Hardware Interface Laboratory Experiments

Hardware Interface Laboratory

***Note: Before performing experiments in this section, please study Appendix D—LabVIEW Tutorial, including the section discussing myDAQ.* When an experiment indicates a provided file, the file is obtained at www.wiley.com/college/nise.**

Experiment 2.4 Programming with LabVIEW Part 1

Objectives

1. To learn how to program LabVIEW, Part 1
2. To learn how to write basic LabVIEW programs and understand LabVIEW flow

Material Required

Computer with LabVIEW Installed

Prelab

Go to the website http://www.learnni.com/getting-started/. Complete modules 0-7.

Lab

1. Write a LabVIEW program that executes an equivalent of the following C-like code, where x is an input and y is an output (Formula Nodes are not allowed):

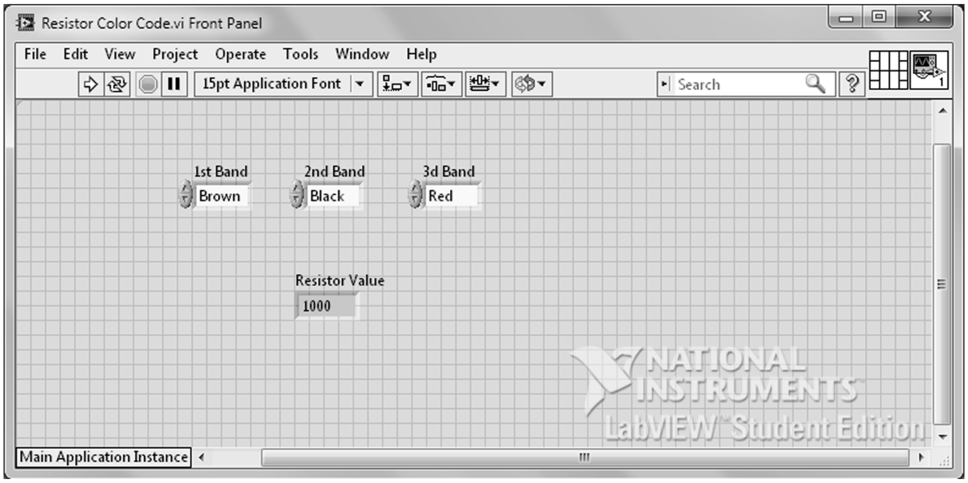
```
if(abs(x)<0.1)
        y=1;
else
        if(x>=0)
             y=0;
        else
             y=2;
```

   Run your program for the following inputs: $x = 0.05, -0.05, 1, -1$.

2. Write a LabVIEW program that receives three colors representing a resistor's value and returns the numeric resistor value in ohms. Your interface should be similar to the one shown in Figure P2.42. The third band should include silver and gold colors.

Run your program at least for the following inputs:

Red Red Black
Brown Black Orange
Orange White Gold



Resistor Color Code.vi Front Panel

1st Band: Brown
2nd Band: Black
3d Band: Red
Resistor Value: 1000

FIGURE P2.42

Experiment 2.5 Programming with LabVIEW Part 2

Objectives

1. To learn how to program LabVIEW, Part 2
2. To learn how to use loops, do basic math inside loops, and graph numerical information using LabVIEW

Material Required

Computer with LabVIEW Installed

Prelab

Go to the website http://www.learnni.com/getting-started/. Complete modules 8-10.

Lab

1. It is well known that $1 + x + x^2 + x^3 + \cdots = \dfrac{1}{1 - x}$ when $|x| < 1$.

   Write a LabVIEW program that takes as an input a value for x, and a number of iterations. The program will use a loop to calculate the sum of the geometric series for the specified number of operations. It will also calculate the closed-form expression for the series. The program will display the two results, and will also display the absolute error of the difference.

   Demonstrate your program with $x = 0.5$ and 3, 10, and 200 iterations.

2. Write a LabVIEW program that generates a 50% duty-cycle square-wave signal between 0 and $X_{\max}$ volts, where $X_{\max} < 10$ V, with a nonzero variable frequency. The amplitude and frequency will be inputs. Display the waveform on a waveform chart, which will be an output. In this part you are not allowed to use the LabVIEW-provided function generation blocks.

   Demonstrate this program for amplitudes of 1, 5, and 10 V and for 1 Hz and 5 Hz.

Experiment 2.6 MyDAQ Programming

Objectives To become familiar with the data acquisition and signal generation capabilities of myDAQ

Material Required Computer with LabVIEW installed, and myDAQ

Files Provided at www.wiley.com/college/nise

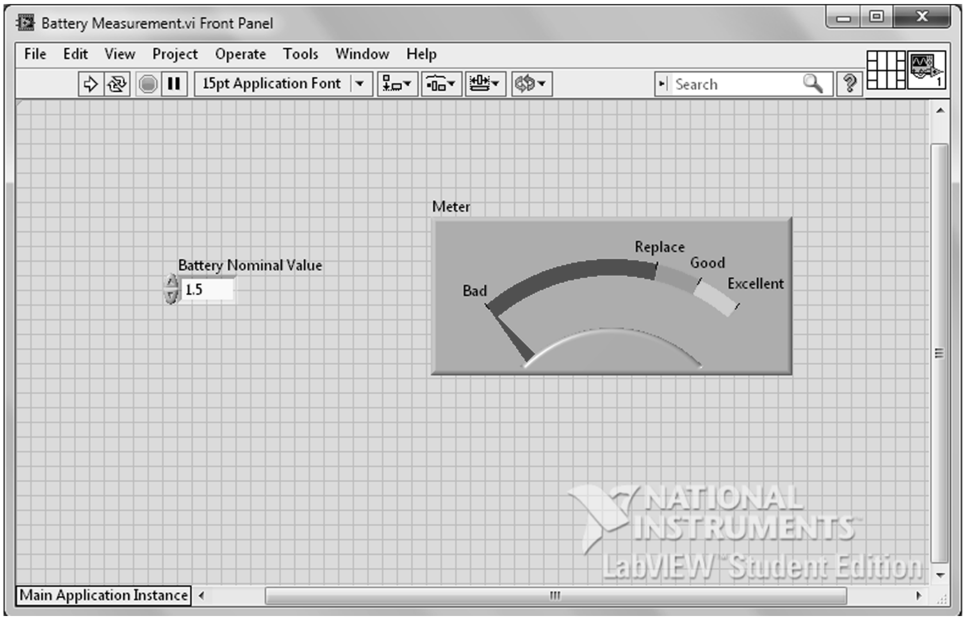
Battery Meter.ctl

Prelab

Go to the website https://decibel.ni.com/content/docs/DOC-11624. Go over Unit 4—DAQ: Lesson 1. Then go over the measuring voltage tutorial in http://zone.ni.com/devzone/cda/epd/p/id/6436.

Lab

1. Write a battery-tester program using LabVIEW and myDAQ as an acquisition device. The battery tester should work for three nominal values of batteries: 1.5 V, 6 V, and 9 V. The batteries are considered dead for voltage values 20% or under the nominal. Between 20% and the nominal value the batteries are in a warning area, and for values above the nominal the batteries are OK. Your interface should be similar to the one shown Figure P2.43. A custom control accepting inputs from 0 to 120 has been created under the name **Battery Meter.ctl**.


Battery Measurement.vi Front Panel
File Edit View Project Operate Tools Window Help
15pt Application Font
Search
Meter
Replace
Good
Excellent
Bad
NATIONAL INSTRUMENTS
LabVIEW Student Edition
Main Application Instance

Battery Nominal Value 1.5

FIGURE P2.43

2. Use the LabVIEW program that you wrote in Experiment 2.5 to generate a 50% duty-cycle square-wave signal. Output your signal through one of the myDAQ's analog channels and read the signal using the myDAQ oscilloscope function (available from myDAQ file: *NI ELVISmx Instrument Launcher*) to verify the generated signal. Print two examples using the scope's automatic measurements.

Hardware Interface Laboratory

Experiment 4.5 Open-Loop Speed Control of a Motor

Objectives

To control the speed of a motor in open-loop fashion and verify the functions of the motor control setup as preparation for future experiments.

Material Required

Computer with LabVIEW Installed; myDAQ; dc brushed gearmotor with Hall Sensor quadrature encoder (–10 V to +10 V normal operation range); and motor control chip BA6886N or a transistor circuit substitute. (Note: For simplicity the input to the motor will be analog. PWM will be avoided as it adds an additional layer of complexity to these experiments. Plan accordingly if you decide to substitute the motor control chip.)

File Provided at www.wiley.com/college/nise

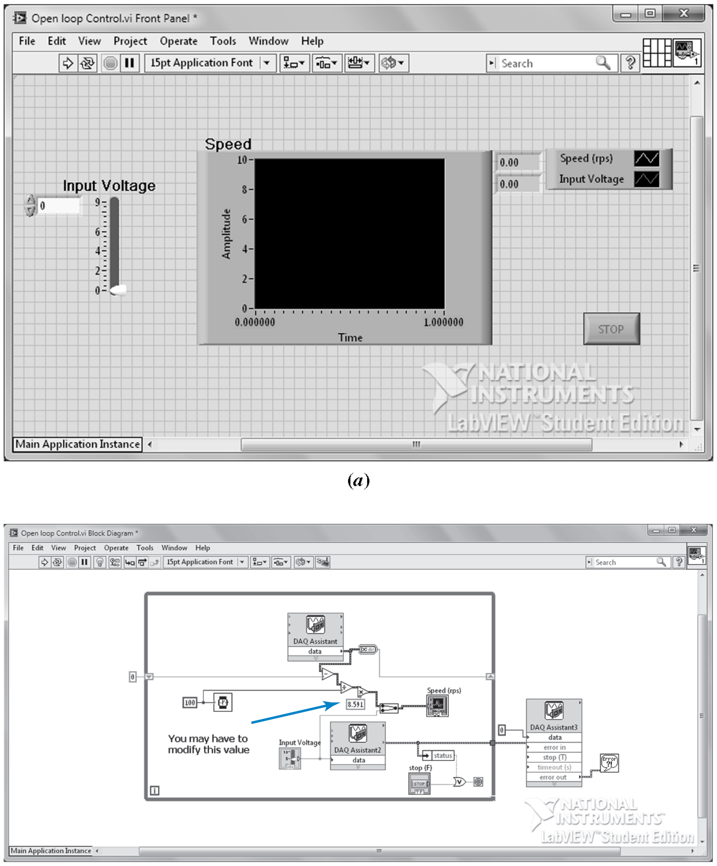
Open Loop Control.vi

Prelab

Plan how you will wire your motor to the breadboard. A possibility is to solder a header to six matching color wires that will allow you to connect and disconnect the motor from the myDAQ in an efficient manner. You can also solder wires to the motor's cables.

Lab

Software: The front panel for the Open Loop Control VI is shown in Figure P4.28(*a*). The input to the system is the voltage applied to the motor. The output is the motor speed in revolutions per second (rps). The corresponding block diagram is shown in Figure P4.28(*b*).


Open loop Control.vi Front Panel *
File Edit View Project Operate Tools Window Help
15pt Application Font
Search
Speed
Input Voltage
Amplitude
Time
0.000000
1.000000
Speed (rps)
Input Voltage
0.00
0.00
STOP
Main Application Instance
NATIONAL INSTRUMENTS LabVIEW Student Edition
Open loop Control.vi Block Diagram *
DAQ Assistant
data
100
8.591
Speed (rps)
You may have to modify this value
Input Voltage
DAQ Assistant2
status
stop (F)
DAQ Assistant3
error in
stop (T)
timeout (s)
error out
Main Application Instance

(*a*)

(*b*)

FIGURE P4.28 Open loop control.vi: **a.** Front Panel; **b.** Block Diagram

Note the value indicated by the blue arrow in Figure P4.28(*b*). In order to get a meaningful reading for the speed of the motor, this value needs to be modified depending on the gear ratio of your motor and the counts of your encoder. To understand how this value is calculated, note that the DAQ Assistant block on top of the diagram reads the encoder input from the myDAQ. One would reasonably assume that the frequency of this signal is proportional to the speed of the motor, which is theoretically true. However, at very low speeds the DAQ assistant "times out" and fails to provide a reading if frequency is measured directly. To avoid this problem, a different method is used to calculate the signal frequency. The DAQ assistant measures the rising edges of the encoder signal every 100 msec and subtracts that number from the ones accumulated during the previous 100 msec period. The frequency of the encoder signal (in edges/msec) is found by dividing the value of this subtraction by the period (100 msec). See the Block Diagram to understand how this algorithm was implemented.

We use an example to illustrate the calculation of the constant pointed to with the red arrow. If a 9.7:1 gear ratio is used with a 48 CPR encoder, with each revolution the encoder will generate a total of 48 edges in each of the encoder channels. Using one channel only and positive edges, there are 12 positive edges/rev of the motor shaft. The total number of counts (positive edges) generated by each revolution of the external shaft is $9.7 \times 12 = 116.4$ positive edges/rev.

In order to find the rotational speed, the frequency (edges/msec) of the signal is divided by the total number of counts generated by the external shaft, adjusting the units for time from msec to sec: Rotational Speed (rps) $=$ Freq $\times$ 1000/(9.7 $\times$ 12) $=$ Freq $\times$ 8.591. This value has to be set as illustrated in the Block Diagram.

The DAQ Assistant2 block transmits the voltage from the control slider to the myDAQ and to the motor control chip. The DAQ Assistant3 block makes sure that the output to the chip is zeroed when the VI terminates.

Hardware: Connect the myDAQ, the motor, and the motor controller as shown in Figure P4.29.

Procedure:

1. Verify the operation of your circuit by running the VI and changing the position of the slider. If everything is correct, the motor speed will vary as the slider's position changes.
2. Verify that you are using the correct scaling factor for your motor by setting your motor to rotate at 0.5 rps. Count the number of rotations in the shaft of the motor over 10 sec using a stopwatch. Repeat by setting the rotational speed at 1 rps. Your measurements must be consistent.
3. Perform the following measurements moving the slider:
   a. Increase the voltage starting at zero and record the minimum voltage for the motor to start rotating.
   b. Starting the slider at a rotating speed, reduce the voltage until the motor stops. Record this voltage.
   Are these values equal? These values are important and will be used in future labs. Keep them in a safe place so that you don't have to repeat these measurements again.
4. Make a graph where the *x*-axis is the input voltage, and the *y*-axis is the speed in rps. Include the results in Part 3.
5. Draw a functional block diagram of the system (similar to Chapter 1), labeling each of the components in the diagram.

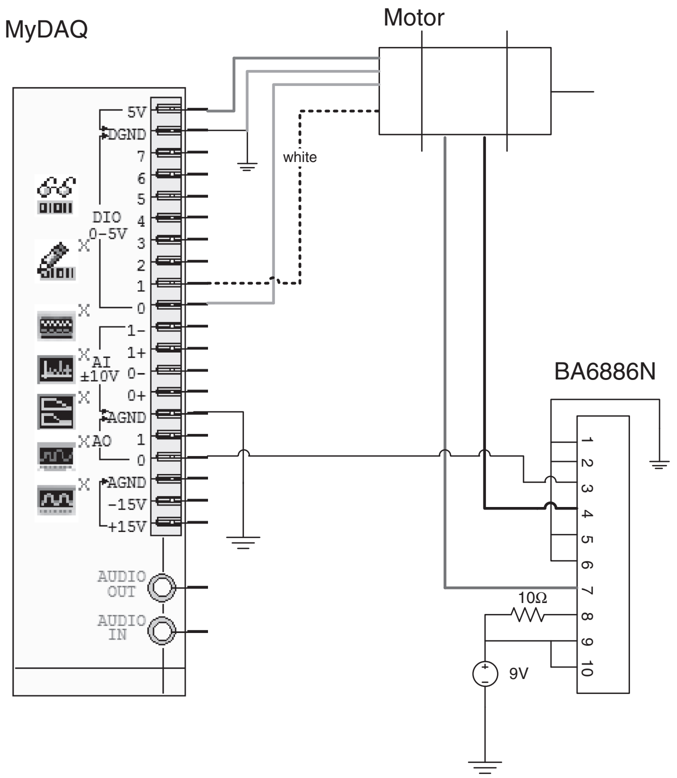


FIGURE P4.29 Wiring diagram[1]

6. The circuit and the VI above allow the motor to rotate in one direction only. Modify the VI and the circuit so that the motor direction and speed can be controlled from the VI.

   Note that the reference input to the chip can only accept positive voltage values. The motor control data sheet indicates that direction of rotation must be changed by flipping the logical values of Pin 2 and Pin 10 on the motor control chip. However, a careful reading of the data sheet indicates that there must be an instant of time (of unspecified duration) in which both inputs must be False before switching direction. You may want to use a LabVIEW ring to simulate a three-way switch. Use two of the selections of the ring to control the motor's rotation direction. The third selection in the ring should provide low inputs to the motor controller logical inputs to be able to stop the motor before switching direction.

[1] MyDAQ right slot shown on left is taken from Multisim program module NI myDAQ design and also reproduced in White-Paper 11423, Figure 2. Both Multisim and the White Paper are from National Instruments.

Experiment 4.6 Transfer Function Identification

Objective To identify the transfer function of a motor from voltage input to angular motor speed using myDAQ and LabView.

Material Required Computer with LabVIEW Installed; myDAQ; dc brushed gearmotor with Hall sensor quadrature encoder (–10 to +10 V normal operation range); and motor control chip B6886N (or a transistor circuit substitute).

File Provided at www.wiley.com/college/nise

Plant Identification 2.vi

Prelab Answer the following questions:

1. What is the unit-step response of a system with transfer function $G(s) = \dfrac{K}{s\tau + 1}$, where K and τ are constants > 0?
2. Make a hand-sketch of the response of the unit-step response of the system in Part 1.
3. What is the value of the step response of the system in Part 1 when $t = \tau$?
4. Find or derive the expression for the transfer function from voltage to angular speed of an unloaded permanent magnet dc motor. Compare this transfer function to the first-order system in Part 1.

Lab Connect the myDAQ, the motor, and the motor controller as shown in Figure P4.30. This setup is identical to the one that was used initially in Experiment 4.5, except that we

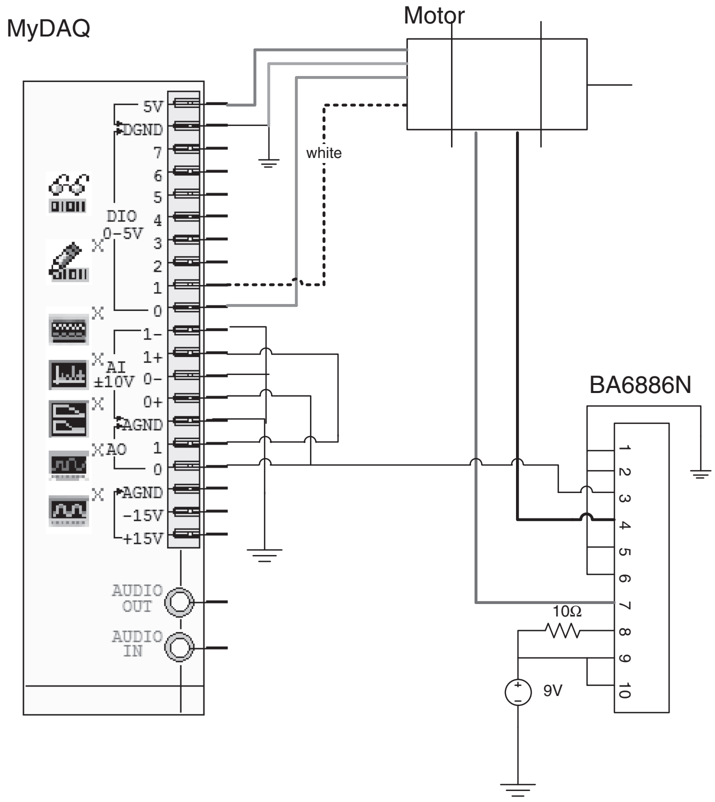
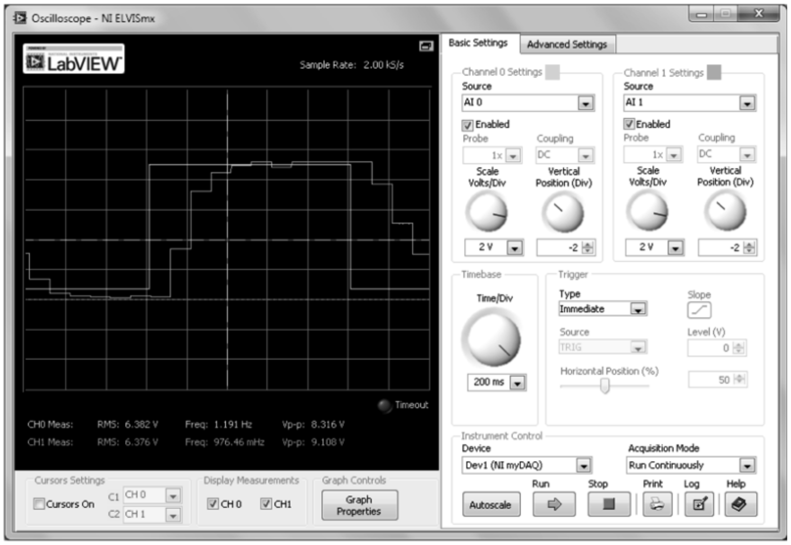


FIGURE P4.30 Wiring diagram[2]

[2] MyDAQ right slot shown on left is taken from Multisim program module NI myDAQ design and also reproduced in White-Paper 11423, Figure 2. Both Multisim and the White Paper are from National Instruments.



Oscilloscope - NI ELVISmx

Sample Rate: 2.00 kS/s

CH0 Meas: RMS: 6.382 V Freq: 1.191 Hz Vp-p: 8.316 V

CH1 Meas: RMS: 6.376 V Freq: 976.46 mHz Vp-p: 9.108 V

Channel 0 Settings: Source: AI 0; ☑ Enabled; Probe: 1x; Coupling: DC; Scale Volts/Div: 2 V; Vertical Position (Div): -2

Channel 1 Settings: Source: AI 1; ☑ Enabled; Probe: 1x; Coupling: DC; Scale Volts/Div: 2 V; Vertical Position (Div): -2

Timebase: Time/Div: 200 ms

Trigger: Type: Immediate; Source: TRIG; Level (V): 0; Horizontal Position (%): 50

Instrument Control: Device: Dev1 (NI myDAQ); Acquisition Mode: Run Continuously

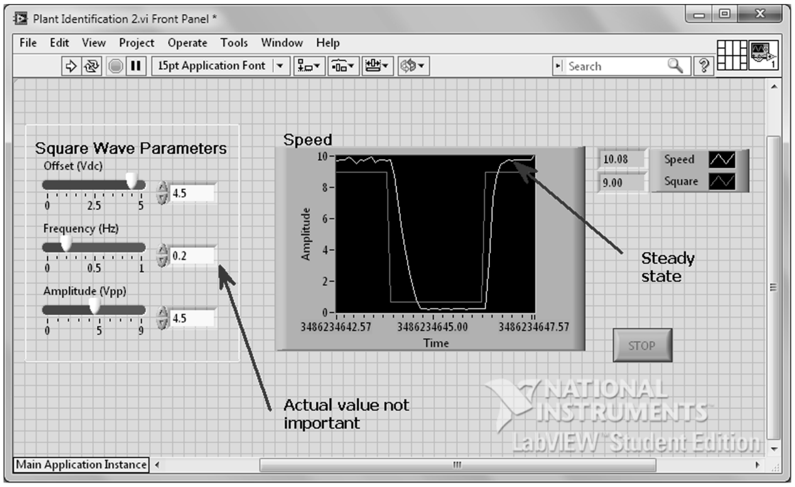
Cursors Settings: □ Cursors On; C1: CH 0; C2: CH 1

Display Measurements: ☑ CH 0 ☑ CH1

FIGURE P4.31 LabVIEW Oscilloscope—NI ELVISmx

have connected the two analog input channels to the two analog output channels. This will allow us to use the myDAQ oscilloscope for measurements. If you decide to use an external oscilloscope, these connections are not necessary.

1. Open the Oscilloscope and Plant Identification 2.vi shown in Figures P4.31 and P4.32, respectively. You can also choose to use an external oscilloscope. Use settings similar to the ones shown in Figure P4.31.
2. In your Plant Identification 2.vi choose the value of amplitude and offset shown in Figure P4.32. A LabView error will be generated if the square wave generates negative values as these are not allowed as inputs to the chip. The value of the frequency is irrelevant; you just have to make sure the input is slow enough so that the motor speed reaches steady state as shown in Figure P4.32.
3. Run the Plant Identification 2.vi and the Oscilloscope. Press the Stop button on the Oscilloscope as soon as it shows a full semi-cycle of positive speed, similar to Figure P4.31.



Plant Identification 2.vi Front Panel *

Square Wave Parameters

Offset (Vdc): 4.5

Frequency (Hz): 0.2

Amplitude (Vpp): 4.5

Speed: 10.08

Square: 9.00

Steady state

Actual value not important

FIGURE P4.32 Plant Identification 2.vi Front Panel

4. Click on the Log button in the Oscilloscope, give the file a name, and save it to disk. Open the file using a spreadsheet program.
5. Note that the response of the system in the Oscilloscope is in all likelihood that of a first-order system, consistent with theoretical expectations. Thus, the transfer function will be of the form: $\frac{\Omega(s)}{E_i(s)} = \frac{K}{s\tau + 1}$.

   K can readily be found from the Oscilloscope or the Plant Identification 2.vi. In the example shown, $K = \frac{9.71}{9} = 1.079$. We will use the spreadsheet data to find the time constant, τ.
6. Use your spreadsheet data to find the time constant. For help on completing this task, go to www.wiley.com/college/nise.
7. Repeat the experiment for input voltages of 2 V, 5 V, and 9 V.

Postlab

1. Is your system linear? How do you know?
2. If your system is linear for a range of inputs, find a judicious interpolation between the three transfer functions you found in Part 7 of the lab. Write down your final transfer function result and save it for use in subsequent experiments.

Hardware Interface Laboratory

Experiment 8.4 Speed Control Using Gain Adjustment

Objective To control the speed of a motor in closed-loop using gain compensation. To make observations about tradeoffs between the compensated transient response and the steady-state error.

Material Required Computer with LabVIEW Installed; myDAQ; dc brushed gearmotor with Hall Sensor quadrature encoder (-10 V to $+10$ V normal operation range); and motor control chip BA6886N or a transistor circuit substitute

Files Provided at www.wiley.com/college/nise

Speed P Control Incomplete.vi
Signal Conditioning (subVI).vi

Prelab Answer the following questions:

1. Find the closed-loop transfer function from $R(s)$ to $C(s)$ for the system in Figure P8.23.
2. Draw the root locus for the system.
3. Draw the unit-step response for the system marking the settling time, peak time, and maximum output.
4. Find an expression for the steady-state error to a unit-step input for the system.

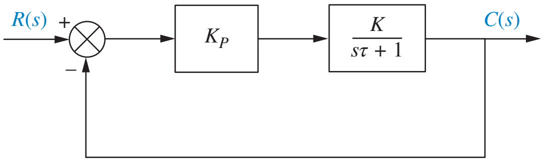


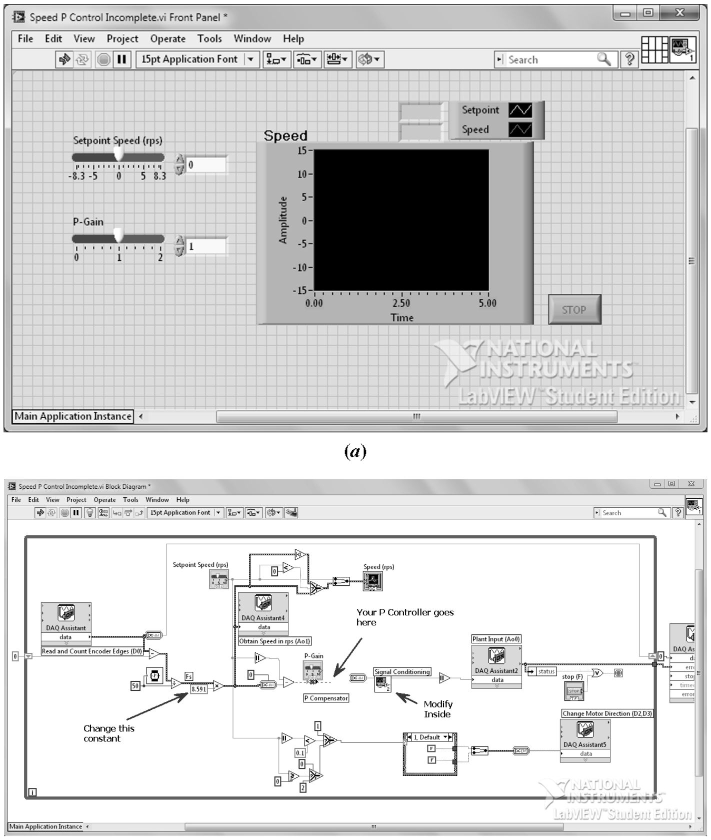
FIGURE P8.23

Lab

Software: The **Speed P Control Incomplete.vi** is provided and illustrated in Figure P8.24. You need to modify it as follows before it becomes operational.

1. You need to change the constant on the left to fit your motor's gear ratio and encoder CPR (counts per revolution) as shown in Figure P8.24(b).
2. You need to write a SubVI for a P controller and place it where the arrow indicates it in Figure P8.24(b). The function of a P controller is $u = K_P e$. Your SubVI has two inputs, the system's error e and the proportional constant K_P. It will have one output u.
3. Double-clicking on the **Signal Conditioning (SubVI)**, Figure P8.24(b), you get Figure P8.25. Modify the indicated constant to reflect the dead zone parameter of your motor. This SubVI limits the input voltage to the motor controller and eliminates the dead zone by offsetting the input to the motor controller.

Hardware: Figure P8.26 is the hardware diagram for speed control. The diagram is identical to the one in Experiment 4.6, except that Pins 2 and 10 in the motor controller chip connect to digital lines D2 and D3 in the myDAQ to allow changes in motor direction.



(*a*)

(*b*)

FIGURE P8.24 **Speed P Control Incomplete.vi: a. Front Panel; b. Block Diagram**

Procedure:

1. Verify the operation of your closed-loop system.
2. Draw a functional block diagram (similar to the ones shown in control systems textbooks) of the system. Do not include the signal conditioning functions, nor the change-of-direction signals.

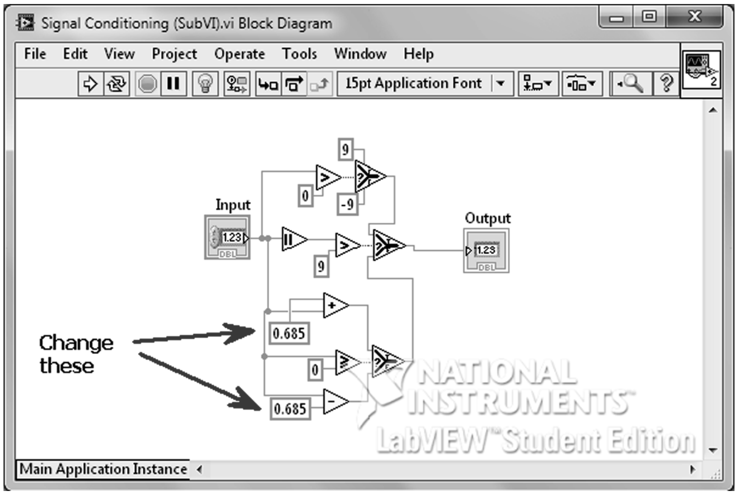

Signal Conditioning (SubVI).vi Block Diagram
File Edit View Project Operate Tools Window Help
15pt Application Font
Input
1.23
DBL
9
0
-9
9
Output
1.23
DBL
0.685
0
0.685
Change these
NATIONAL INSTRUMENTS
LabVIEW Student Edition
Main Application Instance

FIGURE P8.25 **Signal Conditioning (SubVI).vi Block Diagram**

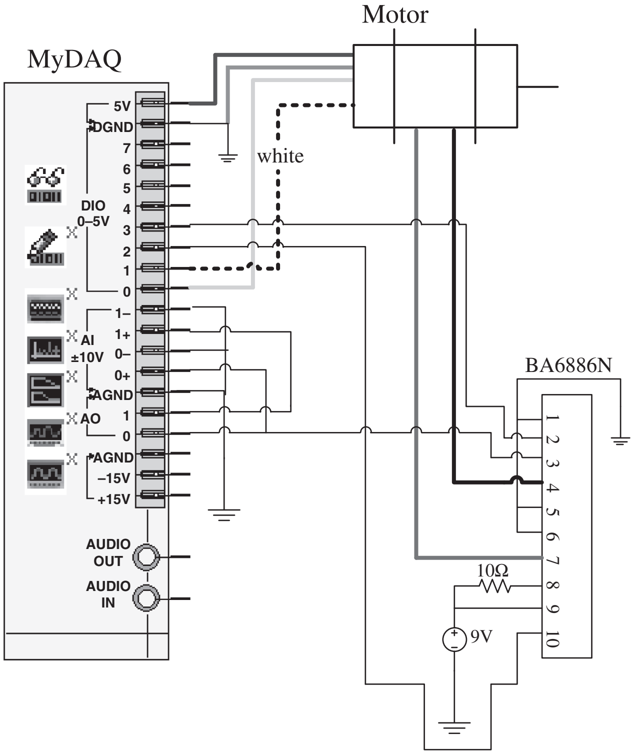


FIGURE P8.26 Wiring diagram[3]

3. Using the transfer function you found in Experiment 4.6, draw the system's root locus.
4. Find the theoretical range of K_P in which the system is closed-loop stable.
5. Run your program and system to find experimentally the range of K_P in which the system is closed-loop stable.
6. Make a judicious choice of three different values of K_P for experimentation.
7. Using the transfer function you found previously and the three judicious choices of proportional gain, complete the following table using hand calculations only (calculators OK, no computer simulations allowed). Show all your work.

K_P			
T_P—Peak time			
%OS—Percent overshoot			
T_s—Settling time			
e_{ss}—Steady state error (step input)			

Theoretical

8. For each one of the three values of K_P, perform step-input experiments; use one single value of step-input amplitude for the three values. Make sure that your oscilloscope

[3] MyDAQ right slot shown on left is taken from Multisim program module NI myDAQ design and also reproduced in White-Paper 11423, Figure 2. Both Multisim and the White Paper are from National Instruments.

captures contain the system's transient response in its entirety. Show measurements of all the parameters in the table above and fill in the following table. Please note that T_s, the settling time, is hard to measure in the current setting because of the limited number of analog channels present. Instead of measuring T_s, mark in your oscilloscope its theoretical value using the scope cursors.

K_P			
T_P—Peak time			
%OS—Percent overshoot			
e_{ss}—Steady-state error (step input)			

Experimental

Postlab Make a detailed comparison of your theoretical and experimental results. Discuss similarities and discrepancies between experimental and theoretical values and give possible reasons.

Experiment 8.5 Position Control Using Gain Adjustment

Objectives To control the angular position of the shaft of a permanent-magnet dc motor in closed-loop using gain compensation. To make observations about tradeoffs between the compensated transient response and the steady-state error.

Material Required Computer with LabVIEW Installed; myDAQ; dc brushed gearmotor with Hall Sensor quadrature encoder (−10 V to +10 V normal operation range); and motor control chip BA6886N or a transistor circuit substitute.

Files Provided at www.wiley.com/college/nise

Position control.vi
Signal Conditioning (SubVI).vi
P Controller (SubVI).vi

Prelab Answer the following questions:

1. For a given permanent-magnet dc motor it has been found that the transfer function from armature voltage $E_a(s)$ to angular velocity $\Omega(s)$ is $\frac{\Omega(s)}{E_a(s)} = \frac{K}{\tau s + 1}$. Find the transfer function of the motor from armature voltage to angular position $\frac{\Theta(s)}{E_a(s)}$.
2. Draw the root locus for the system in Figure P8.27.

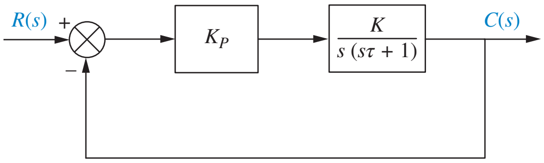


FIGURE P8.27

3. Draw the unit-step response for the system marking the settling time, peak time, and maximum output. Find all the possibilities: overdamped, critically damped, and underdamped.
4. Find an expression for the steady-state error to a unit-step input for the system.

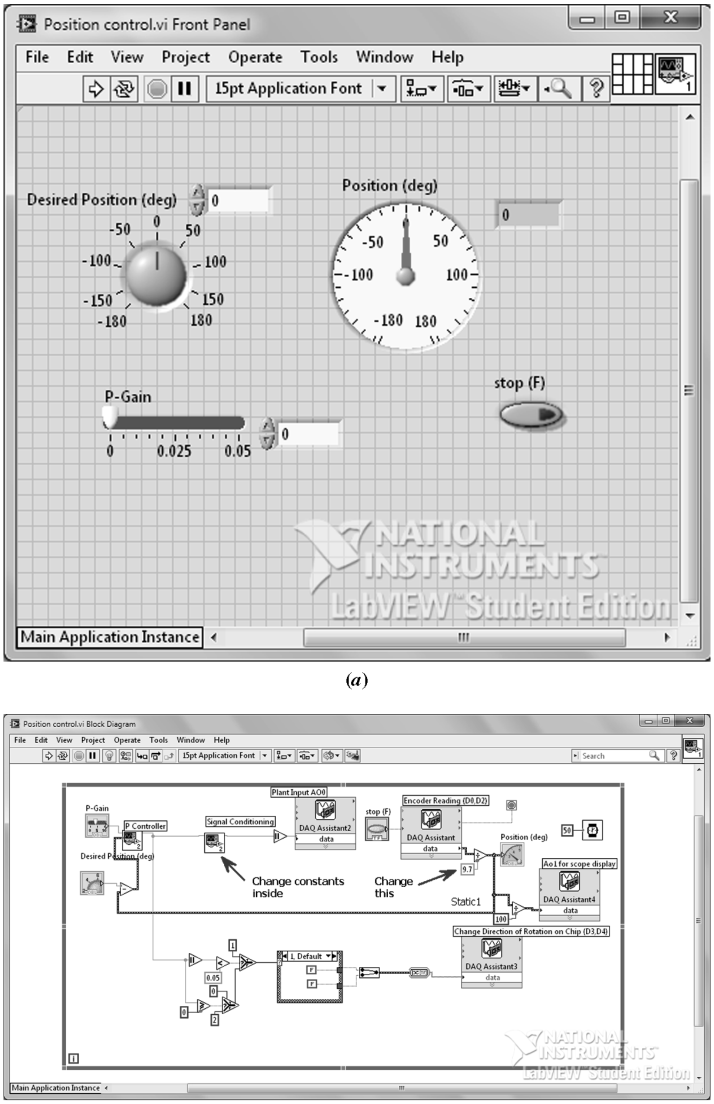
Lab

Software: The front panel and block diagram of the **Position control.vi** are shown in Figure P8.28. Change the constants inside the **Signal Conditioning SubVI** to match your dead zone parameters. The constant on the right of the diagram must be modified to match your motor's gear ratio.

Hardware: Make the following changes to the wiring diagram shown in Figure P8.26: Move the connections from D1, D2, and D3 to D2, D3, and D4, respectively. All the other connections remain the same.

Procedure:

1. Choose a small *P* gain. Verify the operation of your closed-loop system. The motor should be able to move in both directions and through the full-scale range.



(*a*)

(*b*)

FIGURE P8.28 **Position control.vi: a. Front Panel; b. Block Diagram**

2. Using the transfer function found in Experiment 4.6, calculate the motor's transfer function from armature voltage to angular position $\frac{\Theta(s)}{E_a(s)}$.
3. Draw a functional block diagram of the system. Do not include the signal conditioning functions, nor the change-of-direction signals. Label all the pertinent signals.
4. Using the transfer function $\frac{\Theta(s)}{E_a(s)}$ you just calculated, draw the system's root locus.
5. Find the theoretical range of K_P for which the system is closed-loop stable.
6. Run your program and system to find experimentally the range of K_P for which the system is closed-loop stable.
7. Make a judicious choice of three different values of K_P for experimentation.

8. Using the transfer function you calculated above and the three judicious choices of proportional gain, complete the following table using hand calculations only (calculators OK, no computer simulations allowed). Show all your work.

K_P			
T_P—Peak time			
%OS—Percent overshoot			
T_s—Settling time			
e_{ss}—Steady-state error (step input)			

Theoretical

9. For each one of the three values of K_P, perform step-input experiments using one value of step input for the three values. Make sure that your oscilloscope captures contain the system's transient response in its entirety. Show measurements of all the parameters in the table above and fill in the following table. Please note that T_s, the settling time, is hard to measure in the current setting because of the limited number of analog channels available. Instead of measuring T_s, mark in your oscilloscope its theoretical value using the scope cursors.

K_P			
T_P—Peak time			
%OS—Percent overshoot			
e_{ss}—Steady-state error (step input)			

Experimental

Postlab Make a detailed comparison of your theoretical and experimental tables. Discuss similarities and discrepancies between experimental and theoretical results and give possible reasons.

Hardware Interface Laboratory

Experiment 9.3 Speed Control Using PI Control

Objectives To control the speed of the motor in closed loop using integral control and to investigate the tradeoffs of this approach

Material Required Computer with LabView installed; myDAQ; dc brushed gearmotor with Hall Sensor quadrature encoder (−10V to +10V normal operation range); and motor control chip BA6886N (or a transistor circuit substitute)

Files Provided at www.wiley.com/college/nise

Speed PI Control.vi
Signal Conditioning (SubVI).vi
PI Controller (SubVI).vi

Prelab Answer the following questions:

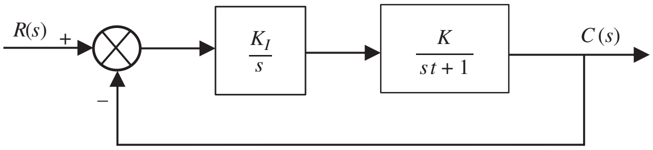


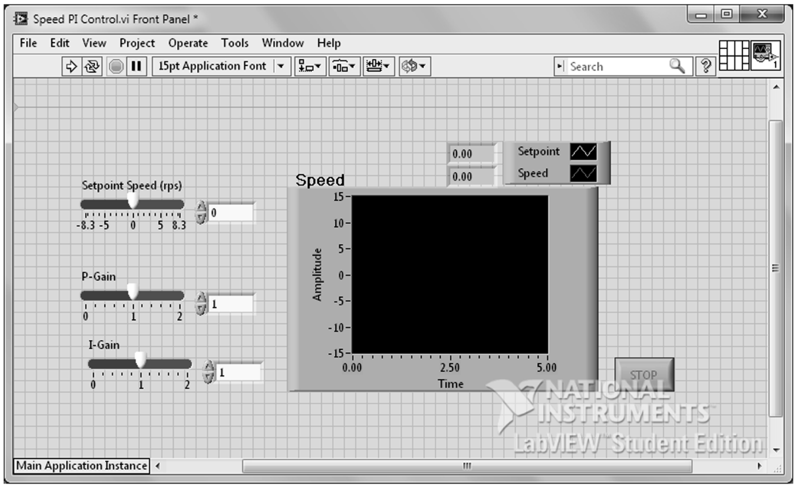
FIGURE P9.14

For the system shown in Figure P9.14, do the following:

1. Find the closed-loop transfer function from $R(s)$ to $C(s)$.
2. Draw the root locus as a function of K_I.
3. Draw the unit-step response marking the settling time, peak time, and maximum output. Find all the possibilities: overdamped, critically damped, and underdamped.
4. Find an expression for the steady-state error for a unit-step input.

Lab

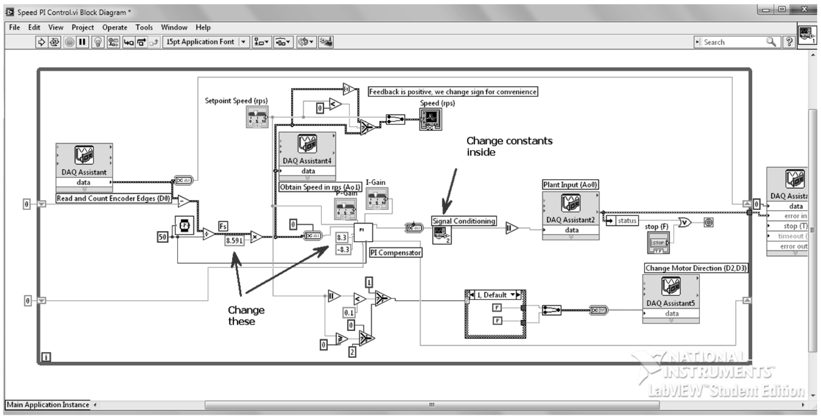
Software: Use the Speed PI Control.vi and change the constant on the left to fit your motor's gear ratio and encoder CPR as shown in Figure P9.15(b). Also change the



Setpoint Speed (rps) 0
P-Gain 1
I-Gain 1
0.00 Setpoint
0.00 Speed

(a)

FIGURE P9.15 Speed PI Control.vi: **a.** Front Panel; (*figure continues*)



(*b*)

FIGURE P9.15 (*Continued*) **b.** Block Diagram

constants wired to the PI controller. These values should be the maximum motor voltage-dead-zone constant. Change the dead-zone constants inside the signal conditioning block just as you did in Experiment 8.4.

Hardware: Same as Experiment 8.4

Procedure:

1. Make the P gain = 0, and choose a small I gain. Verify the operation of your closed-loop system. In this experiment we will keep the P gain at 0.
2. Draw a functional block diagram (similar to that presented in Chapter 1) of the system. Do not include the signal conditioning functions, nor the change-of-direction signals.
3. Using the transfer function you found in Experiment 4.6, draw the system's root locus.
4. Find the theoretical range of K_I in which the system is closed-loop stable.
5. Run your program and system to find experimentally the range of K_I in which the system is closed-loop stable.
6. Make a judicious choice of three different values of K_I for experimentation.
7. Using the transfer function you found in Experiment 8.4 and the three judicious choices of proportional gain, complete the following table using hand calculations only (calculators OK, computer simulations are not acceptable). Show all your work.

K_I			
T_P—Peak time			
%OS—Percent overshoot			
T_s—Settling time			
e_{ss}—Steady-state error (step input)			

Theoretical

8. For each one of the three values of K_I, perform step-input experiments. Use a single value of step input for the three values of K_I. Make sure that your oscilloscope captures

contain the system's transient response in its entirety. Show measurements of all the parameters in the Theoretical table above and fill in the following Experimental table. Please note that T_s, the settling time, is hard to measure in the current setting because of the limited number of analog channels available. Instead of measuring T_s, mark on your oscilloscope the theoretical value using the scope cursors. (**Important:** In this experiment stop the VI before restarting it every time you apply a step input. This action will reset the integrator).

K_I			
T_p—Peak time			
%OS—Percent overshoot			
e_{ss}—Steady-state error (step input)			

Experimental

Postlab Make a detailed comparison of your theoretical and experimental tables. Discuss similarities and discrepancies between experimental and theoretical and give possible reasons.

Founded in 1807, John Wiley & Sons, Inc. has been a valued source of knowledge and understanding for more than 200 years, helping people around the world meet their needs and fulfill their aspirations. Our company is built on a foundation of principles that include responsibility to the communities we serve and where we live and work. In 2008, we launched a Corporate Citizenship Initiative, a global effort to address the environmental, social, economic, and ethical challenges we face in our business. Among the issues we are addressing are carbon impact, paper specifications and procurement, ethical conduct within our business and among our vendors, and community and charitable support. For more information, please visit our website: www.wiley.com/go/citizenship.

The software programs and experiments available with this book have been included for their instructional value. They have been tested with care but are not guaranteed for any particular purpose. The publisher and author do not offer any warranties or restrictions, nor do they accept any liabilities with respect to the programs and experiments.

Library of Congress Cataloging-in-Publication Data

Nise, Norman S.
Control systems engineering / Norman S. Nise, California State Polytechnic University, Pomona. — Seventh edition.
1 online resource.
Includes bibliographical references and index.
Description based on print version record and CIP data provided by publisher; resource not viewed.
ISBN 978-1-118-80082-9 (pdf) — ISBN 978-1-118-17051-9 (cloth : alk. paper)
1. Automatic control–Textbooks. 2. Systems engineering–Textbooks. I. Title.
TJ213
629.8–dc23
2014037468

Printed in the United States of America

10 9 8 7 6 5 4 3 2 1