Hemant Arora: 19 Warun Gumarawell: 96

ELECENG 3CL4: Introduction to Control Systems Lab 1: Introduction and Familiarization with Lab Equipment

Dr. Tim Davidson and Dr. Shahrukh Athar
Ext. 24818, 26503
davidson@mcmaster.ca, athars3@mcmaster.ca

Objective

To establish safety protocols, introduce the laboratory equipment, measure the angle and speed of a DC motor, and examine filtering effects.

Assessment

This laboratory is conducted in groups of no more than two students. The students are required to attend their assigned lab section. The assessment of this lab will occur during the lab itself through Q & A sessions with the Teaching Assistants. Components where you need to have evaluations done by the TA are mentioned in blue color throughout this Lab manual. You will earn a maximum of 100 marks from Lab 1 activities. **Lab 1 will constitute 5% of your total grade for this course.**

- The first component is to carefully read the lab manual.
- The second component is to demonstrate to the TAs that you have understood the specific safety information for the ELECENG 3CL4 labs (see Section 2). You will do this by showing to the TAs that you have completed the ECE Lab Safety Quiz on Avenue. If you do not successfully complete this quiz, you will not be allowed to proceed with the remainder of this laboratory, nor the remaining laboratories.
- The third component is to demonstrate to the TAs that you have familiarized yourself with the equipment. You will do this by going through the familiarization information about the laboratory equipment (Section 3) and conducting the laboratory experiments (Section 4). You will demonstrate your work for Parts 1-4 in Section 4 to the TAs and answer the TAs questions to receive marks for this lab.
- This lab does not have a Pre-Lab nor a Lab Report.

1 Description of Laboratory Equipment

In the laboratories of this course, we will deal with a closed-loop angular positioning system based around a DC motor. Such systems are often used to position heavy or difficult to move

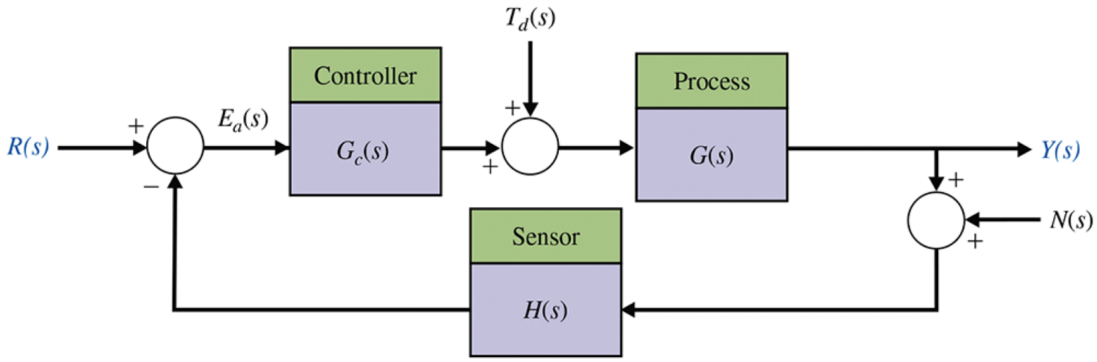


Figure 1: Feedback system with $y(t) = \theta(t)$. In our models, the sensor will be assumed to have a constant gain over the frequencies of interest. (Figure 4.4 of Dorf and Bishop, *Modern Control Systems*, 14th edition, Pearson, 2022.)

objects using a 'command tool' that is easy to move, in which case they are often called *servomechanisms*. One example of a servomechanism is that involved in moving the control surfaces of an aircraft using a lever in the cockpit. The goal of this lab is to introduce the positioning system that will be used in all the laboratories in the course, perform some angle and speed measurements, study filtering effects on the measured variables, and establish the appropriate safety protocols.

A block diagram of the systems that we will develop in the laboratories is illustrated in Figure 1. The "*process*" (at times also called the "*plant*") that we wish to control is the motor and its associated electronics. We will use the function $\theta(t)$ to denote the output of the motor, rather than the function $y(t)$ in Figure 1, because the output of the system is the angular position of the motor shaft measured by an incremental optical encoder which is the sensor. In this case, the sensor can be assumed to have an ideal all-pass transfer function $H(s) = 1$. The broad goal of the series of labs in this course will be to design controllers that ensure that the feedback system as a whole performs in desirable ways.

As mentioned above, the "*process*" that we wish to control is the motor and its associated electronics. As shown in Figure 2, the input to the process is a control voltage, denoted by $x(t)$, and the output of the process is the angular position of the shaft, denoted by $\theta(t)$, and measured by an optical encoder. In Lab 1, students will be introduced to the lab equipment by focusing on the *process* (i.e., the motor) and the sensor connected to it (the optical encoder). Thus, we will focus on Figure 2 in Lab 1 while systems of the form of Figure 1 will be the focus of future labs. More detailed resources about the Quanser Qube-Servo 3 hardware that we are using are provided on Avenue (Content > Laboratory > Lab Resources). All students are encouraged to go through them.

For the purposes of our ELECENG 3CL4 labs, we will assume a linear model for the process. Using information about the structure of the motor and rotational Newtonian mechanics, the linear model for the operation of the motor can be described by the following differential equation:

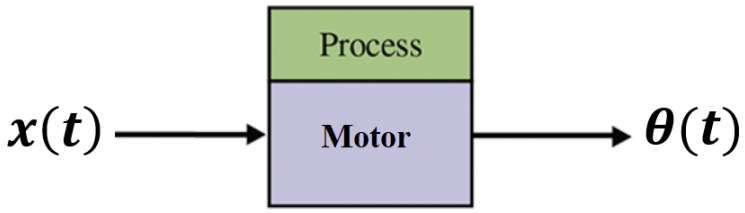


Figure 2: Focus of Lab 1: To learn more about the *process*, i.e., the motor. The input to the process is a control voltage, $x(t)$, and the output of the process is the angular position of the shaft, $\theta(t)$. (Figure adapted from Dorf and Bishop, *Modern Control Systems*, 14th edition, Pearson, 2022.)

$$J\frac{d^2\theta(t)}{dt^2} + b\frac{d\theta(t)}{dt} = K_m x(t), \tag{1}$$

where J is the rotational inertia of the motor, b is the coefficient of viscous friction in the motor structure, and K_m is the (internal) gain. Taking the Laplace transform of both sides of Eq. (1), for a system initially at rest, we obtain the transfer function of the process:

$$s^2 J\Theta(s) + sb\Theta(s) = K_m X(s) \tag{2}$$

$$\implies G(s) = \frac{\Theta(s)}{X(s)} = \frac{A}{s(s\tau_m + 1)}, \tag{3}$$

where $A = K_m/b$ and $\tau_m = J/b$. In typical industrial applications, the gain A and the time constant τ_m are unlikely to be known in advance. Therefore, in the second laboratory we will develop experimental techniques by which A and τ_m can be estimated. In the laboratories which follow that one, we will design controllers for the servomechanism that provide desirable performance characteristics.

2 Departmental Safety Information

For the safety of everyone in the laboratory, it is critical that everyone is fluent in the Departmental laboratory safety procedures and in the specific procedures for the EE3CL4 laboratories. To that end, the first activity for the laboratory is to:

- Read and understand the Departmental laboratory safety information, which is available at: `https://www.eng.mcmaster.ca/ece/resources/health-safety-labs/`. Please ask the instructor or a TA if you have any questions or concerns.
- Read the "*ECE Lab Safety*" document located at: Avenue > Laboratory > ECE Lab Safety, and documents that it mentions.

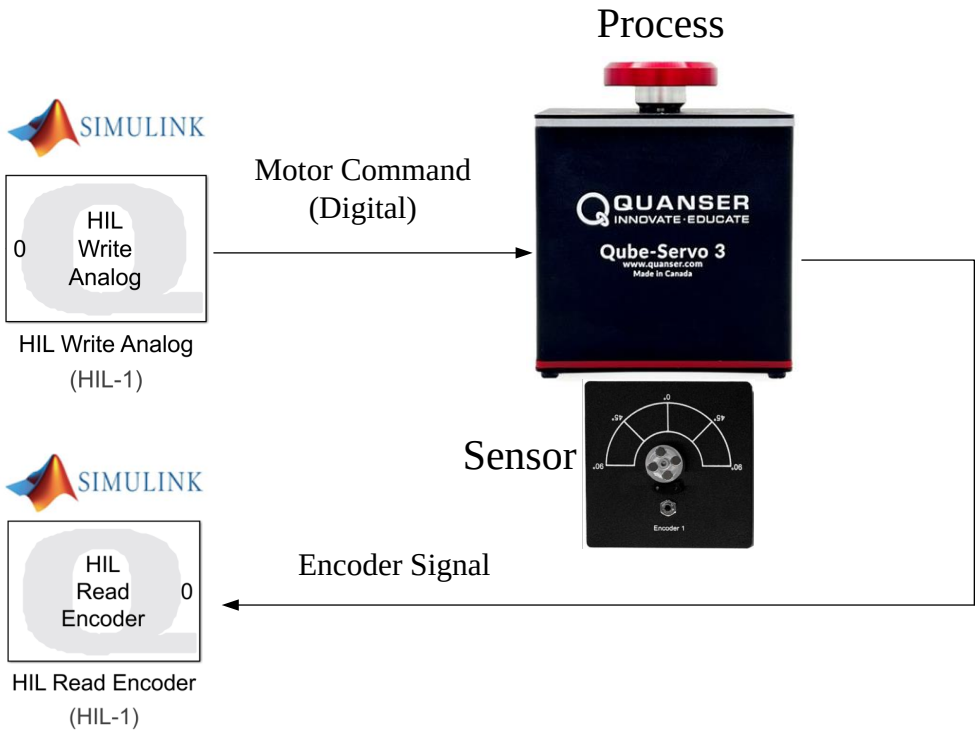


Figure 3: Reading and writing signals between Simulink and the *Process* (motor and load). (This figure has been adapted from the Quanser *Qube-Servo 3 Experiment User Manual*.)

- After going through the above-mentioned resources, complete the Departmental laboratory safety quiz, which is available on Avenue to Learn (Assessments > Quizzes > ECE Lab Safety Quiz). Each student is expected to obtain 100% marks in the quiz (five attempts are allowed).

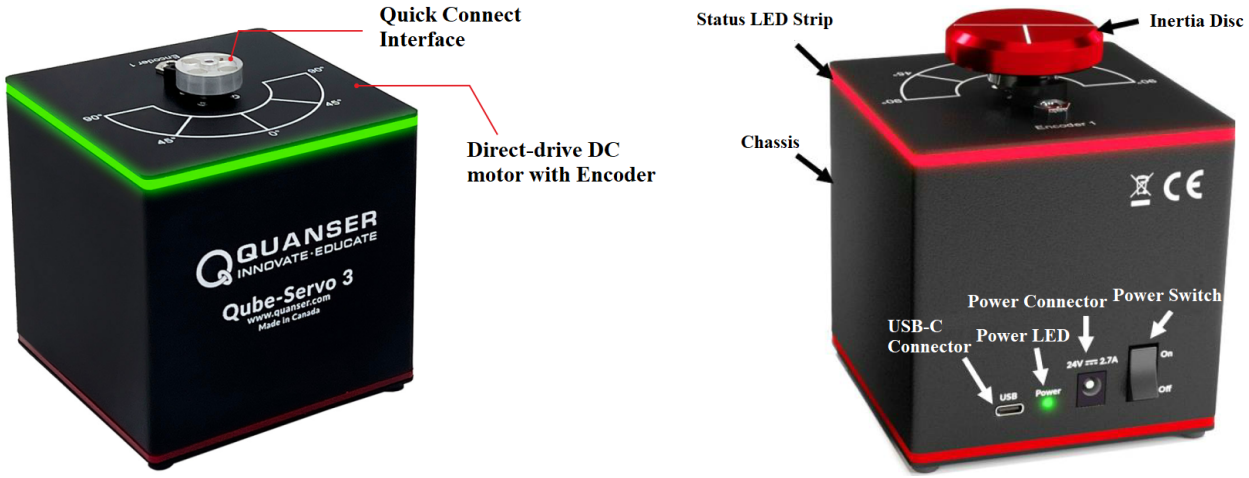
You will not be allowed to proceed with this or any other labs if you have not completed the safety quiz. Both students in a group must show to the TAs that each of them has completed the ECE Lab Safety Quiz on Avenue.

3 Familiarize yourself with the Equipment

3.1 Equipment List

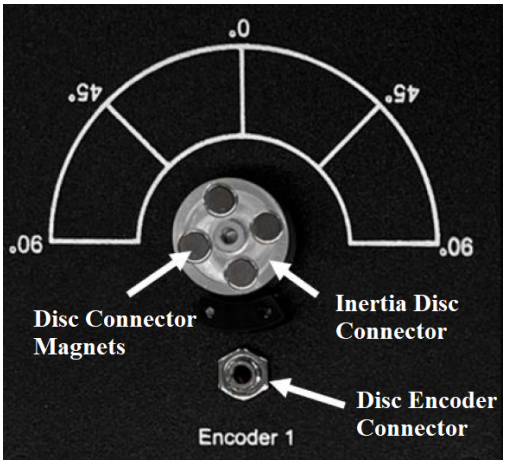
In our laboratory experiments, we are going to employ Quanser's Qube-Servo 3 system to provide the process and measurement modules, and employ Simulink in MATLAB to design the controllers that will send control commands to a DC motor (and its load) as shown in Figure 3.

a) **Simulink Modeling Environment**: The cascade combination of the motor and its electronic drive system can be modeled in the continuous-time domain and represented



(a) Front View

(b) Back View



(c) Top View (with inertia disc removed)

Figure 4: Components of the Qube-Servo 3 System. (This figure has been adapted from the Quanser *Qube-Servo 3 Experiment User Manual*.)

by a transfer function in the Laplace domain. However, all controllers in these labs will be implemented in MATLAB/Simulink environment on a personal computer. Simulink is one core component in MATLAB that is used for model-based design and simulation. This means that the control systems for the course experiments will be designed on a digital computer. This implementation, in principle, requires methods of analysis and design from the theory of discrete-time control systems. This is due to the fact that the control signal applied to the system is held constant between consecutive *sampling times* using a zero-order-hold operator (a digital-to-analog converter). The control computations are also based on a feedback of the system output at the discrete sampling times. While there is a well-developed theory for analysis and design of such mixed discrete/continuous-

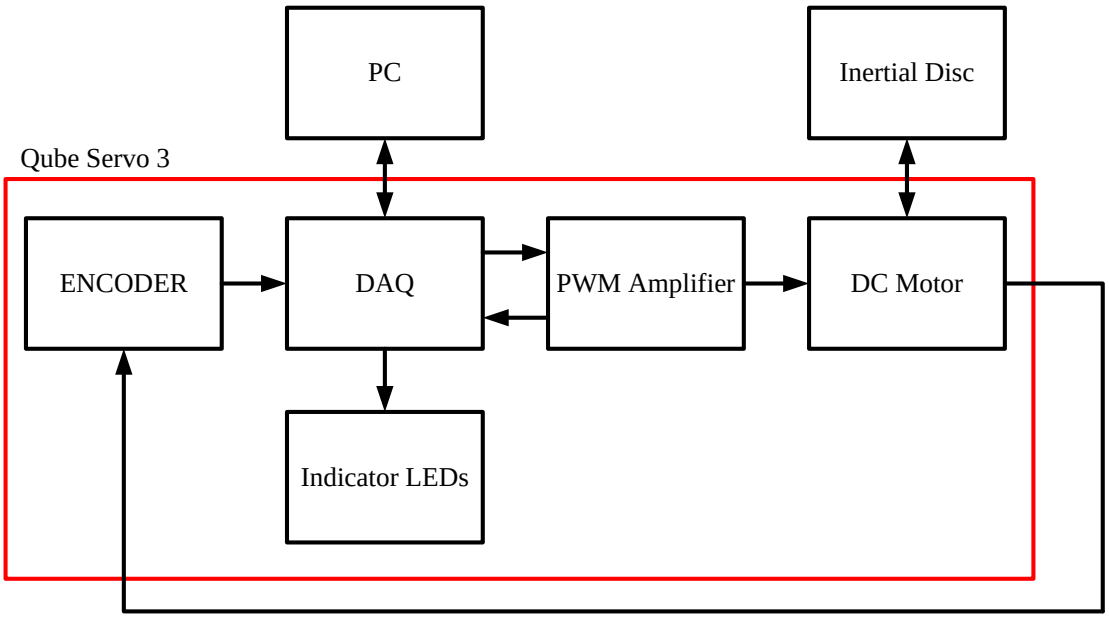
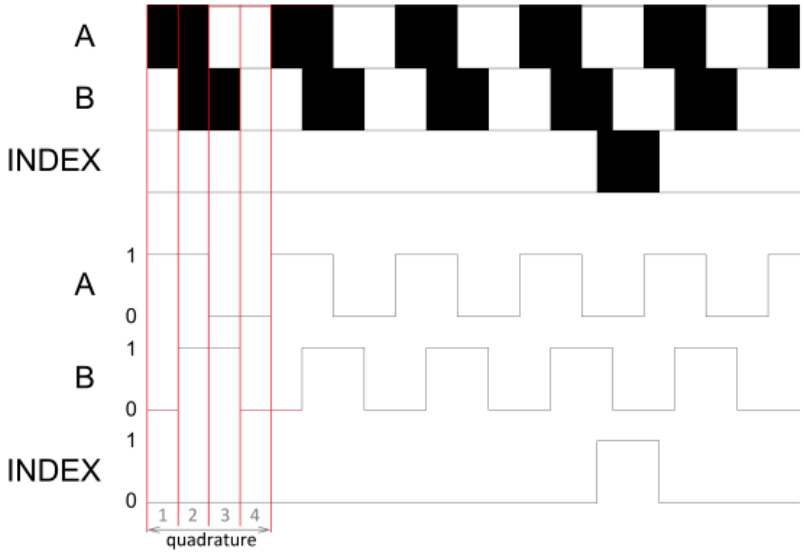


Figure 5: Interaction between Qube-Servo Components. (This figure has been adapted from the Quanser *Qube-Servo 3 Experiment User Manual*.)

time systems, in this course and its labs we will only deal with continuous-time control systems. To mitigate any potential performance and stability issues arising from the application of the continuous-time control techniques to this mixed system, we choose a very small sampling time, which would mean that sampling is quite fast compared to the dominant mechanical dynamics of our process and expected bandwidth of the closed-loop feedback control system. Furthermore, due to the resolution of the analog-to-digital and digital-to-analog converters being employed by the Qube-Servo 3 system, the quantization errors induced by this conversion are small compared to other sources of noise in our system. Therefore, any approximation errors due to the computer-based implementation of continuous-time controllers can be safely neglected. This will be our assumption throughout all the laboratory experiments in this course.

b) **Qube-Servo 3 System**: The Quanser Qube-Servo 3, pictured in Figure 4, is a compact rotary servo system that can be used to perform a variety of classic servo control experiments. The Qube-Servo 3 allows control by a computer via USB connection. The system is driven using a 24 V direct drive brushed DC motor. The motor is powered by a built-in PWM amplifier with integrated current sense. An inertial disc (shown in red in Figure 4b) is supplied with the system, which can be easily attached using magnets mounted on the Qube-Servo 3 module connector (Figure 4c). While the disc provides a visual indication of the angular position of the DC motor, inside the box single-ended rotary encoders are used to measure this angular position. The angular velocity of the motor can also be measured using an integrated software-based tachometer.

(a) US Digital incremental rotary optical shaft encoder.



(b) Optical incremental encoder signals.

Figure 6: The incremental encoder operation. (Figures courtesy of Quanser.)

The interaction between the different system components on the Qube-Servo 3 is illustrated in Figure 5. On the data acquisition (DAQ) device block, the motor encoder is connected to the Encoder Input (EI) channels #0 and #1. The Analog Output (AO) channel is connected to the power amplifier command, which then drives the DC motor. The DAQ Analog Input (AI) channel is connected to the PWM amplifier current sense circuitry. The DAQ also controls the integrated tri-color LEDs.

c) **Data Acquisition Card (DAQ)**: The Qube-Servo 3 system includes a DAQ device with two 24-bit encoder channels with quadrature decoding and one analog output channel. The DAQ also incorporates a 12-bit ADC which provides current sense feedback for the motor, which enables the detection of stalls. Stall warnings can be sent using digital I/O.

d) **Incremental Encoder**: The Qube-Servo system includes a rotary incremental optical encoder (Figure 6a) which is used to measure the angle of the DC motor. 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 6b. An index pulse is triggered once for every full rotation of the disc, which can be used for calibration or homing a system. 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 6b, 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 differs for a clockwise and anti-clockwise rotation.

4 Laboratory Procedures

The laboratory work for this lab has four main parts. As you work through the laboratory procedures below, critically analyze your work and verify it for correctness. This lab does not have a lab report and the entire grade of this lab will be decided based on the demonstration of your work to the Teaching Assistants (TAs) within the allocated lab time.

4.1 Part 1: Reading from the Encoder

Goal: In this part our goal is to read the output of the encoder that is associated with the Qube-Servo 3 and convert the encoder count to angular position of the motor.

1. Download the *Simulink starter file* from Avenue (Content > Laboratory > Simulink Starter File > EE3CL4_Lab.slx). You should create a separate folder for each part of this lab on the computer. For now, place the starter file in the Part 1 folder.

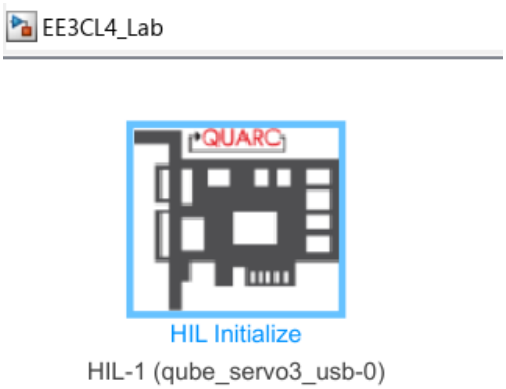


Figure 7: A view of the initial Simulink model file.

2. Rename the above-mentioned file and open it from within MATLAB. There is only one block in the Simulink file shown in Figure 7, namely the *HIL Initialize* or *initialization* block. The purpose of this block is to initialize the Simulink environment so that it can correctly interface with the Qube-Servo 3 DC motor. It is executed automatically once you run the Simulink application.

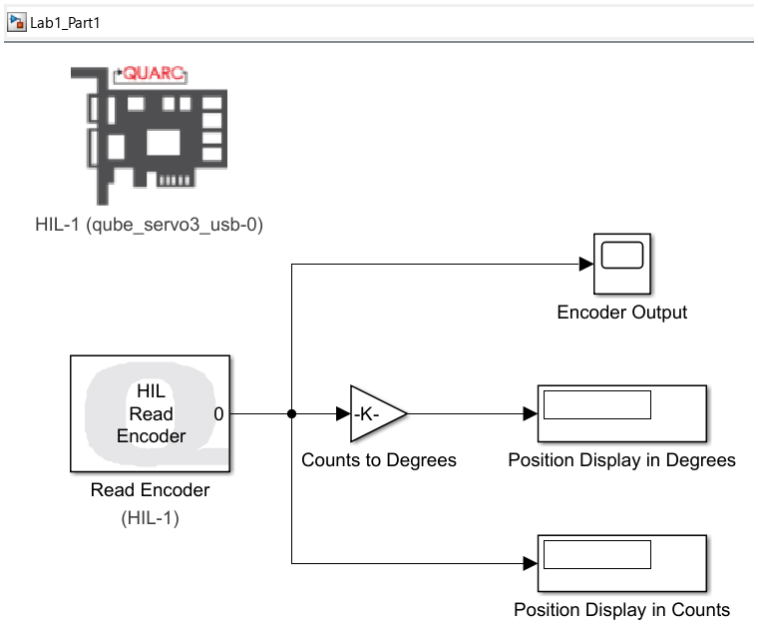


Figure 8: Target Simulink Model for Part 1. Note: The Simulink notation for the gain block, i.e., -K-, does not mean that the gain K is negative. As discussed in Part 1 Step 6, the gain is $\frac{360}{2048}$.

3. Double-click the *HIL Initialize* block and make sure that the *Board type* is set as "qube_servo3_usb" (our motor model).

4. To read the encoder output and convert it to angular position in terms of degrees, we need to build the Simulink model shown in Figure 8. The next three steps outline how to build this model.

5. In the Simulink model window, go to the *Simulation* tab and open the *Library Browser*. Then go to the following library: QUARC Targets > Data Acquisition > Generic > Immediate I/O. Find the *HIL Read Encoder* block and drag-and-drop it into the modeling space. **The *Read Encoder* block returns the measured angular position of the motor shaft in terms of an encoder count. This is the output of the plant/process, i.e., our motor.**

6. Next, go to the *Library Browser* and then to the library: Simulink > Commonly Used Blocks and drag-and-drop the *Gain* block into the modeling space. Since in a single rotation of the motor shaft, i.e., from an angle of 0^o to 360^o, the encoder goes through a count of 2048, counts can be converted to degrees by setting the *Gain* of this block to $\frac{360}{2048}$. This can be done by double-clicking the *Gain* block, setting the gain to this value, and then pressing OK.

7. Now, go to the *Library Browser* and then to the library: Simulink > Sinks and drag-and-drop the *Display* block into the modeling space twice. One of these blocks will

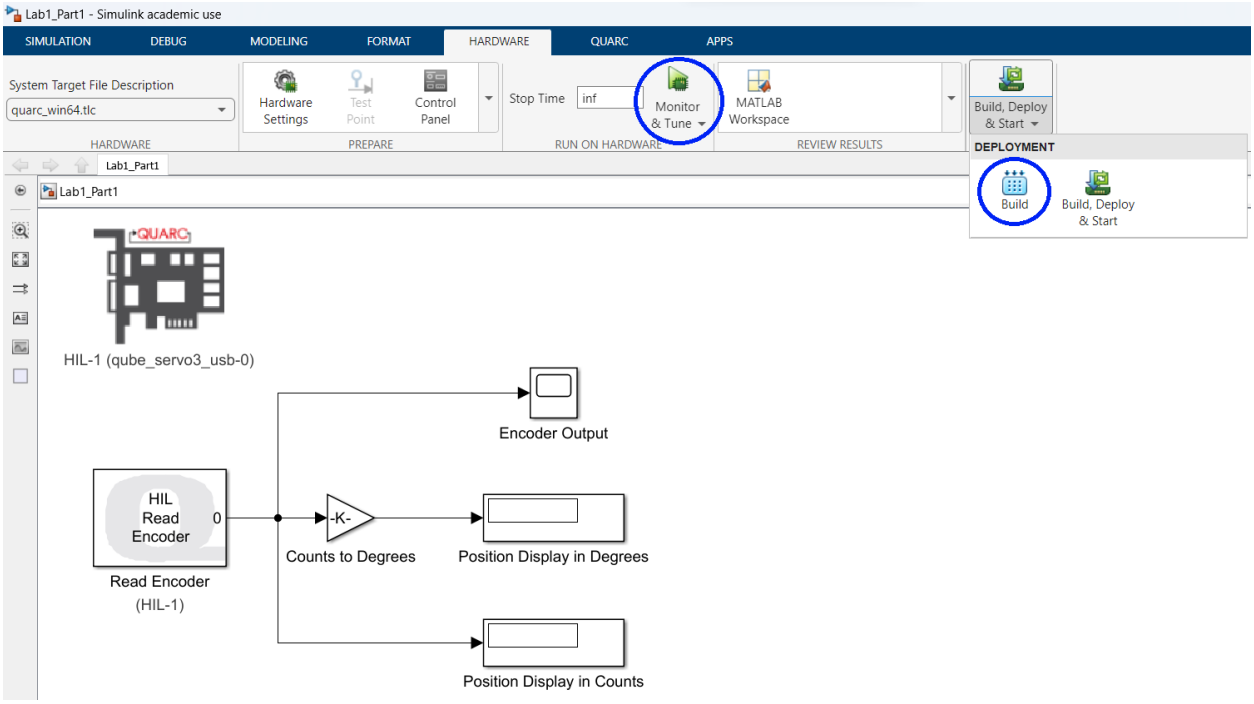


Figure 9: A view of the Simulink Model window for Part 1 with *Build* and *Monitor & Tune* buttons highlighted.

be used to display the output of the *Gain* block, which is *angular position* in terms of *degrees* while the other will be used to display the output of the *Read Encoder* block, which is the *Encoder* output, i.e., counts. From this same library also drag-and-drop the *Scope* block into the modeling space. This block will be used to view the output of the *Read Encoder* block.

8. Connect the five blocks through wires exactly as shown in Figure 8. You should also name each block as shown in Figure 8. This can be done by double-clicking the name beneath a block. At this point your model should look like Figure 8. Save your work by going to the *Simulation* tab and pressing the *Save* button or using the keyboard shortcut (Ctrl + S).

9. Now that we have made the model, we need to generate the code and run it. Open the "Hardware" tab shown at the top of Figure 9. Now build your Simulink model, i.e., press the "Build" button (shown in Figure 9 and accessible after pressing the down-arrow on the "Build, Deploy & Start" button). This will generate the corresponding code for each block in the model which can then be executed.

10. At this point, turn ON the Qube-Servo 3 motor block by using the ON/OFF button at its back. The *Power* LED should turn green and the top LED bar should turn red. You should also align the engraved line on the red circular load with the 0^o mark at the top of the Qube-Servo 3.

11. Next, in your Simulink model, press the "Monitor & Tune" button (shown in Figure 9) to run the model on the motor setup. If this step is successful, the top LED bar on the Qube-Servo 3 will turn green indicating that it is now in your Simulink model loop.

12. Open the scope by double-clicking it in your model. With your hand, rotate the red load disk on the Qube-Servo 3 very slowly in both directions and make the following three observations:

    (a) Observe the output of the *Read Encoder* block connected to the display. You should be able to see the encoder count. What happens to this count when you rotate the motor clockwise? What happens to this count when you rotate the motor anti-clockwise? clockwise is pos, anti-clockwise is neg

    (b) On the scope, which is showing the output of the *Read Encoder* block, zoom in the scope signal and observe the step-wise changes in its values. Note the equal heights of these steps as you turn the motor shaft slowly. You may need to re-scale the Y-axis limits by pressing the appropriate button in the scope window.

    (c) Observe the output of the *Gain* block connected to the corresponding display. You should be able to see the angular position in degrees. What happens to the angle when you rotate the motor clockwise? What happens when you rotate it anti-clockwise? What do you see when you rotate the motor shaft clockwise beyond a single rotation? In this case, does the angle reset to 0^o or go beyond 360^o clockwise is pos, anti-clockwise is neg

13. At this point, you should call the Teaching Assistant to have your Part 1 evaluated.

14. Press the *Stop* button in the "Hardware" tab to stop the hardware execution of your model. Turn OFF the Qube-Servo 3 motor.

4.2 Part 2: Driving the DC Motor

Goal: In Part 1, we rotated the DC motor by hand. In this part our goal is to drive the Qube-Servo 3 motor through Simulink by giving a constant voltage signal to the motor. We will also read the motor output, i.e., its angular position, as we did in Part 1.

1. For this part, continue working with the Simulink Model developed in Part 1. In fact, to run the motor through Simulink by giving a constant voltage signal, we only need to add two more blocks to the Part 1 model which are depicted within the blue box in Figure 10 (which shows the target model to be developed in Part 2). The next two steps discuss adding these blocks.

2. Go to the *Library Browser* in the *Simulation* tab and then to the library: Simulink > Commonly Used Blocks and drag-and-drop the *Constant* block into the modeling space. Double-click the *Constant* block and set its value to 0.5. This will allow us to give a voltage of 0.5 V to the motor.

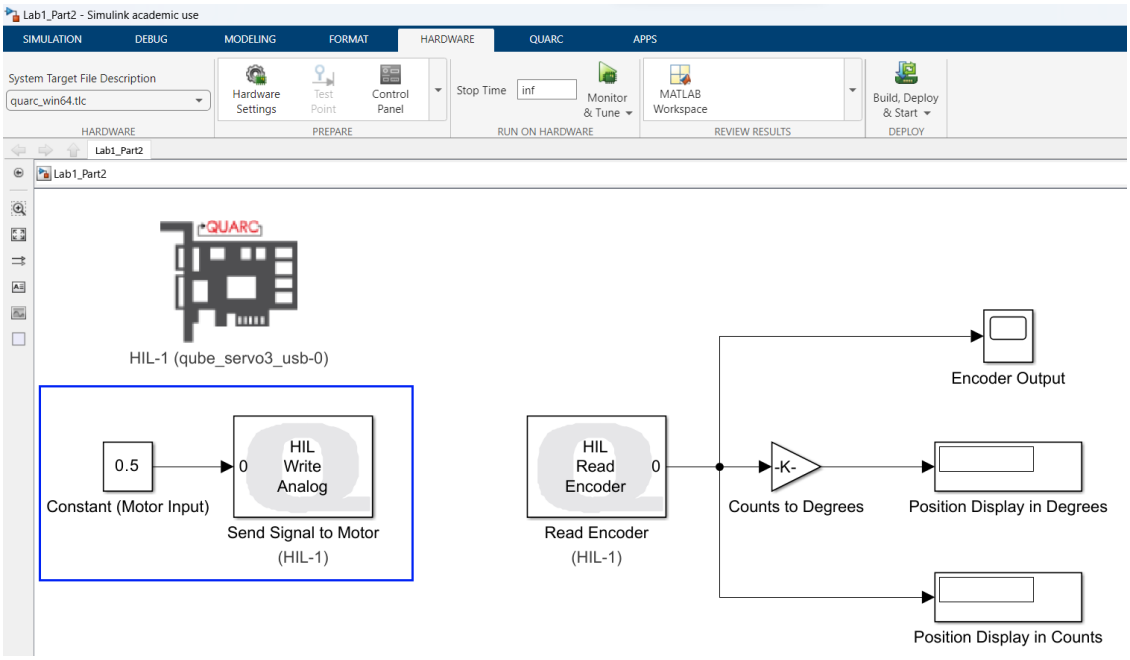


Figure 10: Target Simulink Model for Part 2.

3. Next, we need to convert this input into an analog signal and then send it (write it) to the motor. Again open the *Library Browser* and go to the following library: QUARC Targets > Data Acquisition > Generic > Immediate I/O. Find the *HIL Write Analog* block and drag-and-drop it into the modeling space. **The *HIL Write Analog* block sends the input signal from Simulink to our plant/process, i.e., our motor.**

4. Wire the *Constant* block and the *HIL Write Analog* block together. At this point your model should look like Figure 10. Save your work by going to the *Simulation* tab and pressing the *Save* button or using the keyboard shortcut (Ctrl + S).

5. To generate the code for the model, open the "Hardware" tab and press the "Build" button (shown in Figure 9 and accessible after pressing the down-arrow on the "Build, Deploy & Start" button). This will generate the corresponding code for each block in the model which can then be executed.

6. At this point, turn ON the Qube-Servo 3 motor block by using the ON/OFF button at its back. The *Power* LED should turn green and the top LED bar should turn red.

7. Next, in your Simulink model, press the "Monitor & Tune" button (shown in Figure 9) to run the model on the motor setup. If this step is successful, the top LED bar on the QUBE-Servo 3 will turn green indicating that it is now in your Simulink model loop and the motor will start spinning. Open the scope (connected to the output of the motor) by double-clicking it in your model.

8. Make the following four observations:

(a) With the amplitude of the motor input signal set to +0.5, is the motor rotating in the clockwise or anti-clockwise direction? clockwise

(b) The output of the *Read Encoder* block, which is representing the motor's angular position in terms of counts, is connected to a scope. Open this scope and observe the slope of the motor angular position. Is this what you expect? Why? Yes, it is a line with constant slope

(c) What can you say about the motor angular velocity? Constant velocity

(d) While the motor is in operation, change the amplitude of the motor's input signal to −0.5. This can be done by opening the *Constant* block and setting its value to −0.5 and then pressing OK. What impact does doing so have on the direction of the motor's rotation and the readings in the two *Display* blocks? It goes anti-clockwise

9. At this point, you should call the Teaching Assistant to have your Part 2 evaluated.

10. Press the *Stop* button in the "Hardware" tab to stop the hardware execution of your model. Turn OFF the Qube-Servo 3 motor.

4.3 Part 3: Measuring the Motor Speed

Goal: In Part 2, we developed a model that was able to give a constant input voltage to the servo motor which changed its position at a constant speed. In this part, we will develop a model that is able to measure the speed of the servo motor.

Note: *We have provided detailed instructions on how to build Qube-Servo 3 based Simulink models in Parts 1 and 2. We now expect students to be able to develop Simulink models on their own given the description or diagram of the target model. Thus, from here on, we will only provide concise instructions on how to develop a required model.*

1. For this model, you can either use the Simulink Model developed in Part 2 as a starting point or start with the the Simulink Model that we have provided on Avenue (EE3CL4_Lab.slx). Remember to create separate folders on your computer for each part and save each model separately.

2. You may remember from earlier classes (or even from high school Physics) that speed is the rate of change of position, i.e., it is the derivative of position. Similarly, angular speed can be obtained by taking the derivative of angular position. That is, if angular position is given by $\theta(t)$, then angular speed $\omega(t)$ is given by:

$$\omega(t) = \frac{d\theta(t)}{dt} \tag{4}$$

The encoder gives us the angular position in terms of counts, which we converted to degrees in Parts 1 and 2. By taking the derivative of this signal, we can obtain the motor's angular speed.

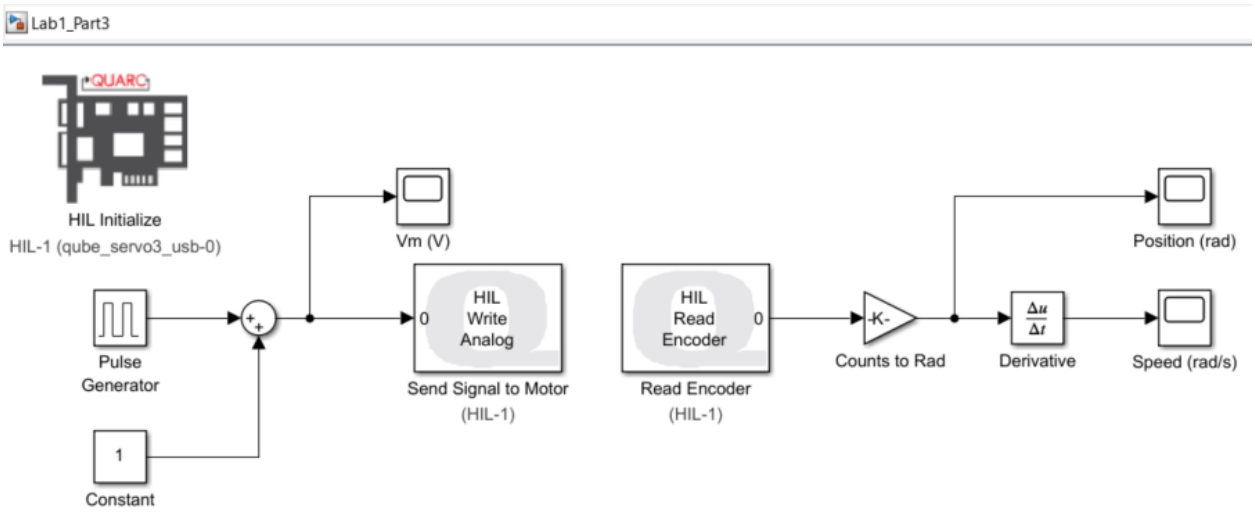


Figure 11: Target Simulink Model for Part 3.

3. In this part, our goal is to develop the target model shown in Figure 11. The next two steps explain what you are required to do.

4. Generate an input signal that consists of a Square Wave going between a low value of 1 V and a high value of 4 V, and that has a frequency of 0.4 Hz. Give this input signal to the motor. To do so, you will need the following blocks in your modeling space:

   (i) The *Pulse Generator* block found in the library: Simulink > Sources. Set the *Amplitude* of the *Pulse Generator* to 3, its *Period* to 2.5 sec, and its *Pulse Width* to 50%.

   (ii) The *Constant* block found in the library: Simulink > Commonly Used Blocks. Set the *Constant value* to 1.

   (iii) The *Sum* block found in the library: Simulink > Commonly Used Blocks. Set the *List of signs* as | ++.

   (iv) Connect the above three block together and then to the *HIL Write Analog* block as shown in Figure 11. This will send the required square wave to the motor.

   (v) Connect a *Scope* to the output of the *Sum* block to see the voltage signal being sent to the motor. By now you should know how to add the scope.

   (vi) Note: In the Q & A session for this part, your TA may ask you how this setup is creating the required square wave mentioned in Point 4 above. Discuss this within your group before calling the TA.

5. To obtain the angular speed at the output of the motor, you will need the following blocks in your modeling space:

(i) The *Gain* block found in the library: Simulink > Commonly Used Blocks. Remember, in Parts 1 and 2, we used this block to convert the output of the encoder (counts) to degrees. However, this time we will use this block to convert the encoder *counts* to *radians* as a unit of position. Discuss with each other as to what the *Gain* value of this block should be to convert counts to radians. Your TA will ask you about this.

(ii) The *Derivative* block found in the library: Simulink > Continuous. The output of this block will be the angular speed in radians/second.

(iii) Connect the above two blocks together and with the *HIL Read Encoder* block as shown in Figure 11. Also connect the two scopes at the output side as shown in Figure 11. These scopes will let us view the angular position and angular speed.

6. We will run the motor only for 10 second intervals in this part. To do this, go to the *Modeling* tab in the Simulink window and then click on *Model Settings*. In the *Solver* tab and in the *Simulation time* area, set the *Stop time* to be 10 seconds. Click OK.

7. By now, your model should look like the one shown in Figure 11. Save your work. Turn ON the Qube-Servo 3 motor, *build* your Simulink model and then press the *Monitor & Tune* button to deploy it on to the Qube-Servo 3.

8. Open all three scopes and make the following observations:

(i) Verify that the motor voltage is a square wave that goes between 1 V and 4 V, has a 50% duty cycle and a frequency of 0.4 Hz.

(ii) When you observe the shape of the angular position plot, you will see that it is piece-wise linear where lines of two distinct slopes come one after the other in a repeating pattern. Why is this so? The voltage in is a square wave

(iii) With the above discussion in mind, observe the shape of the angular speed plot. Are you able to explain its shape? Two distinct velocities, with sloping transitions between them

(iv) **Hint for Points 8(ii) and 8(iii) above**: *You are giving a square wave as an input to the motor. This square wave has two distinct voltage values (1 V and 4 V). Remember, increasing the voltage to a motor will make it change its position more quickly, i.e., increase the rate of change of position, and vice versa.*

(v) Again look at the angular speed plot. You will notice that is is quite noisy with large spikes taking place regularly. You should be able to explain why this is so? Step transitions have a high d/dx

(vi) **Hint for Point 8(v) above**: *Open the encoder position plot and zoom in on the position response. You will see (as you saw in Part 1) that the position measurement is not continuous but that it is composed of small steps. This is due to the nature of the encoder setup. Thus when a step change occurs, we encounter a large rate of change in the position measurement curve. This position measurement signal then enters the Derivative block. Recall that the derivative*

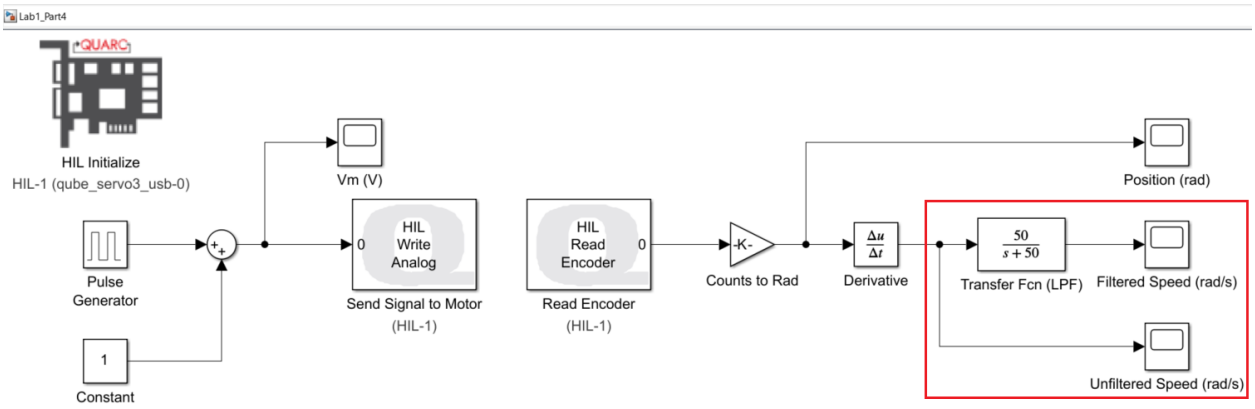


Figure 12: Target Simulink Model for Part 4.

itself is giving you information about the rate of change. What will the value of the derivative (angular speed in this case) be at the point of a step change in position? Large or small? Does this explain the noise in the angular speed plot?

9. At this point, you should call the Teaching Assistant to have your Part 3 evaluated.

10. Turn OFF the Qube-Servo 3 motor.

4.4 Part 4: Filtering the Noisy Speed Signal

Goal: In Part 3, we developed a model that measured the angular speed of the motor. However, we found the speed signal to be quite noisy. In this part, we will use a low-pass filter to block out the high frequency components of the noisy speed signal. A first order low-pass filter transfer function has the form:

$$G(s) = \frac{\omega_f}{s + \omega_f} \tag{5}$$

where ω_f is the cut-off frequency of the filter in radians per seconds (rad/s). All higher frequency components of the signal will be attenuated by at least -3 dB.

1. For this part, continue working with the Simulink Model developed in Part 3. In fact, we just need to add a low-pass filer to the model of Part 3 to filter out the noise. This will require only the parts inside the red box shown in Figure 12 (which shows the target model to be developed in Part 4) to be added to the Part 3 model. The next two steps discuss adding these blocks.

2. Bring a *Transfer Fcn* (function) block into the modeling space and connect it to the output of the *Derivative* block as shown in Figure 12. This block can be found in the library: Simulink > Continuous. To make this transfer function a low-pass filter, we will use the transfer function of the type given in Equation (5) with a cut-off

frequency $\omega_f = 50$ rad/s. To do so, open the *Transfer Fcn* block and set the *Numerator coefficients* as [50] and the *Denominator coefficients* as [1 50]. Press OK.

3. Add two *scopes* to the model, one to see the unfiltered speed and the other to see the filtered speed as shown in Figure 12.

4. Again, we will run the motor only for 10 second intervals in this part. To do this, go to the *Modeling* tab in the Simulink window and then click on *Model Settings*. In the *Solver* tab and in the *Simulation time* area, set the *Stop time* to be 10 seconds. Click OK.

5. By now, your model should look like the one shown in Figure 12. Save your work. Turn ON the Qube-Servo 3 motor, *build* your Simulink model and then press the *Monitor & Tune* button to deploy it on to the Qube-Servo 3.

6. Open only the scopes showing the unfiltered and filtered speed, and make the following observations:

   i Does the filtered angular speed signal have less noise than the unfiltered one?

   ii What is the cut-off frequency of the low-pass filter?

   iii If you reduce the cut-off frequency of the low-pass filter to $\omega_f = 10$ rad/s, what impact does this have on the level of noise in the filtered signal?

7. At this point, you should call the Teaching Assistant to have your Part 4 evaluated.

8. Turn OFF the Qube-Servo 3 motor.