QUANSER
INNOVATE. EDUCATE.

STUDENT WORKBOOK

SRV02 Base Unit Experiment For Matlab®/Simulink® Users

Standardized for ABET Evaluation Criteria

Developed by:
Jacob Apkarian, Ph.D., Quanser
Michel Lévis, M.A.Sc., Quanser
Hakan Gurocak, Ph.D., Washington State University

SRV02 educational solutions are powered by:

MATLAB SIMULINK

Course material complies with:

ABET

CAPTIVATE. MOTIVATE. GRADUATE.

Acknowledgements

Quanser, Inc. would like to thank the following contributors:
Dr. Hakan Gurocak, Washington State University Vancouver, USA, for his help to include embedded outcomes assessment, and
Dr. K. J. Åström, Lund University, Lund, Sweden for his immense contributions to the curriculum content.

ROTARY SERVO BASE UNIT INTEGRATION

In this section, we explain how to send command voltages to the QUANSER® Rotary Servo Base Unit and measure the position and speed of its load shaft in real-time using your computer.

Topics Covered

- Become familiarized with the DC motor servo system using the Rotary Servo Base Unit.
- Using Rapid Control Prototyping tools to interact with Rotary Servo Base Unit system through the QUANSER® QUARC® software.
- Measuring motor position using potentiometer and encoder (i.e. sensor calibration).
- Velocity estimation using encoder measurement.
- Actuator calibration.

Prerequisites

- System has been setup and tested by going through the Rotary Servo Base Unit Quick Start Guide.
- Familiar with Transfer function fundamentals, e.g. obtaining a transfer function from a differential equation.
- Familiar with MATLAB® and SIMULINK® fundamentals.

Caution

Make sure the system has been setup and tested by going through the Rotary Servo Base Unit Quick Start Guide before starting this experiment!

1 Background

1.1 QUARC Software

The QUARC® software is used with SIMULINK® to interact with the hardware of the Rotary Servo Base Unit system. QUARC® is used to drive the DC motor and read angular position of the disc.

The basic steps to create a SIMULINK® model with QUARC® in order to interact with the Rotary Servo Base Unit hardware are:

1. Make a SIMULINK® model that interacts with your installed data acquisition device using blocks from the *QUARC Targets* library.
2. Build the real-time code.
3. Execute the code.

Type `doc quarc` in MATLAB® to access QUARC® documentation and demos.

1.2 DC Motor

Direct-current motors are used in a variety of applications. As discussed in the Rotary Servo Base Unit User Manual, the Rotary Servo Base Unit has a brushed DC motor that is connected to a linear power amplifier. See the Rotary Servo Base Unit User Manual for details.

1.3 Rotary Potentiometer

A rotary potentiometer, or pot, is a manually controlled variable resistor. See the example shown in Figure 1.1. It typically consists of an exposed shaft, three terminals (A, W, and B), an encased internal resistive element shaped in a circular pattern, and a sliding contact known as a wiper. By rotating the shaft, the internal wiper makes contact with the resistive element at different positions, causing a change in resistance when measured between the center terminal (W) and either of the side terminals (A or B). The total resistance of the potentiometer can be measured by clamping a multimeter to terminals A and B.

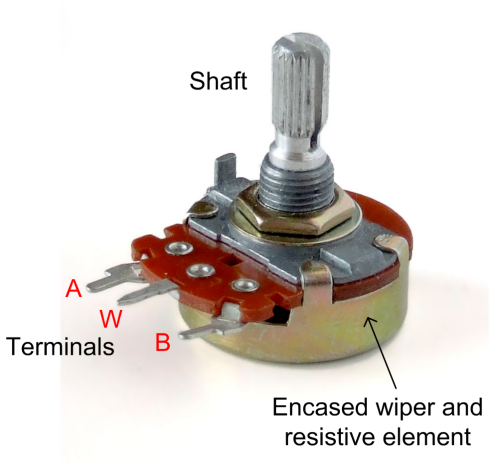


Figure 1.1: Schematic diagram of a potentiometer (source: Wikipedia).

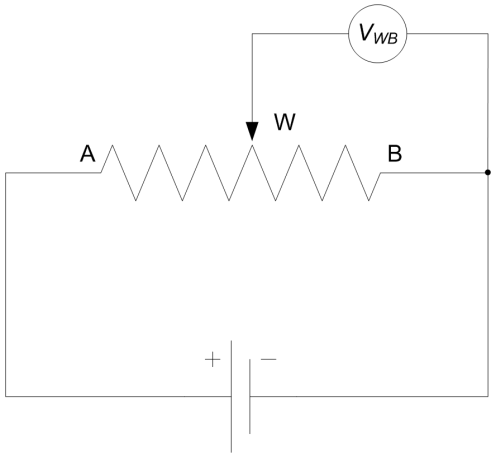


Figure 1.2: Schematic diagram of the voltage dividing characteristic of a rotary potentiometer.

A schematic diagram of the voltage dividing characteristic of a potentiometer is illustrated in Figure 1.2. By applying a known voltage between terminals A and B (V_{AB}), voltage is divided between terminals AW and WB where:

$$V_{AB} = V_{AW} + V_{WB}$$

When connected to an external shaft, a rotary potentiometer can measure absolute angular displacement. By applying a known voltage to the outside terminals of the pot, we can determine the position of the sensor based on the output voltage V_{AW} or V_{WB} which will be directly proportional to the position of the shaft.

One of the advantages of using a potentiometer as an absolute sensor is that after power loss, position information is retained since the resistance of the pot remains unchanged. While pots are an effective way to obtain a unique position measurement, caution must be used since their signal output may be discontinuous. That is, after a few revolutions potentiometers may reset their signal back to zero. Another disadvantage of most pots is that they have physical stops that prevent continuous shaft rotation.

1.4 Encoders

An incremental optical encoder (Figure 1.3) is a relative angular displacement sensor which measures angular displacement relative to a previously known position. Unlike an absolute encoder, an incremental encoder does not retain its position information upon power loss. An incremental encoder outputs a series of pulses which correlate to relative change in angular position. Encoders are commonly used to measure angular displacement of rotating load shafts. Information extracted from an incremental encoder can also be used to derive instantaneous rotational velocities.

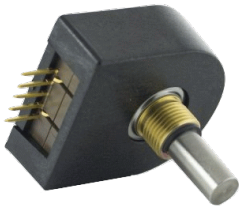


Figure 1.3: US Digital incremental rotary optical shaft encoder.

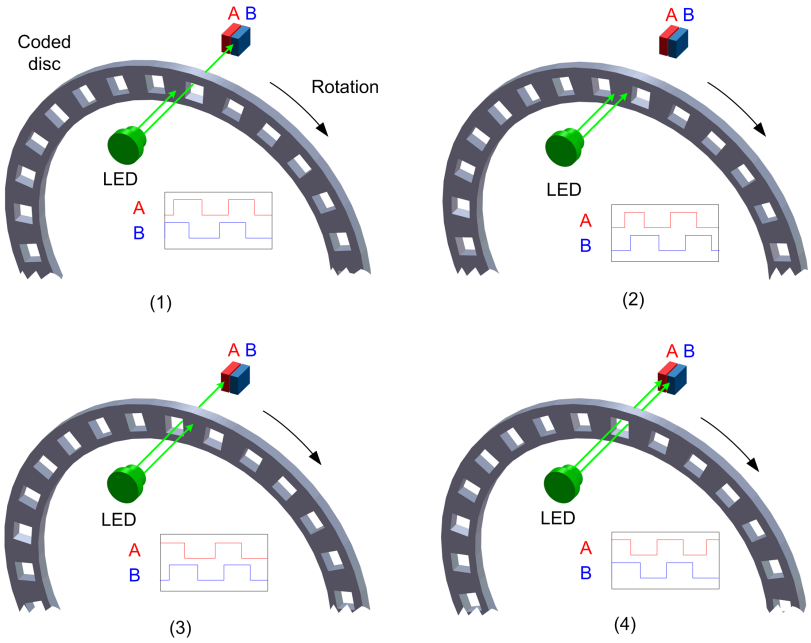


Figure 1.4: Output of an incremental encoder showing signals A and B when rotating in a clockwise manner.

An incremental optical encoder typically consists of a coded disk, an LED, and two photo sensors. The disk is coded with an alternating light and dark radial pattern causing it to act as a shutter. As shown schematically in Figure 1.4, the light emitted by the LED is interrupted by the coding as the disk rotates around its axis. The two photo sensors (A and B) positioned behind the coded disk sense the light emitted by the LED. The process results in A and B signals, or pulses, in four distinct states: (1) A off, B on; (2) A off, B off; (3) A on, B off; (4) A on, B on. Encoders which output A and B signals are often referred to as quadrature encoders since the signals are separated in phase by $90°$. The resolution of an encoder corresponds to the number of light or dark patterns on the disk and is given in terms of pulses per revolution, or PPR.

In order to make encoder measurements, you need to connect the encoder to a counter to count the A and B signals. Then use a decoder algorithm to determine the number of counts and direction of rotation. Three decoding algorithms are used: X1, X2, X4.

X1 Decoder: When an X1 decoder is used, only the rising or falling edge of signal A is counted as the shaft rotates. When signal A leads signal B, the counter is incremented on the rising edge of signal A. When signal B leads signal A, the counter is decremented on the falling edge of signal A. Using an X1 decoder, a $1,024$ PPR encoder will result in a total of $1,024$ counts for every rotation of the encoder shaft.

X2 Decoder: When an X2 decoder is used, both the rising and falling edges of signal A are counted as the shaft rotates. When signal A leads signal B, the counter is incremented on both the rising and falling edge of signal A. When signal B leads signal A, the counter is decremented on both the rising and falling edges of signal A. Using an X2 decoder, a $1,024$ PPR encoder will generate a total of $2,048$ counts for every rotation of the encoder shaft.

X4 Decoder: When an X4 algorithm is used, both the rising and falling edges of both signals A and B are counted. Depending on which signal leads, the counter will either increment or decrement. An X4 decoder generates four times the number of counts generated by an X1 decoder resulting in the highest resolution among the three types of decoders. Using an X4 decoder, a $1,024$ PPR encoder will generate a total of $4,096$ counts for every rotation of the encoder shaft.

The angular resolution of an encoder depends on the encoder's pulses per revolution (PPR) and the decoding algorithm used:

$$\Delta\theta = \frac{2\pi}{N \times PPR} \tag{1.1}$$

where N = 1, 2, or 4 corresponds to X1, X2, and X4 decoders respectively. Figure 1.5 compares the number of

counts generated by each of the X1, X2, and X4 decoders.

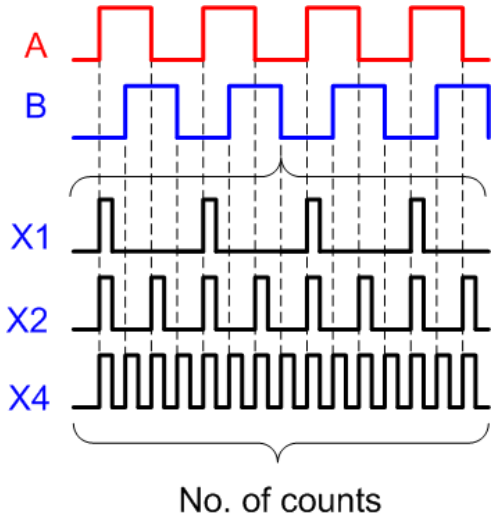


Figure 1.5: Comparison of the number of counts generated by the X1, X2, and X4 decoding algorithms.

1.5 Velocity Estimation

While encoders are typically used for measuring angular displacement, they can also measure rotational speeds. The velocity can be found by taking the difference between consecutive angle measurements

$$\omega(k) = \frac{\theta(k) - \theta(k-1)}{h}$$

where $\theta(k)$ represents the k^{th} position measurement sample and h is the sampling interval of the control software. The resolution or *ripple* in the velocity measurement is given by

$$\Delta\omega = \frac{\Delta\theta}{h} \tag{1.2}$$

where ω is rotational speed (rad/s) and $\Delta\theta$ is the resolution of the encoder given in Equation 1.1, and h is the sampling interval. Note that the ripple velocity increases when the sampling period decreases.

1.6 Filtering the Velocity Signal

In practice, determining rotational speeds by means of derivation of the discontinuous encoder output often results in signal noise. The ripple in the velocity signal due to sampling can be reduced by filtering. Applying a first-order low-pass filter to the measured velocity signal

$$Y_f(s) = \frac{1}{T_f s + 1}$$

where Y is the measured signal, Y_f is the filtered signal, and T_f is the filter time constant. Low-pass filters are also often represented in terms of cutoff frequency ω_f:

$$Y_f(s) = \frac{\omega_f}{s + \omega_f}.$$

where $\omega_f = 1/T_f$. The low-pass filter attenuates the high-frequency components of the signal higher that the cut-off frequency specified, i.e. passes signal with frequencies $\omega \leq \omega_f$.

Rearranging the filter transfer function above,

$$sT_f Y_f(s) + Y_f(s) = Y(s)$$

we can represent the filtering using the differential equation

$$T_f \frac{dy_f(t)}{dt} + y_f(t) = y(t).$$

The derivative can be approximate by the difference between the currently measured sample, $y(t)$, and the previous sample, $y(t-h)$:

$$T_f \frac{y_f(t) - y_f(t-h)}{h} + y_f(t) = y(t).$$

Solving for $y_f(t)$,

$$y_f(t) = \frac{T_f}{T_f + h} y_f(t-h) + \frac{h}{T_f + h} y(t). \tag{1.3}$$

Thus the high-frequency error is reduced by a factor of $h/(T_f + h)$. The ripple in the filtered velocity signal caused by the encoder is

$$\Delta y_f(t) = \frac{h}{T_f + h} \frac{\Delta\theta}{h} \tag{1.4}$$

Note: Filtering does add dynamics to the signal, i.e. small delay, as given by the term $\frac{T_f}{T_f+h} y_f(t-h)$ in Equation 1.3.

1.7 Encoder Counter Wrapping

Encoder counters on data acquisition (DAQ) devices have a rated number of bits, n. For example, a DAQ with an 8-bit signed counter has range between -128 and $+127$ (i.e. -2^7 and $2^7 - 1$). When the encoder counts go beyond this this range, the measured counts from the DAQ ends up *wraps*.

Consider this example. If the encoder is measuring 100 counts, then the DAQ counter will output 100 (no wrapping). However, if the encoder is measuring 150 counts, then the 8-bit signed counter will read $(150+128)$ mod $256-128 = -106$ counts. Thus the encoder wraps and causes a discontinuity. This can be very problematic in closed-loop feedback control. The QUARC® software includes an Inverse Modulus block that unwraps the counter signal and keep a continuous signal. For more information, see the Inverse Modulus block in the QUARC documentation.

2 Pre-Lab

1. Given the Rotary Servo Base Unit rotary encoder has 1024 PPR and quardrature decoding is used in the DAQ, what is the resolution of the encoder?
2. What is the sensor gain for the rotary potentiometer that converts voltage measured into load gear angle (in radians)? See the Rotary Servo Base Unit User Manual for potentiometer specifications.
3. The rotary incremental encoder and rotary potentiometer both measure the position of the load gear but are different types of sensors. Give one difference between the two sensors.
4. Assuming the DAQ device you are using has a 16-bit signed encoder counter. What is the counter range and how does that affect the measuring encoder signal?
5. Find the velocity estimation ripple of the Rotary Servo Base Unit when the sampling interval is 0.01 s, 0.002 s, and 0.001 s. How does increasing the sampling rate (i.e. decreasing the sampling interval) affect the ripple?
6. A low-pass filter with a time constant of $T_f = 0.01$ is used in the velocity estimator and the sampling of the control runs at $h = 0.001$ s. Evaluate the ripple in the velocity with this filter.

3 In-Lab Exercises

In this lab, we will make a SIMULINK® model using QUARC® blocks to drive the DC motor and measure the corresponding load gear angle and velocity. To measure the position, a model similarly as shown in Figure 3.1 will be designed.

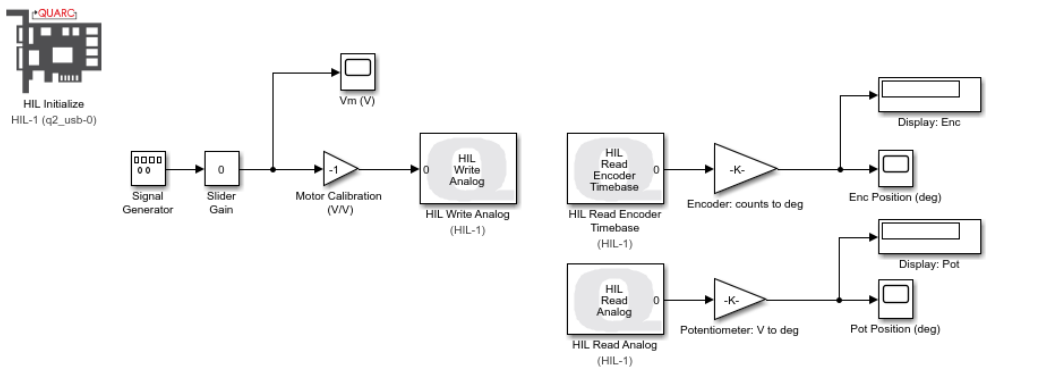


Figure 3.1: Simulink model used with QUARC® to drive motor and read angle on Rotary Servo Base Unit

3.1 Configuring a Simulink Model for QUARC

Follow these steps build a SIMULINK® model that will interface to the Rotary Servo Base Unit using QUARC®:

1. Load the MATLAB® software.
2. Create a new SIMULINK® diagram by going to *File* | *New* | *Model* item in the menu bar.
3. Open the SIMULINK® Library Browser window by clicking on the *View* | *Library Browser* item in the SIMULINK® menu bar or clicking on the SIMULINK® icon.
4. Expand the *QUARC Targets* item and go to the *Data Acquisition* | *Generic* | *Configuration* folder, as shown in Figure 3.2.
5. Click-and-drag the HIL Initialize block from the library window into the blank SIMULINK® model. This block is used to configure your data acquisition device.
6. Double-click on the HIL Initialize block.
7. In the *Board type* field, select the DAQ device that is installed in your PC/laptop. For example, if you are using a Quanser Q2-USB then select q2_usb. If you are using the Q8-USB board to connect to the Rotary Servo Base Unit, then select *q8_usb*.
8. Go to the *QUARC* | *Set default options* item to configure the SIMULINK® model to *external* mode - as opposed to the *normal* mode used only for simulation - and generate real-time QUARC code for the local target, i.e. your 64-bit PC/laptop running Windows.
9. Select the *QUARC* | *Build* item. Various lines in the MATLAB® Command Window should be displayed as the model is being compiled. This creates a QUARC® executable (.exe) file which we will commonly refer to as a QUARC® controller.
10. Run the QUARC® controller. To do this, go to the SIMULINK® model tool bar, shown in Figure 3.3, and click on the *Connect to target* icon and then on the *Run* icon. You can also go *QUARC* | *Start* to run the code.
11. If you successfully ran the QUARC® controller without any errors, then you can stop the code by clicking on the *Stop* button in the tool bar (or go to *QUARC* | *Stop*).

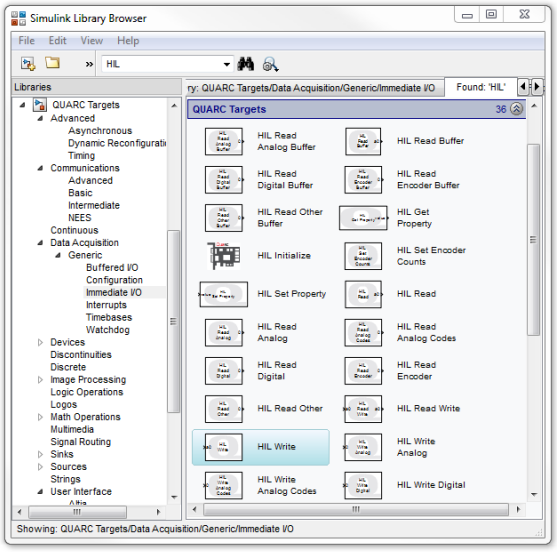


Figure 3.2: *QUARC Targets* in SIMULINK® Library Browser

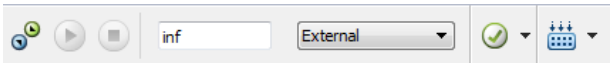


Figure 3.3: SIMULINK® model toolbar: connect to target and compilation

3.2 Reading the Motor Position

3.2.1 Using the Encoder

Follow these steps to read the encoder:

1. Using the SIMULINK® model you configured for the Rotary Servo Base Unit in the previous section, add the HIL Read Encoder Timebase block from the *QUARC Targets* | *Data Acquisition* | *Generic* | *Timebases* category in the Library Browser.

   Note: Timebase blocks: Using a *timebase* block in your model makes the sampling of the QUARC controller is triggered off the hardware clock on the DAQ - as opposed to the system PC timer (if only HIL blocks from the Immediate category were used). Only one HIL Timebase block may appear in the model.

2. Connect the HIL Read Encoder Timebase to a Gain and Display block similar to Figure 3.1 (without the HIL Write Analog block). In the Library Browser, you can find the Display block from the *Simulink* | *Sinks* and the Gain block from *Simulink* | *Math Operations*.

3. Build the QUARC® controller. The code needs to be re-generated again because we have modified the Simulink diagram.

4. Run the QUARC® controller.

5. Rotate the disc back and forth. The Display block shows the number of counts measured by the encoder. The encoder counts are proportional to the angle of disc.

6. What happens to the encoder reading every time the QUARC® controller is started? Stop the controller, move around the disc, and re-start the controller. What do you notice about the encoder measurement when the controller is re-started? Is this what you expected from this type of sensor?

7. Measure how many counts the encoder outputs for a full rotation. Briefly explain your procedure to determine this and validate that this matches the specifications given in the Rotary Servo Base Unit User Manual.

8. Ultimately we want to display the disc angle in degrees, not counts. Set the Gain block to a value that converts counts to degrees. This is called the *sensor gain*. Run the QUARC® controller and confirm that the Display block shows the angle of the disc correctly.

3.2.2 Using the Potentiometer

Follow these steps to read the potentiometer:

1. Using the SIMULINK® model you configured for the Rotary Servo Base Unit in the previous section, add the HIL Read Analog block from the *QUARC Targets* | *Data Acquisition* | *Generic* | *Immediate* category in the Library Browser, similary as shown in Figure 3.1.

2. Connect the HIL Read Analog to a Gain and Display block similar to Figure 3.1 (without the HIL Write Analog block). In the Library Browser, you can find the Display block from the *Simulink* | *Sinks* and the Gain block from *Simulink* | *Math Operations*.

3. Build the QUARC® controller. The code needs to be re-generated again because we have modified the Simulink diagram.

4. Run the QUARC® controller.

5. Rotate the disc back and forth. The Scope block shows the voltage measured by the potentiometer. The voltage is proportional to the angle of disc.

6. What happens to the potentiometer reading every time the QUARC® controller is started? Stop the controller, move around the disc, and re-start the controller. What do you notice about the potentiometer measurement when the controller is re-started? Given the type of sensors, is this result as expected?

7. Measure the voltage output for a full rotation. What do you notice in the signal? Briefly explain your procedure to determine this and validate that this matches the specifications given in the Rotary Servo Base Unit User Manual.

8. In order to measure disc angle in degrees, not voltage, set the Gain block to a value that converts voltage to degrees based on your calculation in Section 2. Run the QUARC® controller and confirm that the Display block shows the angle of the disc correctly.

3.3 Driving the DC Motor

1. Add the HIL Write Analog block from the *Data Acquisition* | *Generic* | *Immediate I/O* category into your SIMULINK® diagram, similarly as shown in Figure 3.1. This block is used to output a signal from analog output channel #0 on the data acquisition device. This is connected to the power amplifier (e.g. Quanser VoltPAQ-X1) that drives the DC motor.

2. Add the Constant block found in the *Simulink* | *Sources* folder to your Simulink model. Connect the Constant and HIL Write Analog blocks together, as shown in Figure 3.1.

3. Build and run the QUARC® controller.

4. Set the Constant block to 0.5. This applies 0.5 V to the DC motor in the Rotary Servo Base Unit. Confirm that we are obtaining a *positive measurement when a positive signal is applied*. This convention is important, especially in control systems when the design assumes the measurement goes up positively when a positive input is applied. If this convention is not followed, add a Gain block to the Simulink model. Finally, in what direction does the disc rotate (clockwise or counter-clockwise) when a positive input is applied?

5. Stop the QUARC® controller.

3.4 Reading the Motor Velocity

In this lab, we will make a SIMULINK® model using QUARC® blocks to drive the DC motor and measure the corresponding load gear velocity, as shown in Figure 3.4.

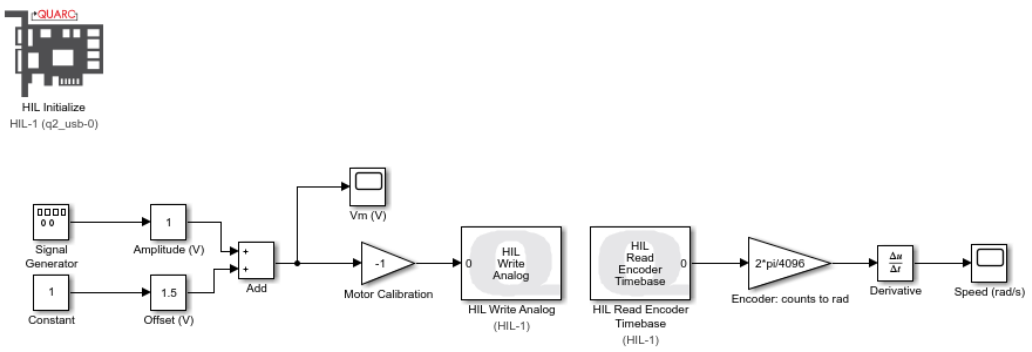


Figure 3.4: Simulink model used with QUARC® to drive motor and read angular velocity on Rotary Servo Base Unit

Follow these steps to read the motor velocity:

1. Based on the SIMULINK® model you configured for the Rotary Servo Base Unit in the previous section, add a Derivative block from the *Math Operations* library to the output of the encoder-based motor position measurement and connect its output to a scope to monitor the load gear velocity, similarly as shown in Figure 3.4. You may want to save this as another Simulink model (e.g. q_servo_vel).

   Note: The potentiometer will not be used in this lab.

2. Configure the model to apply a 0.4 Hz square wave voltage signal going between 0.5V and 2.5V. Use the Signal Generator, Constant, and Add or Sum blocks, similarly as shown in Figure 3.4

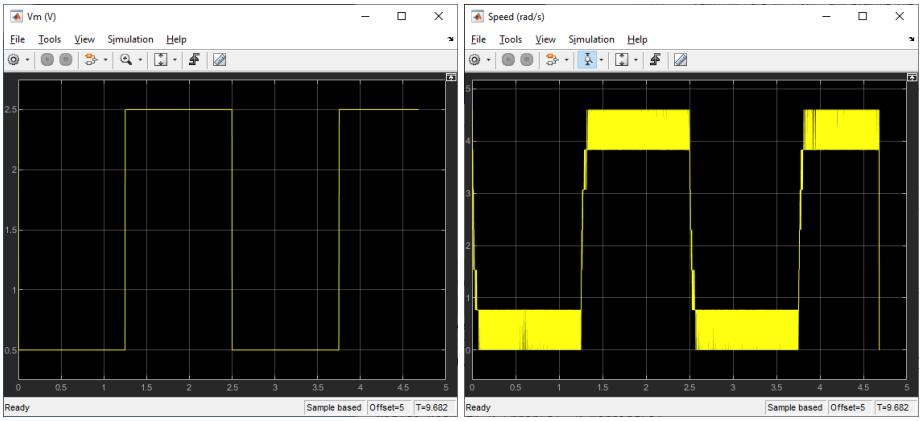
3. Finally, change the encoder sensor gain such that **it measures radians instead of degrees**.

4. Build the QUARC® controller. The code needs to be re-generated again because we have modified the Simulink diagram.

5. Run the QUARC® controller.

6. Notice that, as shown in Figure 3.5, the velocity measurement using direct derivative is noisy. Stop the controller and measure the velocity ripple. You can use *Cursor Measurements* tool in the Scope for more accurate measurements. Given that, by default, the QUARC controller is set to run at a sampling interval of 0.002 s (i.e. 500 Hz) compare the measured ripple to the calculation made in Section 2.

7. Increase the control loop rate of the QUARC controller by decreasing the sampling interval to 0.001 s, i.e. increasing the sampling rate to 1 kHz, and examine the change in the ripple velocity measurement. To do this, go to *QUARC* | *Options* and change the *Fixed-Step Size* parameter in the *Solver* pane, as shown in Figure 3.6. Once the sampling is set, build and start the QUARC controller. How does this compare with your calculated value in Section 2.



(a) Voltage input (b) Motor velocity

Figure 3.5: Sample velocity estimate usign direct derivative when running at 500 Hz

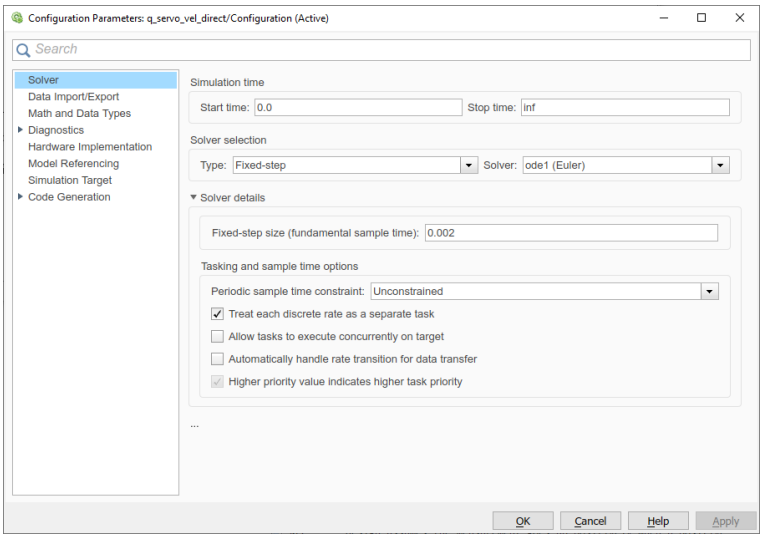
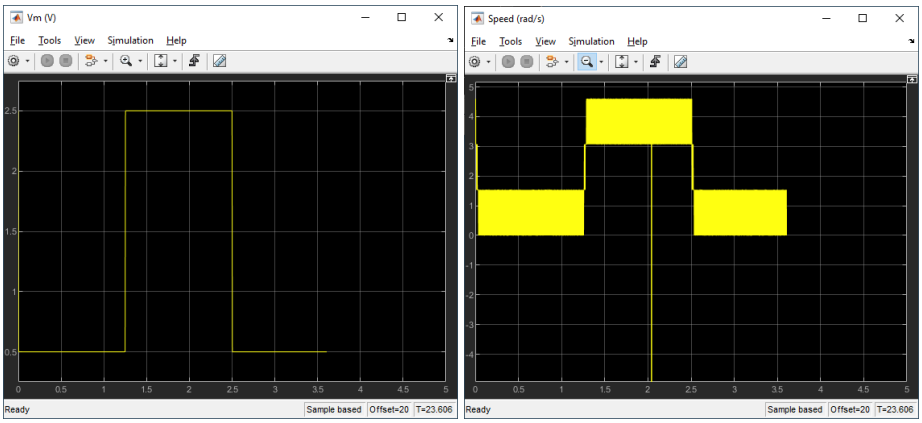


Figure 3.6: Solver pane in the Simulink configuration parameters

8. Run the controller for a minute or two. Depending on the data acquisition device you are using (e.g. Q2-USB or Q8-USB), you may notice a *spike* or discontinuity that occurs in the velocity estimation, as shown in Figure 3.7. This encoder wrapping is due to the range of the DAQ encoder counter. To remove the wrapping and maintain a continuous signal, use the QUARC Inverse Modulus block. See the QUARC Documentation for information on what value to use according to the bits on your DAQ encoder counter. Build and the run your QUARC controller and ensure the signal is now continuous (i.e. no *spike* occurs).

   Hint: The data acquisition encoder counter specifications can be found in DAQ User Manual.

9. Add a low-pass filter with a time constant of $T_f = 0.01$ s using the Simulink Transfer Fcn block. This translate into having a filter with a cutoff frequency of $1/0.01 = 100$ rad/s. Compare the velocity estimation using the direct derivative with filtering. How much noise does the filter remove? Is this in-line with the ripple velocity calculated in Section 2? Attach the response of the filtered velocity estimate and the Simulink model you designed.



(a) Voltage input (b) Motor velocity

Figure 3.7: Discontinuity in speed measurement due to encoder coder range

4 File Description and Configuration

4.1 Overview of Files

File Name	Description
Rotary Servo Base Unit - Integration Workbook (Student).pdf	This laboratory guide contains pre-lab and in-lab exercises demonstrating how to interface to the Quanser Rotary Servo Base Unit rotary plant using QUARC®.

Table 4.1: Files supplied with the Rotary Servo Base Unit Integration laboratory.

4.2 Configuring the Rotary Servo Base Unit and the Lab Files

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

1. Set up the Rotary Servo Base Unit in the high-gear configuration and with the disc load as described in Rotary Servo Base Unit User Manual.
2. Ensure the Rotary Servo Base Unit has been connected properly and tested given in the Rotary Servo Base Unit Quick Start Guide.
3. Follow the directions in Section 3.1 to create your model to interface to the Rotary Servo Base Unit.

5 Lab Report

This laboratory contains four experiments:

1. Configuring Simulink Model for QUARC
2. Reading Motor Position
3. Driving the DC motor
4. Reading the Motor Velocity

When you prepare your lab report, you can follow the outline given in Section 5.1 to build the *content* of your report. Also, in Section 5.2 you can find some basic tips for the *format* of your report.

5.1 Template for Content

I. PROCEDURE

I.1. Configuring Simulink Model for QUARC

1. Briefly describe the main goal of this experiment and the procedure.
2. Briefly describe the experimental procedure in Step 10 in Section 3.1, *Configuring and running a model in QUARC*

I.2. Reading Motor Position

1. Briefly describe the main goal of this experiment and the experimental procedure (3.2).
2. Briefly describe the experimental procedure in Step 6 in Section 3.2.1, *Reading motor position using the encoder*
3. Briefly describe the experimental procedure in Step 6 in Section 3.2.2, *Reading motor position using the potentiometer*

I.3. Driving the DC Motor

1. Briefly describe the main goal of this experiment and the experimental procedure (Section 3.3).
2. Briefly describe the experimental procedure in Step 4 in Section 3.3), *Applying voltage to the DC motor*

I.4. Reading Motor Velocity

1. Briefly describe the main goal of this experiment and the experimental procedure (Section 3.4).
2. Briefly describe the experimental procedure in Step 6 of Section 3.4, *Reading the motor velocity using a direct derivative*.

II. RESULTS

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

1. Response and Simulink model designed in step 9 in Section 3.4, *Motor velocity estimation*.

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

III.1. Reading Motor Position

1. Step 7 in Section 3.2.1, *Encoder sensor response compared to specifications*.
2. Step 8 in Section 3.2.1, *Encoder sensor gain validation*.
3. Step 7 in Section 3.2.2, *Potentiometer sensor response compared to specifications*.
4. Step 8 in Section 3.2.2, *Potentiometer sensor gain validation*.

III.2. Reading Motor Velocity

1. Steps 6 and 7 in Section 3.4, *Measuring and assessing ripple in velocity measurement*.

IV. CONCLUSIONS
Interpret your results to arrive at logical conclusions.

1. Step 9 in Section 3.4, *compare low-pass filtering experimental result with calculated*.

5.2 Tips for Report Format

PROFESSIONAL APPEARANCE

- Has cover page with all necessary details (title, course, student name(s), etc.)
- Each of the required sections is completed (Procedure, Results, Analysis and Conclusions).
- Typed.
- All grammar/spelling correct.
- Report layout is neat.
- Does not exceed specified maximum page limit, if any.
- Pages are numbered.
- Equations are consecutively numbered.
- Figures are numbered, axes have labels, each figure has a descriptive caption.
- Tables are numbered, they include labels, each table has a descriptive caption.
- Data are presented in a useful format (graphs, numerical, table, charts, diagrams).
- No hand drawn sketches/diagrams.
- References are cited using correct format.

Quanser Inc.
119 Spy Court
Markham, Ontario
L3R 5H6
Canada
info@quanser.com
Phone: 1-905-940-3575
Fax: 1-905-940-3576

Printed in Markham, Ontario.

For more information on the solutions Quanser Inc. offers, please visit the web site at:
http://www.quanser.com

Waste Electrical and Electronic Equipment (WEEE)

This symbol indicates that waste products must be disposed of separately from municipal household waste, according to Directive 2002/96/EC of the European Parliament and the Council on waste electrical and electronic equipment (WEEE). All products at the end of their life cycle must be sent to a WEEE collection and recycling center. Proper WEEE disposal reduces the environmental impact and the risk to human health due to potentially hazardous substances used in such equipment. Your cooperation in proper WEEE disposal will contribute to the effective usage of natural resources. For information about the available collection and recycling scheme in a particular country, go to ni.com/citizenship/weee.

电子信息产品污染控制管理办法 （中国 RoHS）

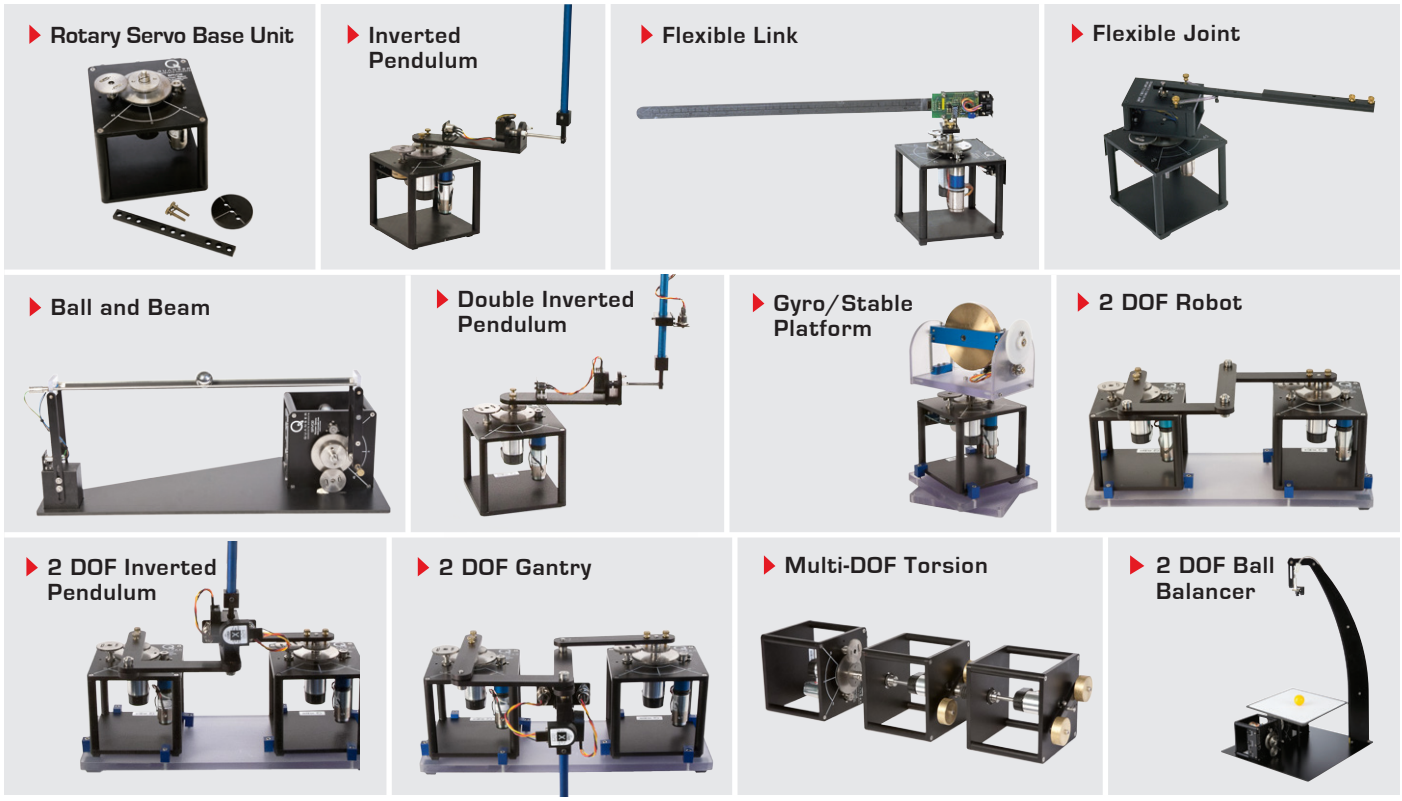
中国客户 National Instruments 符合中国电子信息产品中限制使用某些有害物质命令 （RoHS）。
关于National Instruments 中国 RoHS合规性信息，请登录 ni.com/environment/rohs_china
(For information about China RoHS compliance, go to ni.com/environment/rohs_china)

CE Compliance CE

This product meets the essential requirements of applicable European Directives as follows:

- 2006/95/EC; Low-Voltage Directive (safety)
- 2004/108/EC; Electromagnetic Compatibility Directive (EMC)

Over ten rotary experiments for teaching fundamental and advanced controls concepts



Quanser's rotary collection allows you to create experiments of varying complexity – from basic to advanced. Your lab starts with the Rotary Servo Base Unit and is designed to help engineering educators reach a new level of efficiency and effectiveness in teaching controls in virtually every engineering discipline including electrical, computer, mechanical, aerospace, civil, robotics and mechatronics. For more information please contact info@quanser.com.

QUANSER
INNOVATE · EDUCATE

INFO@QUANSER.COM +1-905-940-3575 QUANSER.COM

Solutions for teaching and research. Made in Canada.