HARDWARE INTERFACING

Topics Covered

- Getting familiarized with the QUANSER® QUBE-Servo 2 Rotary Servo Experiment hardware.
- Using QUARC® to interact with QUBE-Servo 2 system.
- Sensor calibration.

Prerequisites

- The QUBE-Servo 2 has been setup and tested. See the QUBE-Servo 2 Quick Start Guide for details.
- Inertia disc load is on the QUBE-Servo 2.
- You have the QUBE-Servo 2 User Manual. It will be required for some of the exercises.
- You are familiar with the basics of SIMULINK® .

1 Background

1.1 QUARC Software

The QUARC® software is used with SIMULINK® to interact with the hardware of the QUBE-Servo 2 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 QUBE-Servo 2 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 QUBE-Servo 2 User Manual, the QUBE-Servo 2 has a brushed DC motor that is connected to a PWM amplifier. See the QUBE-Servo 2 User Manual for details.

1.3 Encoders

Similar to rotary potentiometers, encoders can also be used to measure angular position. There are many types of encoders but one of the most common is the rotary incremental optical encoder, shown in Figure 1.1. Unlike potentiometers, encoders are relative. The angle they measure depends on the last position and when it was last powered. It should be noted, however, that absolute encoders are available.

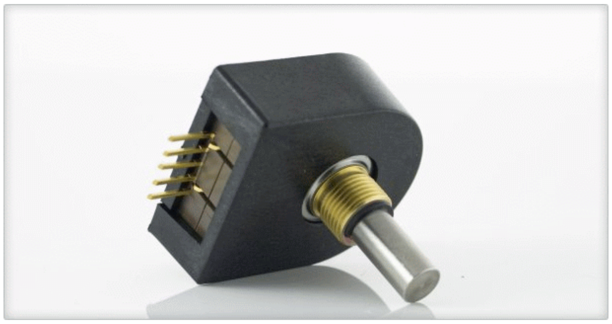

Figure 1.1: US Digital incremental rotary optical shaft encoder

The encoder has a coded disc that is marked with a radial pattern. This disc is connected to the shaft of the DC motor. As the shaft rotates, a light from a LED shines through the pattern and is picked up by a photo sensor. This effectively generates the A and B signals shown in Figure 1.2. An index pulse is triggered once for every full rotation of the disc, which can be used for calibration or homing a system.

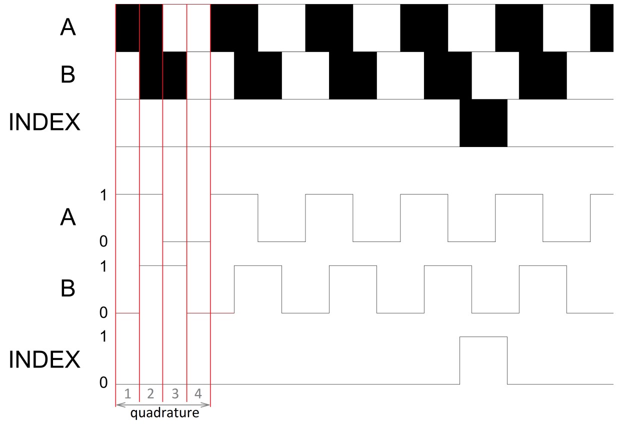

Figure 1.2: Optical incremental encoder signals

The A and B signals that are generated as the shaft rotates are used in a decoder algorithm to generate a count. The resolution of the encoder depends on the coding of the disc and the decoder. For example, a single encoder with 512 lines on the disc can generate a total of 512 counts for every rotation of the encoder shaft. However, in a quadrature decoder as depicted in Figure 1.2, the number of counts (and thus its resolution) quadruples for the same line patterns and generates 2048 counts per revolution. This can be explained by the offset between the A and B patterns: Instead of a single strip being either on or off, now there is two strips that can go through a variety of on/off states before the cycle repeats. This offset also allows the encoder to detect the directionality of the rotation, as the sequence of on/off states differs for a clockwise and counter-clockwise rotation.

2 In-Lab Exercises

In this lab, we will make a SIMULINK® model using QUARC® blocks to drive the DC motor and then measure it's corresponding angle - as shown in Figure 2.1.

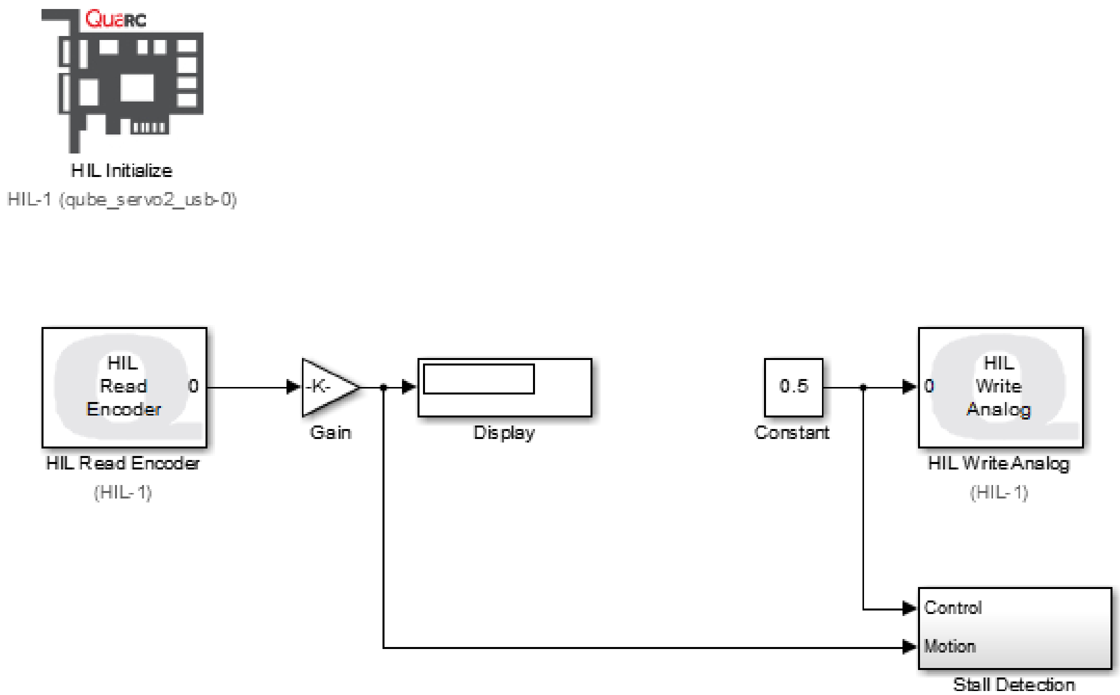

Figure 2.1: Simulink model used with QUARC® to drive motor and read angle on QUBE-Servo 2

2.1 Configuring a Simulink Model for the QUBE-Servo 2

Follow these steps build a SIMULINK® model that will interface to the QUBE-Servo 2 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 2.2.
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. Make sure the QUBE-Servo 2 is connected to your PC USB port and the USB Power LED is lit green.
8. In the *Board type* field, select *qube_servo2_usb*.

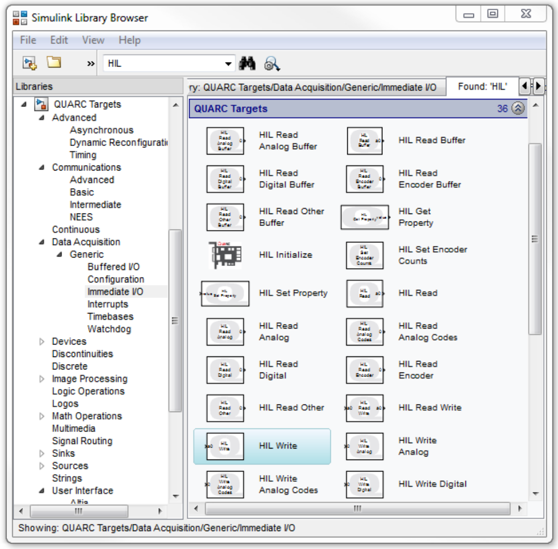

Figure 2.2: *QUARC Targets* in SIMULINK® Library Browser

9. Go to the *QUARC* | *Set default options* item to set the correct Real-Time Workshop parameters and setup the SIMULINK® model for external use (as opposed to the simulation mode).

10. 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.

11. Run the QUARC® controller. To do this, go to the SIMULINK® model tool bar, shown in Figure 2.3, and click on the *Connect to target* icon and then on the *Run* icon. You can also go *QUARC* | *Start* to run the code. The Power LED on the QUBE-Servo 2 (or your DAQ card) should be blinking.

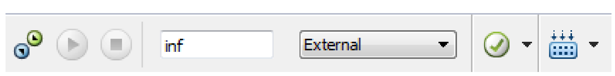

Figure 2.3: SIMULINK® model toolbar: connect to target and compilation

12. 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*).

2.2 Reading the Encoder

Follow these steps to read the encoder:

1. Using the SIMULINK® model you configured for the QUBE-Servo 2 in the previous section, add the HIL Read Encoder block from the *QUARC Targets* | *Data Acquisition* | *Generic* | *Timebases* category in the Library Browser.

2. Connect the HIL Read Encoder to a Gain and Display block similar to Figure 2.1 (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?

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 QUBE-Servo 2 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.

2.3 Driving the DC Motor

1. Add the HIL Write Analog block from the *Data Acquisition* | *Generic* | *Immediate I/O* category into your SIMULINK® diagram. This block is used to output a signal from analog output channel #0 on the data acquisition device. This is connected to the on-board PWM amplifier which 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 2.1.

   Note: We recommend including a Stall Monitor block that is part of the Stall Detection block in Figure 2.1 and Figure 2.4. This block will monitor the applied voltage and speed of the DC motor to ensure that it does not stall. If the motor is motionless for more than 20 s with an applied voltage of over ± 5 V, the simulation is halted to prevent the QUBE-Servo 2 from overheating and subsequent potential damage to the motor.

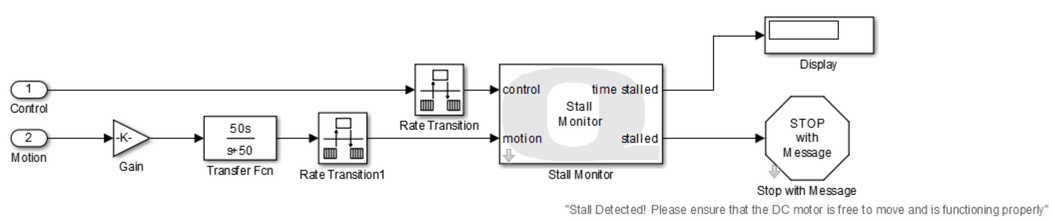

Figure 2.4: Stall Detection Subsystem

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 QUBE-Servo 2. 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. Finally, in what direction does the disc rotate (clockwise or counter-clockwise) when a positive input is applied?

5. Stop the QUARC® controller.

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