems: The sum of kinetic and potential energy is constant. However, friction will damp the oscillation in practice and the overall system energy will not be constant. It is possible to measure the loss of energy with respect to the pivot acceleration, which in turn can be used to find a controller to swing up the pendulum.

The dynamics of the pendulum can be redefined in terms of the pivot acceleration, *u*, as

Equation 3-1

Here, *u* is the linear acceleration of the pendulum.

The potential energy of the pendulum is

and the kinetic energy is

The pendulum angle, *α*, and the lengths of the pendulum are illustrated in the free body diagram in Figure 3-1.

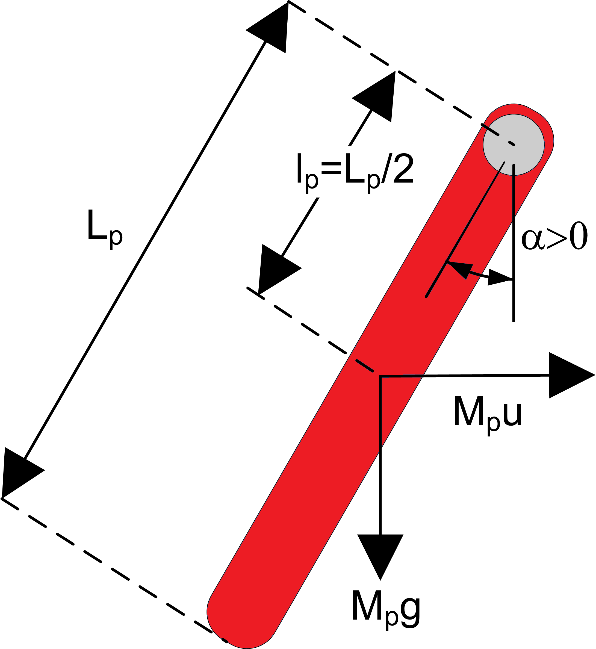


Figure 3-1: Rotary pendulum response

The potential energy is zero when the pendulum is at rest at and equals when the pendulum is upright at . The sum of the potential and kinetic energy of the pendulum is

Equation 3-2

Differentiating Equation 3-2 yields

Equation 3-3

Using Equation 3-1, the terms can be rearranged as

which eventually yields

Since the acceleration of the pivot is proportional to current driving the arm motor and thus also proportional to the drive voltage, it is possible to control the energy of the pendulum with the proportional control law

Equation 3-4

By setting the reference energy to the pendulum potential energy (), the control law will swing the link to its upright position. Notice that the control law is nonlinear because the proportional gain depends on the cosine of the pendulum angle *α*. Further, the control changes sign when changes sign and when the angle is ±90 degrees.

For the system energy to change quickly, the magnitude of the control signal must be large. As a result the following swing-up controller is implemented in the controller as

Equation 3-5

where is a tunable control gain and the function saturates the control signal at the maximum acceleration of the pendulum pivot,. The expression is used to enable faster control switching.

#### Hybrid Control

The energy swing-up control in Equation 3-4 (or Equation 3-5) can be combined with the balancing control law from the Balance Control Lab to obtain a control law that swings up the pendulum and then balances it.

Similarly, the balance control is to be enabled when the pendulum is within ±20 degrees. When it is not enabled, the swing-up control is engaged. Thus the switching can be described mathematically by

Equation 3-6

### 3.2 Implement

The VI shown below implements the algorithm described in Section 3.1 to swing-up and balance the pendulum.

#### Energy Control

1. Open the project **Quanser Controls Board.lvproj**, and open **Swing-up Control.vi**.

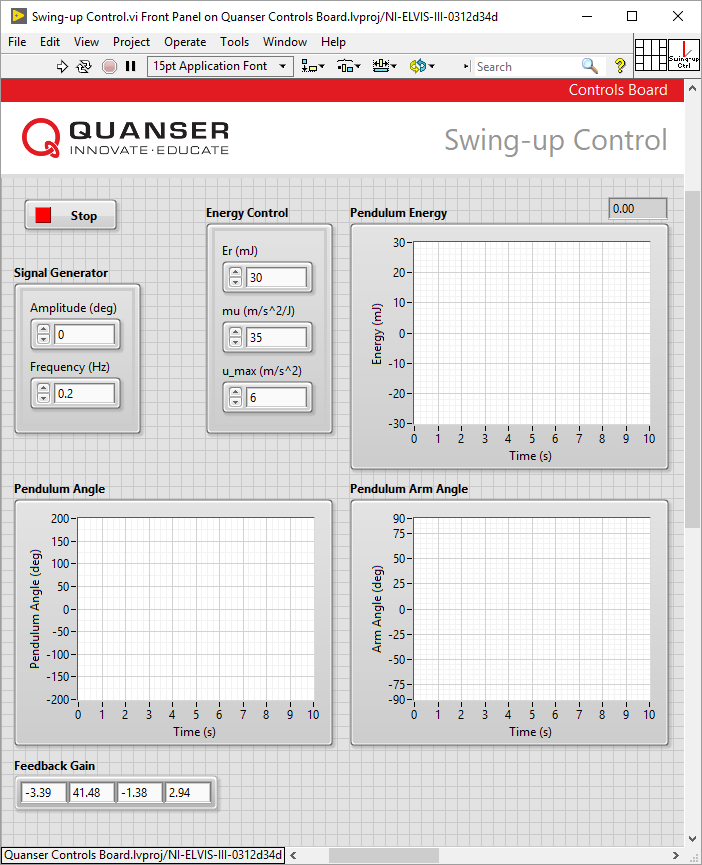


Figure 3-2: Application for hybrid swing-up control

1. Ensure that the swing-up controller is disabled by setting the gain, *μ*, to 0.
2. Run the VI.
3. Manually rotate the pendulum to different angles and examine the pendulum energy shown in the **Pendulum Angle (deg)** and **Pendulum Energy (mJ)** charts.
4. Record the energy when the pendulum is being balanced upright.
5. Click on the **Stop** button to return the pendulum to the downward gantry position.

#### Hybrid Swing-Up Control

1. Set the swing-up control parameters to the following values:

* Er (mJ) = **10**
* mu (m/s2/J) = **50**
* u\_max (m/s2) = **6**

1. Run the VI
2. If the pendulum is not moving, gently perturb the pendulum with your hand from the downward position.
3. Vary the reference energy, *Er*, between 10.0 mJ and 20.0 mJ. As it is changed, examine the pendulum angle and energy response in the **Pendulum Angle (deg)** and the **Pendulum Energy (mJ)**charts and the control signal in the **Motor Voltage (V)** chart.
4. Fix the value of *Er* at 20.0 mJ and vary the swing-up control gain *μ* between 20 and 60 m/s2/J. Observe any changes in the performance of the energy controller.
5. Stop the VI by clicking on the **Stop** button.
6. Set the swing-up control parameters to the following values:

* mu (m/s2/J) = **20**
* u\_max (m/s2) = **6**

1. Based on your observations from the previous section, enter an appropriate value for the reference energy parameter, *Er*.
2. Make sure that the pendulum is hanging down motionless and the encoder cable is not interfering with the pendulum.
3. Run the VI.
4. The pendulum should begin going back and forth. If not, perturb the pendulum lightly with your hand. Click on the **Stop** button if the pendulum appears to be unstable.
5. Gradually, in increments of 5 m/s2/J, increase the swing-up gain *μ* until the pendulum swings up to the vertical position. Capture the response of the pendulum, pendulum energy, and motor voltage. Be sure to record the swing-up gain that was required.
6. Stop the VI by clicking on the **Stop** button.

### 3.3 Analyze

3-1 What do you notice about the energy of the pendulum when it is moved to various angles? Does the measured energy when the pendulum is balanced upright make sense according to the equations in Section 3.1?

3-2 Attach the responses specified in Step 10 showing how changing the reference energy affects the system.

3-3 Describe how the changing the swing-up control gain in Step 11 affected the performance of the energy controller.

3-4 What did you set the reference energy to in Step 5?

3-5 Attach your responses from Step 18. What gain was required?