Control Systems Laboratory (EE 3321) — Experiment 6

QUARC Software, Hardware Integration, Filtering, and Constant Determination

I. Overview of Experimental Procedure

For this experiment, the student will make a Simulink model using QUARC blocks to drive a DC motor and measure the corresponding angular position or speed. Using the principles learned from that task, the student will then design a model that measures the servo velocity. Additionally, the student will analyze the response of the angular velocity to determine the gain and time constants for the equipment used. This will allow for the student to create a detailed transfer function which models the system, and the student will be able to compare the theoretical and actual responses in real time.

II. Background on the QUARC Software and Hardware

The QUARC software is used with Simulink to interact with the hardware of the QUBE-Servo system. QUARC is used to drive the direct-current (DC) motor and read angular position of the disk. The basic steps to create a Simulink model with QUARC in order to interact with the QUBE-Servo hardware are:

- Make a Simulink model that interacts with your installed data acquisition device using blocks from the QUARC Targets library.
- Build the real-time code.
- Execute the code.

Type `doc quarc` in the MATLAB Command Window to access QUARC documentation and demos.

DC motors use used in a variety of applications; the QUBE-Servo has a brushed DC motor that is connected to a pulse-width modulation (PWM) amplifier. See the *QUBE-Servo User Manual* for details.

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, an example of which can be seen in the figure below. Unlike potentiometers, encoders are relative — the angle that is measured depends on the position when the device is powered up. It should be noted, however, that absolute encoders are available.

FIGURE 6.1: US DIGITAL INCREMENTAL ROTARY OPTICAL SHAFT ENCODER

The encoder has a coded disk that is marked with a radial pattern. As the disk rotates (with the shaft), the light from an LED shines through the pattern and is picked up by a photo sensor. This effectively generates the A and B signals shown in the image below. An index pulse is triggered once for every full rotation of the disk, which can be used for calibration.

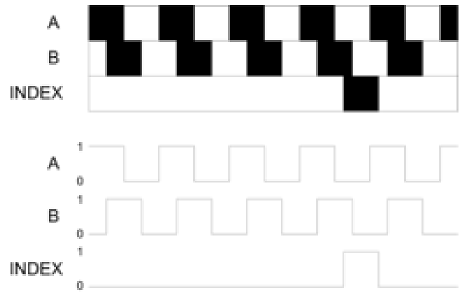

FIGURE 6.2: OPTICAL INCREMENTAL ENCODER SIGNALS

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 disk and the encoder. For example, a single encoder with 512 lines on the disk can generate a total of 512 counts for every full revolution of the encoder shaft. However, in a quadrature decoder, as shown on the previous page, 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 are 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 for a clockwise rotation differs from that of a counterclockwise rotation.

III. Filtering Theory

A low-pass filter is used to block out, known as filtering, the high-frequency components of a signal. A first-order low-pass filter transfer function has the form

$$G(s) = \frac{\omega_f}{s+\omega_f}, \tag{6.1}$$

where ω_f is the cut-off frequency of the filter in radians per second. All higher frequency components of the signal will be attenuated by -3 dB (or approximately 50%) or more.

IV. Bump Test Modeling

A bump test is a simple test 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 transfer function

$$\frac{Y(s)}{U(s)} = \frac{K}{1+s\tau}. \tag{6.2}$$

The following step response is generated using this transfer function with K = 5 (rad/Vs) and τ = 0.05 (s).

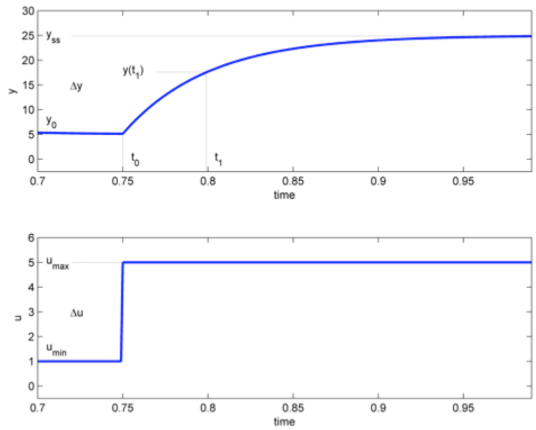

FIGURE 6.3: INPUT AND OUTPUT SIGNAL USED IN THE BUMP TEST METHOD

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

$$K = \frac{\Delta y}{\Delta u} = \frac{y_{SS} - y_0}{u_{max} - u_{min}}. \tag{6.3}$$

The time constant, τ, is defined as the time it takes for the first-order system output to change by $1 - 1/e$, or by approximately 63.2%, of its steady-state response (i.e., $y_{SS} - y_0$), given that the output change is in response to the application of a step input.

Keeping in mind the QUBE-Servo system (with transfer function given in Section IV), the Laplace transform of an input step voltage with amplitude A_V and time delay t_0 is given as

$$V_m(s) = \frac{A_V e^{-st_0}}{s}. \tag{6.4}$$

By multiplying the system transfer function by $V_m(s)$, we can obtain the Laplace transform of the theoretical angular speed as a function of the given input voltage, or

$$\omega_m(s) = \frac{K A_V e^{-st_0}}{(1+s\tau)s}. \tag{6.5}$$

The motor speed step response can then be determined in the time domain by taking the inverse-Laplace transform of this equation, noting any initial angular velocity.

V. Configuring a Simulink Model for the QUBE-Servo

The following steps should be used to build a Simulink model that will interface with the QUBE-Servo using QUARC:

- Make sure the QUBE-Servo is connected to the PC through the USB port, and that it is powered on (the Power LED should be lit).
- Open MATLAB Simulink, and create a new Simulink diagram.

- Open the Simulink Library Browser, and expand the QUARC Targets item. From the **Data Acquisition >> Generic >> Configuration** folder, choose the **HIL Initialize** block. Drag this block into the Simulink model. This block is used to configure your data acquisition device. Use the image below for reference.

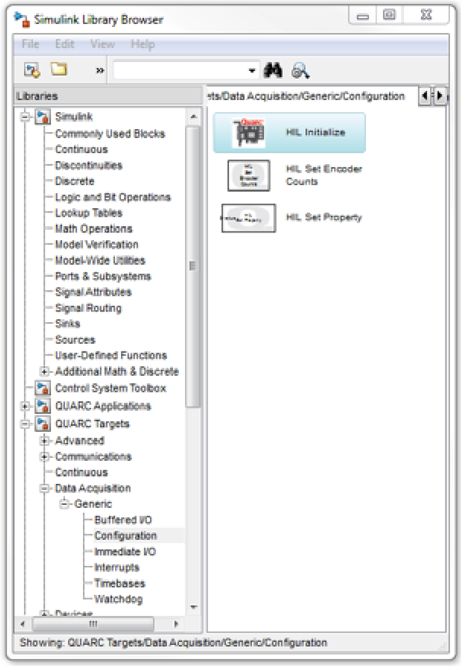

FIGURE 6.4: QUARC TARGETS IN SIMULINK LIBRARY BROWSER

- Double-click on the HIL Initialize block. In the Board type field, select **qube_servo_usb**.
- Use 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).
- Select **QUARC >> Build**. 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.
- Run the QUARC controller. To do this, go to the Simulink model toolbar and click on the **Connect to Target** icon and then the **Run** icon. You can also select **QUARC >> Start** to run the code. The Power LED on the QUBE-Servo should be blinking.
- If you have successfully ran the QUARC controller without any errors, you can stop the code by clicking the **Stop** icon in the toolbar, or by selecting **QUARC >> Stop**.

VI. Reading the Encoder

The following steps should be used to read the encoder:

- Using the Simulink model created in the last section, add the **HIL Read Encoder Timebase** block from the **QUARC Targets >> Data Acquisition >> Generic >> Timebases** folder in the Library Browser.
- Connect the HIL Read Encoder to a Gain block; connect the output of the Gain block to a Display block.
- Build the QUARC controller. The code needs to be regenerated because the Simulink diagram has been modified since the last build. Run the QUARC controller.

- Rotate the disk back and forth. The Display block shows the number of counts measured by the encoder, which are proportional to the angle of the disk.
- Stop QUARC, move the disk around, and restart the QUARC controller. **Record** your observation about the encoder measurement when QUARC is restarted.
- Set the Gain block to a value that converts counts to degrees. Do this by noting the number of counts for a full revolution. This is called the *sensor gain*. Run the QUARC controller and confirm that the Display block shows the angle of the disk correctly.
- Stop the QUARC controller.

VII. Driving the DC Motor

The following steps should be used to drive the DC motor:

- Using the same Simulink model from the previous section, add the **HIL Write Analog** block from the **QUARC Targets >> Data Acquisition >> Generic >> Immediate I/O** folder in the Library Browser. This block is used to output a signal from analog output channel #0 on the data acquisition device to the on-board PWM amplifier, which drives the DC motor.
- Add a Constant block to the Simulink model, connecting it to the HIL Write Analog block.
- Include a Stall Monitor block in the main model or a subsystem. This block will monitor the applied voltage and speed of the DC motor to ensure that if the motor is motionless for more than 20 seconds with an applied voltage exceeding 5 volts, the simulation is halted to prevent damage to the motor.
- Build and run the QUARC controller. Set the Constant block to 0.5; this applies 0.5 volts to the DC motor. Confirm that a *positive measurement* is obtained when a *positive signal* is applied. This convention is important, especially in control systems when the design assumes the measurement increases when the input also increases. Finally, note the direction of disk rotation (clockwise or counter clockwise) for a positive input. Adjust the Control block to be a negative value and verify the disk rotates in the opposite direction.
- Stop the QUARC controller.

VIII. Measuring the Servo Velocity

The following steps should be used to measure the servo velocity using the encoder:

- Using the same Simulink model from the previous section, remove the Display and Constant blocks. Change the encoder calibration gain measure the gear position in radians (instead of degrees).
- Insert an **Inverse Modulus** block between the HIL Read Encoder Timebase block and the Gain block. Since the QUBE-Servo DAQ has 16-bit counters, the valid count range is from -2^{15} to $2^{15} - 1$. To eliminate the discontinuous jump that would occur when the encoder reaches the limits of this range, add an Inverse Modulus block from the **QUARC Targets >> Discontinuous** folder into the Simulink diagram. Set the modulus to the total buffer size, or 2^{16}.
- Attach a **Derivative** block to the encoder calibration gain output to measure the gear speed using the encoder (in radians per second).
- Connect the output of the Derivative block to a **Scope**.

- Create a setup that injects a square wave with limits of 1 volt to 3 volts at 0.4 Hz to the HIL Write Analog block. Note that this is referred to as a **bump test**. Build and run the QUARC controller. Examine the encoder speed response, paying close attention to the level of noise shown on the scope.
- Stop the QUARC controller.
- One way to remove some of the high-frequency components (otherwise considered noise) of the angular velocity signal is by adding a low-pass filter to the derivative output. Insert a **Transfer Function** block after the Derivative block, connecting the output of the low-pass filter to the scope. Set the transfer function to be 50/(s+50).
- Build and run the QUARC controller. Compare the output of the derivative to the output of the low-pass filter. Vary the cutoff frequency between 10 and 200 (rad/s), and note the effect of this change on the filtered response. Consider the benefit and the trade-off of using a filter such as this.

IX. Experimental Procedure

1. Complete the steps listed in Section V of this manual.
2. Complete the steps listed in Section VI of this manual. Use the image below for reference.
3. Complete the steps listed in Section VII of this manual. Use the image below for reference.

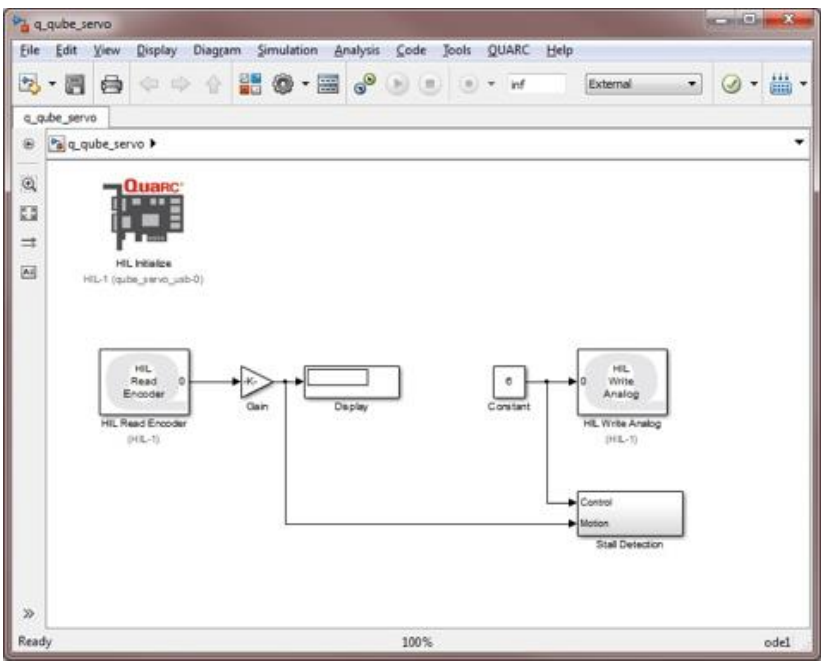

FIGURE 6.5: SIMULINK MODEL USED WITH QUARC TO DRIVE MOTOR AND READ ANGLE ON QUBE-SERVO

4. Complete the steps listed in Section VIII of this manual. Use the image below for reference. Note that the equipment will be used first without the transfer function shown in the image.

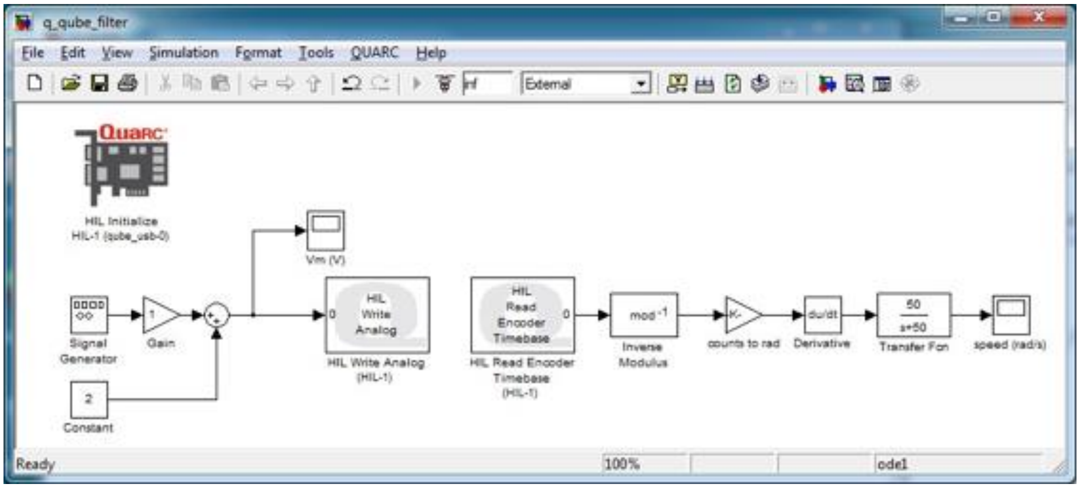

FIGURE 6.6: MEASURING SPEED USING THE ENCODER

5. Using the results of Procedure 4, create a transfer function that represents the QUBE-Servo transfer function for a voltage input and an angular speed output. Use the values provided in Section IV to begin, and adjust until you find a good fit. Compare the output of the transfer function to the true measured speed of the QUBE-Servo. (Use a MUX block to overlap the display for both signals on the scope.) Try using other input signals, such as sinusoids. Compare the phase of the voltage input to the phase of the outputs (theoretical and measured).

VII. Conclusion

Answer the following questions:

- What happens to the encoder reading every time the QUARC controller is started?
- What is the advantage of using a low-pass filter on the measured speed of the servo? Are there any disadvantages? For what cutoff frequency did you find to be the best tradeoff between these advantages and disadvantages?
- Explain, in your own words, the purpose of conducting the bump test. What does the results of this test tell us about the system?
- How does this system react to sinusoidal inputs? Using the transfer function from Procedure 5, determine the theoretical phase lag of the system.