Rotary Motion Servo Plant: SRV02

Rotary Experiment #00: QUARC Integration

Using SRV02 with QUARC

Instructor Manual

Table of Contents

1. Introduction

This manual demonstrates how to design QUARC controllers for the Quanser SRV02 system. Using QUARC blocks, several Simulink model are designed to send a voltage to the SRV02, read the load gear angle using a potentiometer and an encoder, and measured the speed of the load shaft using the tachometer.

The following topics are covered in this laboratory:

- Designing a Simulink model that interfaces with hardware using the blocks from the QUARC library
- Building QUARC code from a Simulink diagram
- Running the code to implement controllers and other algorithms on the SRV02.
- How to use scopes in order to view data as well as save that data for off line analysis.

Regarding Gray Boxes:

Gray boxes present in the **instructor manual** are not intended for the students as they provide solutions to the pre-lab assignments and contain typical experimental results from the laboratory procedure.

2. Prerequisites

In order to successfully carry out this laboratory, the user should be familiar with the following:

- Data acquisition card (e.g. QPID), the power amplifier (e.g. VoltPAQ), and the main components of the SRV02 (e.g. actuator, sensors), as described in References [1], [4], and [5], respectively.
- Wiring and operating procedure of the SRV02 plant with the amplifier and DAQ device, as discussed in Reference [5].
- Simulink environment enough to know its basic operations.
- QUARC should be installed as dictated in Reference [3].

3. Overview of Files

Table 1 below lists and describes the various files supplied with the SRV02 QUARC Integration laboratory.

File Name	Description
00 – SRV02 QUARC Integration – Student.pdf	Manuals that explains how to use QUARC with the Quanser SRV02 plant.
00 – SRV02 QUARC Integration – Instructor.pdf	Same as student version except the shaded boxes show the solutions.
q_srv02_volt.mdl	Simulink file used with QUARC to send a voltage to the SRV02.
q_srv02_pot_raw.mdl	Simulink file used with QUARC to send a voltage to the SRV02 and display the potentiometer voltage measurement.
q_srv02_pot.mdl	Simulink file used with QUARC to send a voltage to the SRV02 and display the potentiometer load gear angular measurement.
q_srv02_tach.mdl	Simulink file used with QUARC to send a voltage to the SRV02 and read angular rate of the load gear using the tachometer (as well as read the angle from the potentiometer).
q_srv02_enc.mdl	Simulink file used with QUARC to send a voltage to the SRV02 and display the load gear angle using the encoder (also displays the load gear velocity with the tachometer and the load gear angle using the potentiometer).
q_srv02_pot_addl.mdl	Same as q_srv02_pot.mdl, except it includes a Digital Display to display the load gear angle and the scope is configured to save a variable to the Matlab workspace.
ex_plotting_measured_data.m	Matlab script used to plot the data in the saved variable theta_l from q_srv02_pot_addl.mdl. Note that q_srv02_pot_addl must be ran with QUARC before running this script in order to have a theta_l variable in the Matlab workspace.

Table 1: Files supplied with the SRV02 QUARC Integration experiment.

4. SRV02 and QUARC Integration

In Section 4.1, a Simulink model is designed using blocks from the QUARC library to send a voltage to the SRV02. This Simulink model is then modified to read from the potentiometer, as dictated in Sections 4.2. The model from Section 4.2 is further modified to include the angular rate of the load gear using the tachometer, discussed in Section 4.3, and the angular position of the load gear using an encoder measurement, described in Section 4.4.

4.1. Applying a Voltage to the DC Motor

Here are the basic steps to apply a voltage to the SRV02 motor using QUARC:

1. Design a Simulink model that interacts with the data-acquisition device using block from the

QUARC library. This explained in Section 4.1.1.
2. Build the code from the Simulink model, as dictated in Section 4.1.2.
3. As explained in Section 4.1.3, execute the code.

4.1.1. Designing the Simulink Model

In this section, it is shown how to design a Simulink Model using QUARC blocks to feed a sinusoid voltage to the SRV02, as shown in Figure 1. The blocks from the *QUARC Targets* library are used to interact with the data acquisition board, e.g. Quanser Q2-USB or QPID device.

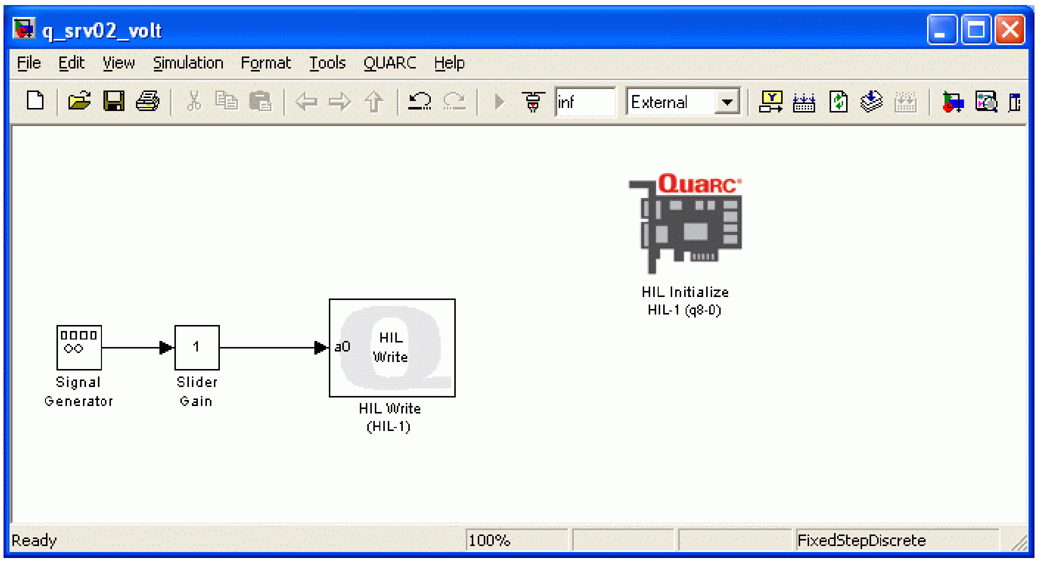

Figure 1: Sample Simulink Model used to feed sinusoid voltage to the SRV02.

Follow these steps to design the Simulink diagram:
1. Load the Matlab software.
2. Create a new Simulink Diagram by clicking on *File | New | Model* item from 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. As illustrated in Figure 2, expand the *QUARC Targets* item and go to the *Data Acquisition \ Generic \ Configuration* folder.
5. Click-and-drag the HIL Initialize block from the library window into the blank Simulink model. This is used to configure your data-acquisition device, e.g. the Quanser QPID or Q2-USB board.
6. In the Library Browser, go to the *Generic \ Immediate I/O* category. This contains various blocks used to interact with actuators and sensors.

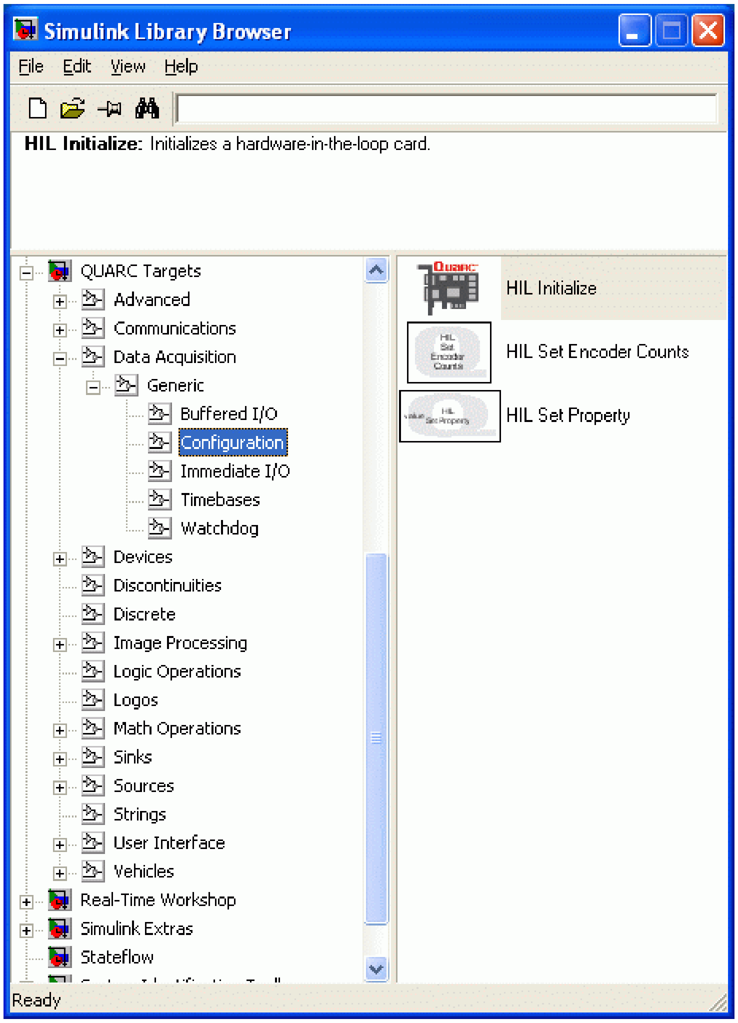

Figure 2: QUARC Configuration folder in the Simulink Library Browser window.

7. Click-and-drag the HIL Write Analog block from the library into the Simulink diagram. This block is used to output a voltage from a digital-to-analog channel, i.e. D/A, on the data-acquisition device.
8. Add a Signal Generator block (*Simulink* \ *Source* folder), a Slider Gain block (*Simulink* \ *Math Operations* category), and a Gain (*Simulink* \ *Math Operations* category) into the Simulink model. Connect the blocks as shown in Figure 1.

If using a VoltPAQ, make sure you set the gain on the VoltPAQ to 1.

9. Double-click on the HIL Initialize block.
10. In the *Board type* field, select the board that is installed in your PC, e.g. if using the Q2-USB then select *q2_usb*.
11. If more than one board is installed, ensure the *Board number* field is set correctly, e.g. if two boards are used then choose either 0 or 1.
12. The default settings for the board are fine for this experiment. For more information on this block, click on its *Help* button. Otherwise click on the OK button and proceed.

13. Double-click on the HIL Write Analog block.
14. Set *Board name* to *HIL-1*. By default, *Channels* is set to 0 and the *Sample time* field is set to -1. Recall that, as instructed in Reference [5], the DC motor is connected to Analog Output Channel #0 on the hardware-in-the-loop board. Therefore *Channels* should be set to 0. Also, setting the *Sample Time* to -1 implies that the sampling interval is inherited from the previous block. This setting is fine as well.
15. Click on the OK button to save and close the HIL Write Analog block properties.
16. Save the Simulink mode by selecting the *File | Save* item in the menu bar or clicking on the *Save* icon.

4.1.2. Compiling the Model

The Simulink model can now be used by QUARC to generate code which will later be executed to send a voltage to the SRV02.

Follow these steps to generate code from a Simulink diagram:

1. In the Simulink diagram designed in Section 4.1.1, 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).
2. To view the compiler options, go to *QUARC | Options* in the Simulink model tool bar. In the Real-Time Workshop pane, the *System target file* is set to Target Language Compiler file *quarc_windows.tlc* and, in the *Makefile configuration* section, the *Make command* is set to *make_rtw* and the *Template makefile* is set to *quarc_default_tmf* file.
3. Click on the the *Solver* item in the left-hand pane.
4. In the *Simulation time* section, the *Stop time* is set to *inf* in order for the code to be executed continuously until it is stopped manually by the user. Alternatively, the *Stop time* parameter can be set to the desired duration (code will cease executing when the stop time value is reached).
5. In the *Solver options* section, the *Type* parameter is set to *Fixed-step* and the *Solver* is set to *discrete*. There are no continuous blocks inside the designed Simulink model, therefore having a discrete solve is fine. However, if an Integrator block or another continuous system is used then the *Solver* field would have to changed to an integration method such as *ode1 (Euler)*.
6. The *Fixed-step size* field sets the sampling interval, or sampling time, of the controller. By default this is set to 0.001 seconds, which is a sampling rate of 1 kHz.
7. Click on the OK button to close the Configuration Parameters window.
8. Select the *QUARC | Build* item. Various lines in the Matlab Command Window should be displayed as the model is being compiled.
9. Once done compiling, a QUARC Windows executable file along with a folder containing various C and Matlab files are generated. Note that once the executable is created, the folder is no longer needed.
10. Note that the executable and associated code folder may be removed from the current directory by clicking on *QUARC | Clean* item.

4.1.3. Running QUARC Code

Once the Simulink Model has been compiled, the code can be executed and the voltage set in the Simulink model can be sent to the SRV02 motor. Go through these steps:

1. Turn ON the power of the the power amplifier.
2. To begin executing the code, click on the *QUARC* | *Start* item in the Simulink model. The SRV02 external gears on the SRV02 should begin rotating back-and-forth. This command actually does two things: it connects to the target and then executes the code. Alternatively, users may decide to go through these steps individually. To do this first click on the *Connect to target* icon, or select the *Simulation* | *Connect to target* item from the menu bar, and then click on the *Run* icon, or from the menu choose the *Simulation* | *Start Real-Time Code* item, to execute the code.
3. Double-click on the Signal Generator block to open its parameter window.
4. Set the *Frequency* field to 0.5 Hz and click on the OK button. Remark how the parameter change effects the SRV02 immediately: the gears on the SRV02 begin to switch back-and-forth more slowly.
5. Vary the value of the Slider Gain block between 0 and 2. Examine how the angular rate of the SRV02 gears change proportionally with the amplitude of the sine wave.
6. Select the *QUARC* | *Stop* item to stop the code from running (or click on the *Stop* button in the Simulink model tool bar).
7. Power OFF the amplifier if no more experiments will be ran in this session.

4.2. Measuring SRV02 Potentiometer

In this section, the Simulink Model previously designed in Section 4.1.1 is modified to obtain readings from the SRV02 potentiometer sensor.

4.2.1. Reading Potentiometer Voltage

Design a Simulink model, similarly as pictured in Figure 3, below, that reads the potentiometer voltage.

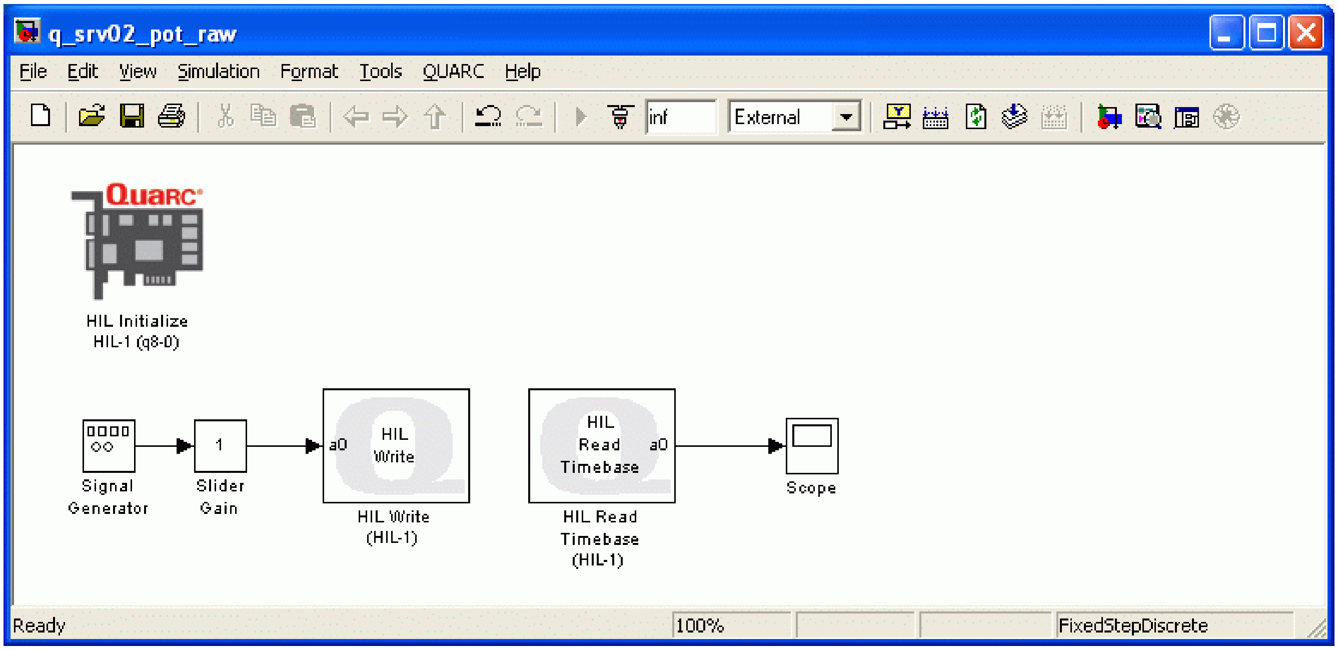

Figure 3: Simulink model used to send voltage to the SRV02 and obtain readings from the potentiometer sensor.

Follow this procedure to design the Simulink diagram:

1. From the *Generic \ Timebases* category in the Simulink Library Browser, click-and-drag the *HIL Read Timebase* block into the Simulink diagram created previously in Section 4.1.1. This block can be configured to read from analog, encoder, and digital channels. In this case it will be used to read a voltage from an analog-to-digital channel, i.e. A/D, on the data-acquisition device. Using a *Timebase* type block causes the running model to be triggered by the hardware hardware timer on the data-acquisition board as opposed to the system clock. This increases performance of the controller by reducing jitter and allowing for greater sampling rates. The system clock would instead be used if, for instance, the HIL Read block from the *Immediate I/O* category was selected.
2. Double-click on the HIL Read Timebase block to open its properties.
3. Set *Board name* to *HIL-1*.
4. Recall that, as instructed in Reference [5], the potentiometer is connected to Analog Input #0 on the hardware-in-the-loop board. Therefore the default setting of *Analog channels* set to 0 is fine.
5. Add a Scope block from the *Simulink \ Sinks* folder in the Library Browser and connect it to the output of the HIL Read Analog block, similarly as shown in Figure 3, above,
6. Set the *Frequency* parameter in the Signal Generator block to 1.0 Hz and the Slider Gain block to 1.
7. Double-click on the Scope icon.
8. Save the Simulink model (you may want to save the model as a different file).
9. Power ON the amplifier.
10. Go to *QUARC | Build* to compile the code.
11. Click on *QUARC | Start* to execute the code. As the SRV02 rotates back-and-forth, the Scope should display the potentiometer readings similarly as shown in Figure 4. Remark that the readings shown are the voltage output of the potentiometer and not an angular measurement in degrees or radians.

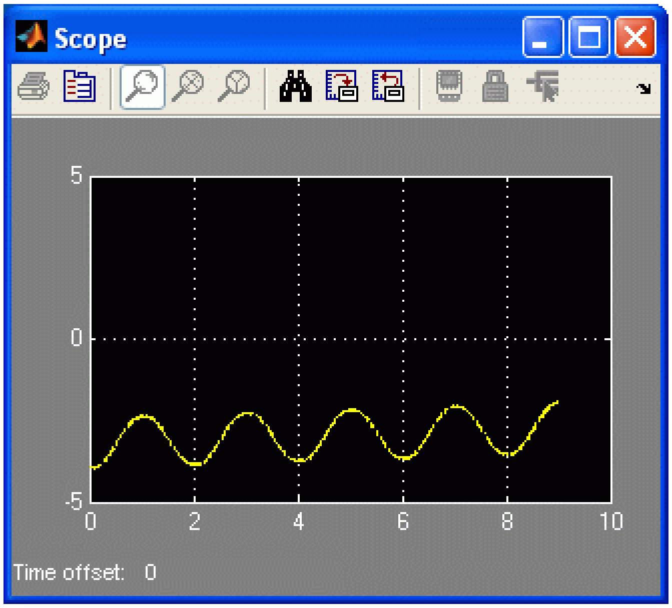

Figure 4: Reading raw voltage from the SRV02 potentiometer.

NOTE: If the scope is not displaying any data and the SRV02 is rotating, then see Section 4.5.3 to re-configure the RTW Signal & Triggering properties.

12. Vary the sine wave frequency between 0.1 and 2 Hz and the Slider Gain value between 0 and 2. You may notice the discontinuity present in the potentiometer, as illustrated in Figure 5.

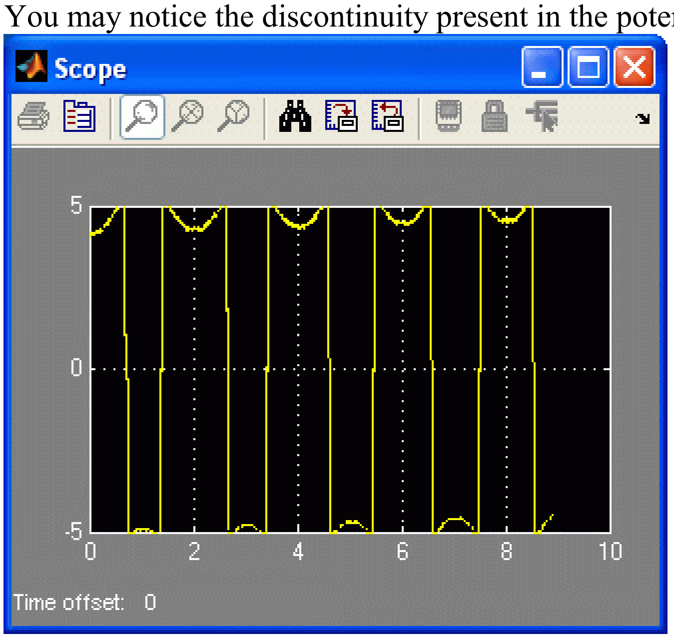

Figure 5: Potentiometer reading when encountering its discontinuity.

13. As discussed in Reference [5], the potentiometer is an absolute sensor. To demonstrate this, set the Slider Gain to 0 V.

14. Manually rotate the gears such that the potentiometer is reading a value other than 0 V.
15. Stop the code by clicking on QUARC | Stop.
16. Start the code again, i.e. select QUARC | Start and examine how the voltage reading is the same as previously set.
17. Stop the code again.

4.2.2. Adding Calibration and Scopes

The Simulink model will now be modified to read the load angle from the potentiometer and the voltage commanded to the motor, similarly as shown in Figure 6.

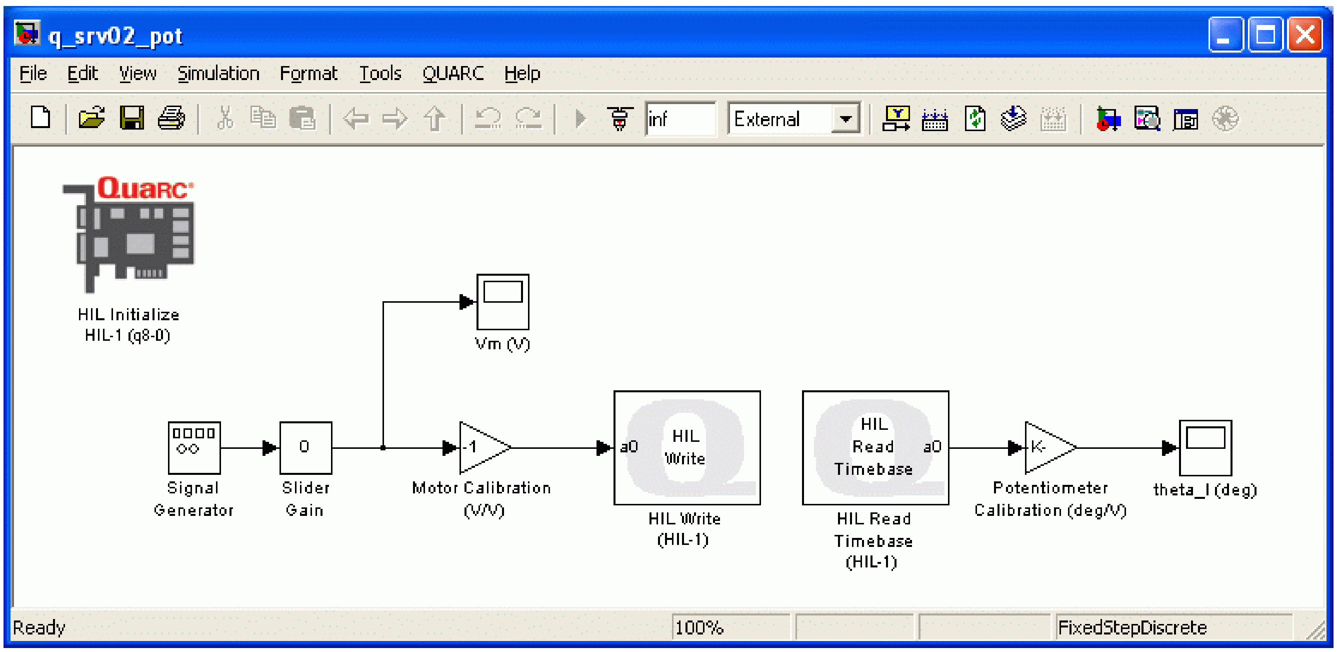

Figure 6: Simulink model used to send voltage to the SRV02 and obtain angular readings from the potentiometer.

Continue to modify the Simulink diagram built in the previous section with these steps:

1. Label the Scope block that is presently connected to the HIL Read Analog block to *theta_l (deg)*. This will display the angular measurement of the load gear in degrees.
2. Add a Scope block from the *Simulink \ Sinks* folder in the Library Browser and connect it to the output of the Slider Gain block, as shown in Figure 6, above. Label the Scope block *Vm (V)* which stands for the input motor voltage.
3. From the *Simulink \ Math Operations* category in the Library Browser, add two Gain blocks into the Simulink model.
4. Connect the Gain blocks as illustrated in Figure 6. Place one Gain block between the *Amplifier Pre-Compensation* Gain and the HIL Write Analog blocks and label it *Motor Calibration (V/V)*. Add the second Gain block between the HIL Read Timebase and *theta_l* Scope blocks and denote it *Potentiometer Calibration (deg/V)*.
5. Re-compile the Simulink model, i.e. click on *QUARC | Build*.
6. Click on *QUARC | Start* to run the code.

7. Open the *Vm (V)* and *theta_l (deg)* scopes.
8. In the Signal Generator, set the *Frequency* parameter to 0.25 Hz.
9. Set the Slider Gain block to 1. The input voltage signal in the *Vm (V)* scope should be as shown in Figure 7.

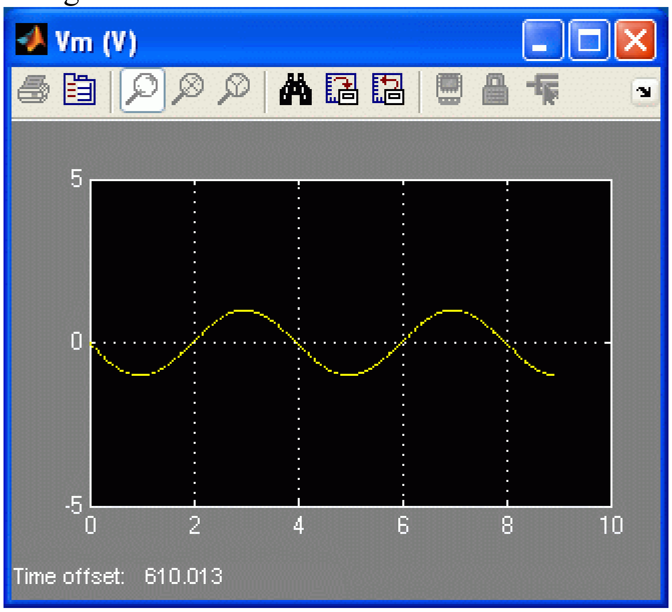

Figure 7: Sinusoidal SRV02 input voltage.

10. Currently when the voltage goes positive the load gear rotates in the clockwise direction. However, the desired convention is for the load gear to rotate in the counter-clockwise direction when the voltage goes positive. Thus set the *Motor Calibration (V/V)* block to -1.
11. As described in Reference [5], the potentiometer outputs between +/- 5V when it rotated 352 degrees. Enter the value 352 / 10 in the *Potentiometer Calibration (deg/V)* block.
12. In the *theta_l (deg)* scope, click on the *Autoscale* icon so the y-axis range is increased and the full signal can be viewed.
13. Set the Slider Gain block to 0.
14. Rotate the load gear manually and examine the corresponding response in the *theta_l (deg)* scope. Confirm that, indeed, the correct measurement is being taken.
15. Position the load gear such that 0 is read in the *theta_l (deg)* scope .
16. Set the Slider Gain block to 1.
17. Examine the relationship between the input voltage and load position. When the input voltage increases in the positive direction, the potentiometer angle decreases. Add a negative sign to the *Potentiometer Calibration (deg/V)* block so the entered value becomes -352/10. Now, the angular position of the load gear, displayed in the *theta_l (deg)* scope, reads increasingly positive when the commanded motor input voltage, i.e. signal in the *Vm (V)* scope, is positive.

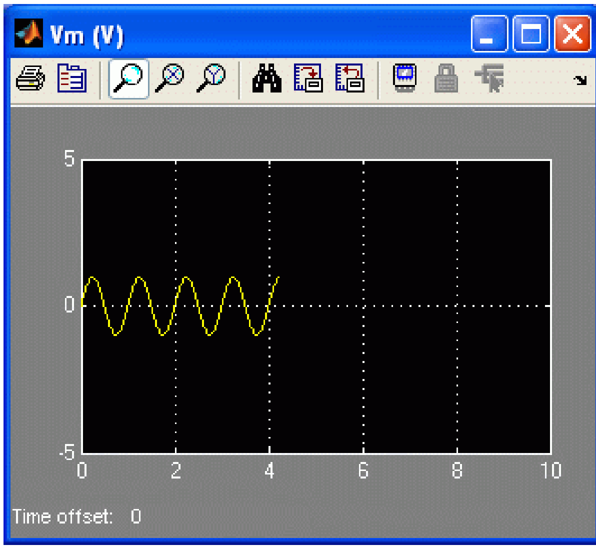

Figure 8: Input voltage scope.

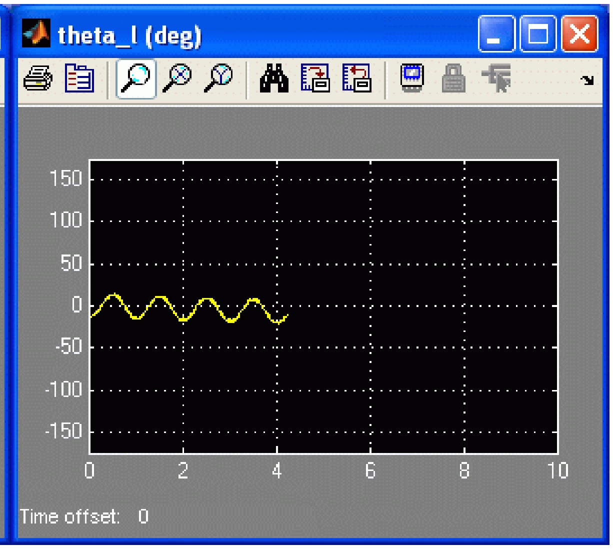

Figure 9: Load gear position response when calibrated.

18. Select the *QUARC* | *Stop* item to stop the code from running.
19. Power OFF the amplifier if no more experiments will be ran in this session.

4.3. Measuring SRV02 Tachometer

In this section, the Simulink diagram designed in Section 4.2 is modified to include the readings from the tachometer. Before continuing, please ensure that the SRV02 unit being used has a tachometer, i.e. SRV02-T option.

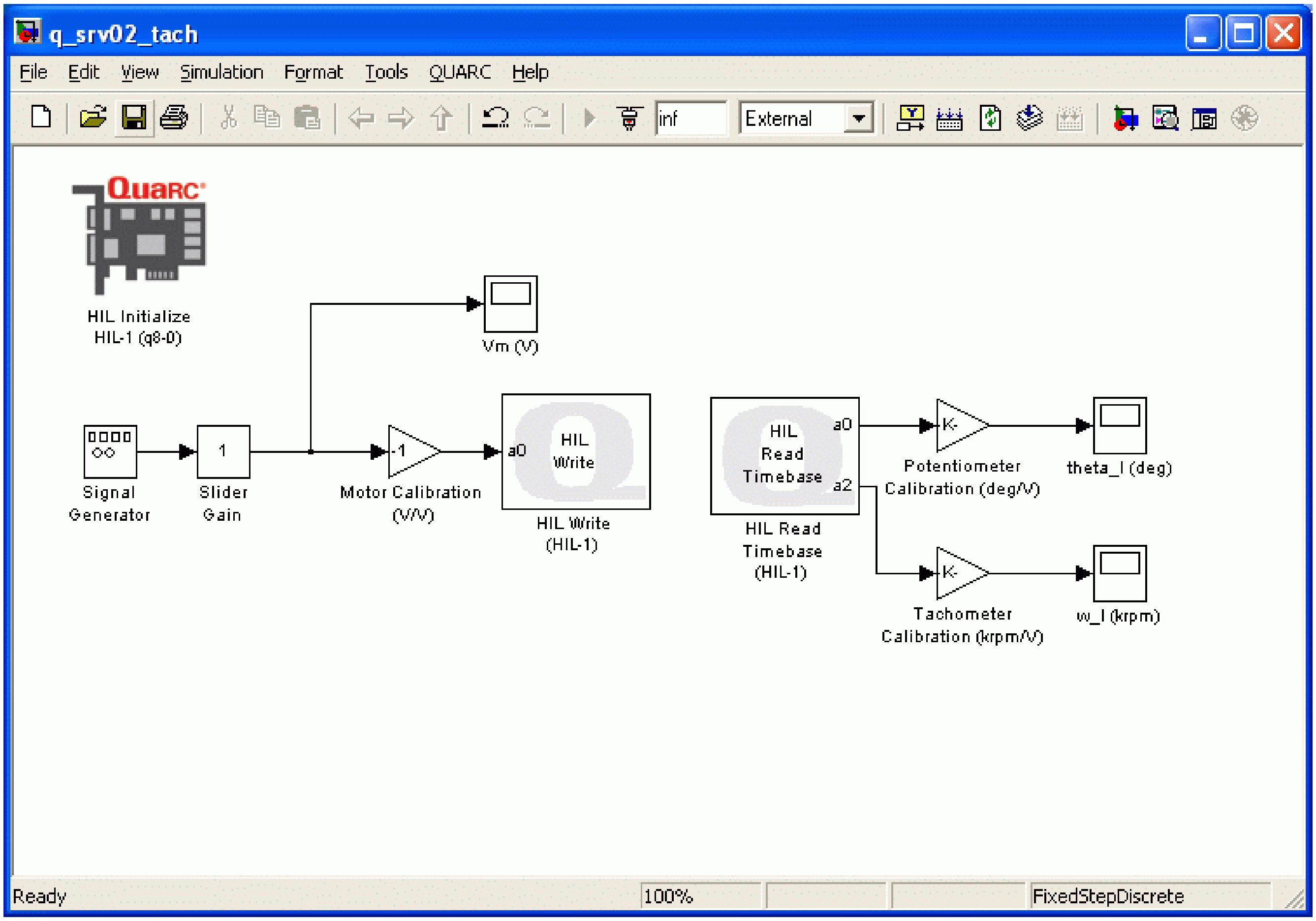

Figure 10: Simulink model used to send voltage to SRV02 and read its potentiometer and tachometer sensors.

Using the Simulink diagram built in the previous section, go through this procedure to add the tachometer functionality:

1. Double-click on the HIL Read Timebase block to open its properties.
2. As detailed in Reference [5], the tachometer is connected to Analog Input #2 on the hardware-in-the-loop board. To add a channel, set the *Analog channels* field to [0,2] and click on the OK button.
3. Add a Scope block and a Gain block from Library Browser into the Simulink model.
4. Connect the Scope and Gain blocks as depicted in Figure 10, above. That is, connect the Gain block to the output of Channel #2 from the HIL Read Timebase block and label it *Tachometer Calibration (rpm/V)*. Then, connect the output of this gain block to the input of the Scope block and label the scope *w_l (rpm)*. This scope will display the measured speed of the load gear in thousand revolutions per minute.
5. Set the Signal Generator Frequency parameter to 1.0 Hz and the Slider Gain block to 1.
6. Open the *Vm (V)* and the *w_l (rpm)* scopes.
7. Power ON the amplifier.
8. Go to *QUARC* | *Build* to compile the code.
9. Click on *QUARC* | *Start* to execute the code. As the SRV02 rotates back-and-forth, the *w_l (rpm/V)* scope should display the tachometer readings. Since the *Tachometer Calibration*

(krpm/V) gain has not been configured yet, the scope is displaying the tachometer output voltage, which is proportional to the speed of the load shaft.

10. The back-emf constant of the tachometer sensor is 1.5 mV/rpm. However, the measurement is taken directly from the motor itself (see Reference [5]). Thus to read the velocity of the gear the tachometer calibration gain must be divided by the gear ratio. Enter 1 / 1.5 / 70 when using the SRV02 in the high-gear configuration or - 1 / 1.5 / 14 when using the SRV02 in the low-gear configuration in the *Tachometer Calibration (krpm/V)* gain block.
11. The measurement will be very small. Click on the *Autoscale* icon in the scope to zoom up on the signal. Alternatively, the y-range of the scope can be set manually. To do this, right-click on the y-axis, select *Axes Properties* from the drop-down menu, and set the desired y-range values.
12. Examine the relationship between the input voltage and load speed. When the input voltage increases in the positive direction, the tachometer velocity decreases. As with the potentiometer when reading position, the speed of the load shaft should go positive when the input voltage is positive. Thus add a negative sign to the *Tachometer Calibration (rpm/V)* block so the entered value becomes -1 / 1.5 / 70 for high-gear or 1 / 1.5 / 14 for low-gear . The input voltage and load velocity scopes should read as shown in Figure 11 and Figure 12.

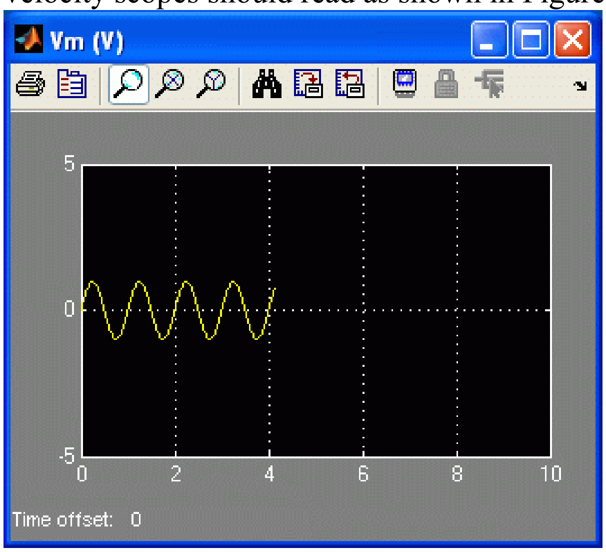

Figure 11: Input voltage scope.

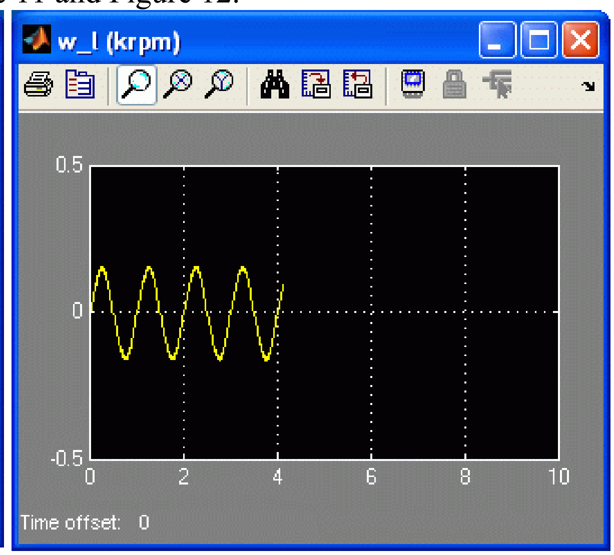

Figure 12: Load shaft velocity measured by tachometer.

Note: If you want the measurement to be in RPM instead of kRPM, enter - 1000 / 1.5 / 70 for high-gear or - 1000 / 1.5 / 14 for low-gear.

13. Select the *QUARC | Stop* item to stop the code from running.
14. Power OFF the amplifier if no more experiments will be ran in this session.

4.4. Measuring SRV02 Encoder

The Simulink diagram designed previously is modified to include an encoder measurement, as illustrated in Figure 13 below. Before continuing, please ensure that the SRV02 unit being used has an encoder, i.e. SRV02-E option.

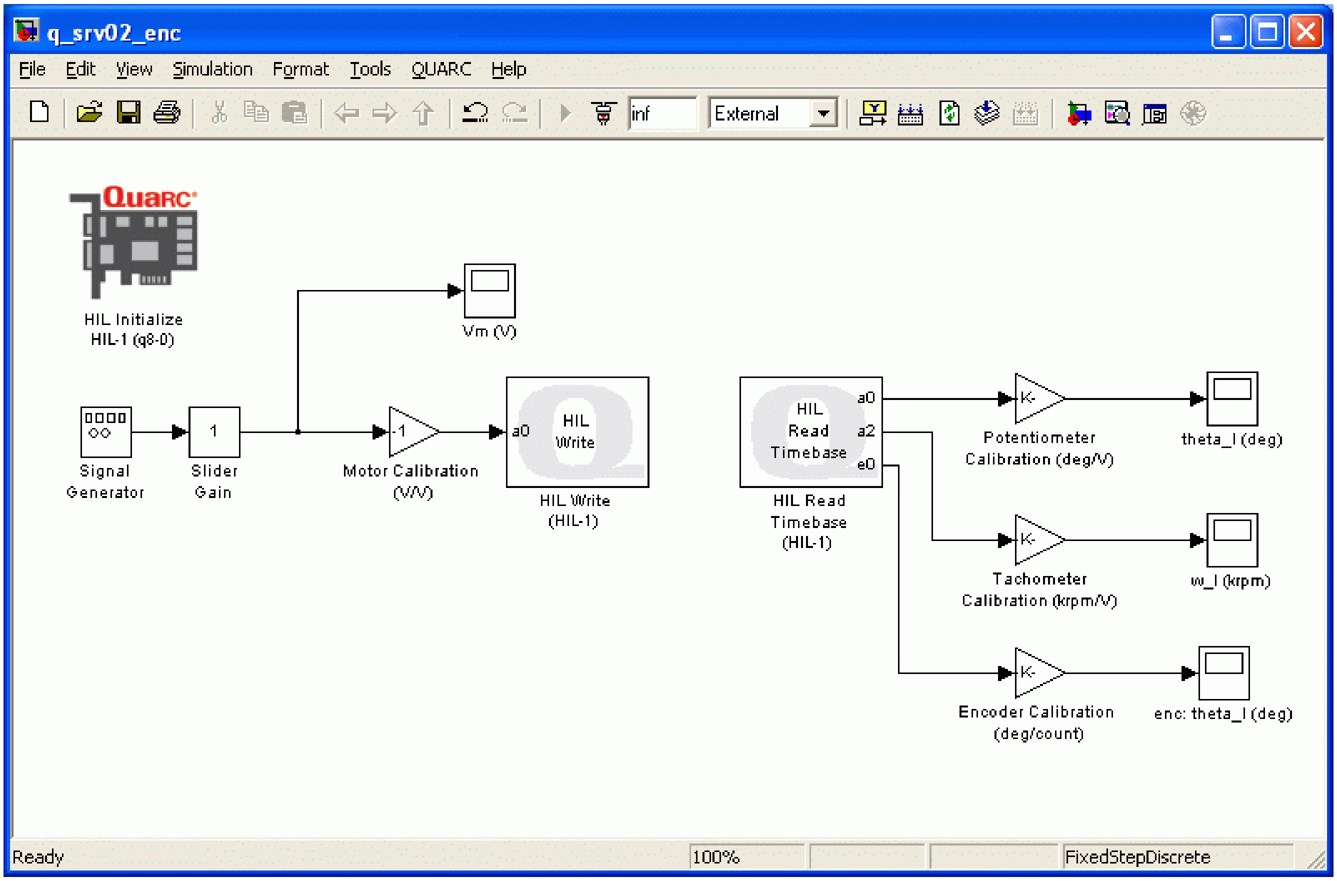

Figure 13: Simulink model used to send voltage to the SRV02 and read its potentiometer, tachometer, and encoder sensors.

Using the Simulink model designed in either section 4.2 or 4.3, follow this procedure to add encoder functionality:

1. Double-click on the HIL Read Timebase block to open its properties.
2. Recall that, as instructed in Reference [5], the encoder is connected to Encoder Input #0 on the data acquisition board. Thus set the *Encoder channels* field to [0]. The default encoder configurations in the HIL Initialize block are fine but keep in mind that these can be changed.
3. Add a Gain block and a Scope block from the *Math Operations* and *Sinks* folders in the Library Browser, respectively, into the Simulink model.
4. Connect the Scope and Gain blocks as depicted in Figure 13, above. Thus connect the Gain block to the output labeled *e0* of the HIL Read Timebase block and label it *Encoder Calibration (deg/count)*. Then, connect the output of this gain block to the input of the Scope block and label the scope *enc: theta_l (deg)*. This scope will display the measured angular position of the load gear in degrees.
5. Set the *Frequency* parameter in the Signal Generator block to 1.0 Hz and the Slider Gain block to 1.
6. Open the *Vm (V)* and *enc: theta_l (deg)* scopes.
7. Save the Simulink model (you may want to save the model as a different file).
8. Power ON the amplifier.
9. Go to *QUARC | Build* to compile the code.

10. Click on *QUARC | Start* to execute the code. As the SRV02 rotates back-and-forth, the *enc: theta_l (deg)* should display the encoder readings. Since the *Encoder Calibration (deg/count)* gain has not been configured yet, the scope is displaying the number of counts from the encoder output, which is proportional to the position of the load shaft.
11. The measurement will be very large. Click on the *Autoscale* icon in the scope to zoom out and view the entire signal. Alternatively, the y-range of the scope can be set manually. To do this, right-click on the y-axis, select *Axes Properties* from the drop-down menu, and set the desired y-range values
12. As discussed in Reference [5], the encoder outputs 4096 counts for every full revolution. To measure the load gear angle, set the *Encoder Calibration (deg/count)* gain block to 360 / 4096 degrees per count.
13. The measurement will be very small. Click on the *Autoscale* icon in the scope to zoom up on the signal or adjust the range of the y-axis manually. The input voltage and position scopes should appear similarly as shown in Figure 14 and Figure 15. Remark that no further calibration is needed: the encoder position increases when the input voltage goes positive.

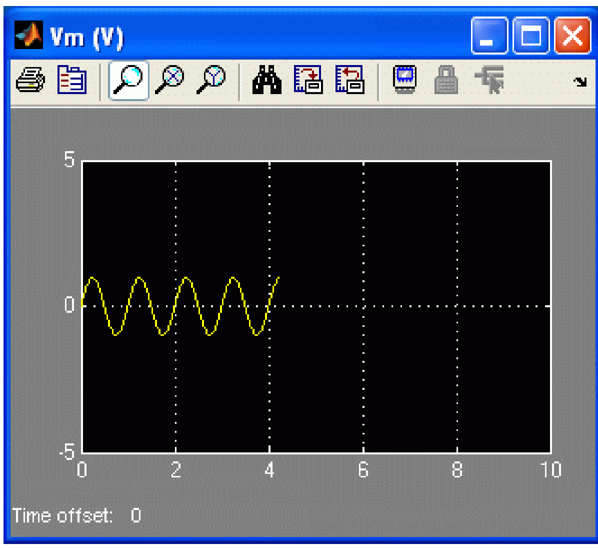

Figure 14: Input sinusoidal voltage.

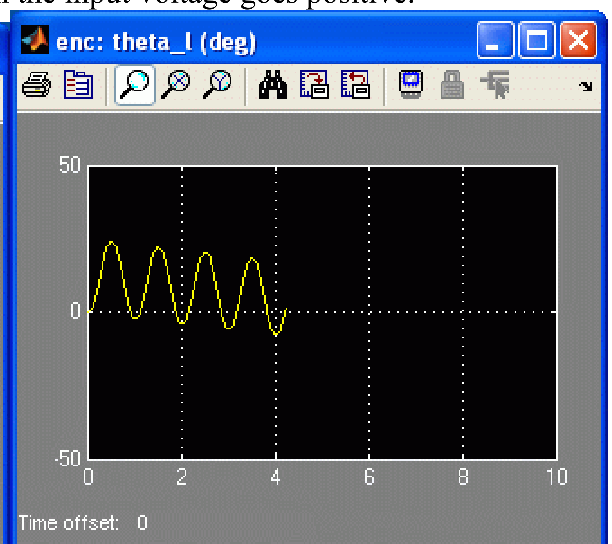

Figure 15: Load gear angle measured with encoder.

14. Set the Slider Gain block to 0.
15. Move around the load gear and examine the response in the *enc: theta_l (deg)* scope. Rotate the gear so the measurement is about 120 degrees.
16. Select the *QUARC | Stop* item to stop the code from running.
17. Begin running the controller again. Notice in the *theta_l (deg)* scope that the encoder is now reading 0 degrees. This demonstrates the difference between a relative position measurement, e.g. incremental encoder, and an absolute measurement, e.g. potentiometer.
18. Stop running the controller.
19. Power OFF the amplifier if no more experiments will be ran in this session.

4.5. Other QUARC Functions

4.5.1. Digital Display

It has been shown how data from the output of a block can be displayed in a Scope. However, this data can also be displayed using a Digital Display. It is sometimes beneficial to view the numerical value of the block at that time instant. In Figure 16 for example, the measured angle of the load gear is displayed.

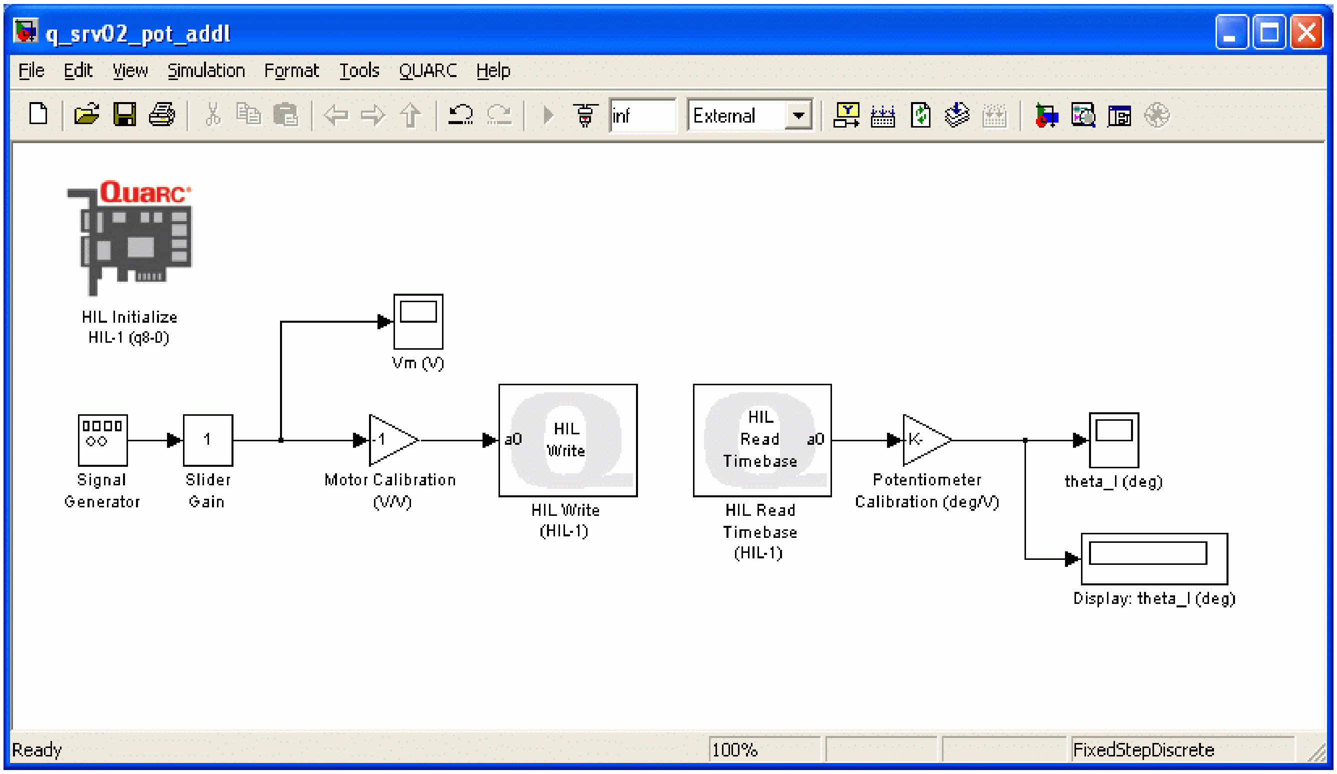

Figure 16: Simulink model that sends a voltage to the SRV02 and displays the potentiometer using both a scope and a digital display.

4.5.2. Saving Scope Data

The scopes in the Simulink model can be configured to save variables in the Matlab workspace. For instance, to configure the load gear position scope *theta_l (deg)* in the Simulink model from Section 4.2 perform the following:

1. Open the *theta_l (deg)* scope (double-click on it).
2. Click on *Parameters* icon.
3. In the *Parameters* window, select the *Data History* tab.
4. Select the *Save data to workspace* check box.
5. Change the *Variable* name field to a desired variable, e.g. *theta_l*.
6. Set *Format* to *Array*.
7. By default, the *Limit data points to last* box is set to 5000. This means that only the last 5000

points of data will be saved in the variable. Thus, given that the controller runs at 1 kHz, this implies that the last 5 seconds of data shown in the *theta_l (deg)* scope will be captured.

8. Click on the *OK* button.
9. Save the Simulink model.
10. Select *QUARC | Build* to rebuild the model.
11. Click on *QUARC | Run*.
12. After the controller has ran for at least 5 seconds, click on *QUARC | Stop* to cease running the code.
13. When the controller is stopped, the variable is saved to the workspace. The variable *theta_l* is a two-dimensional array with 5000 elements. The first vector is the running time and the second vector is the position of the load gear. To plot these into a Matlab figure, see the script shown in and the resulting scope that is generated after the script is ran.

```
%% Example of Loading Response Data
%
%% Load sample data from the Matlab workspace into variables
% Time vector (s)
t = theta_l(:,1);
% Load gear position (deg)
th_l = theta_l(:,2);
%
%% Plot response
plot(t,th_l,'r-');
label('\theta_l (deg)')
xlabel('time (s)');
title('\bf SRV02 Open-loop Position Response');
```

Text 1: Sample script used to plot saved variable theta_l in a Matlab figure.

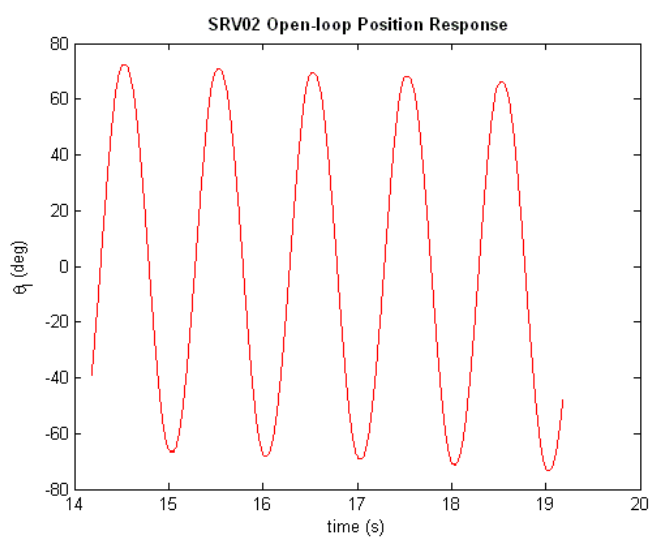

Figure 17: Matlab figure showing measured response from the saved theta_l variable.

4.5.3. Signal and Triggering

If a scope is not displaying any traces when running QUARC in External mode or you wish to adjust the amount of data that can be displayed, then the Signal & Triggering properties need to be configured.

Follow this procedure these settings:

1. In the Simulink model, select the *Tools* | *External Model Control Panel* item.
2. Click on the *Signal & Triggering* button and the window shown in Figure 18 will load.

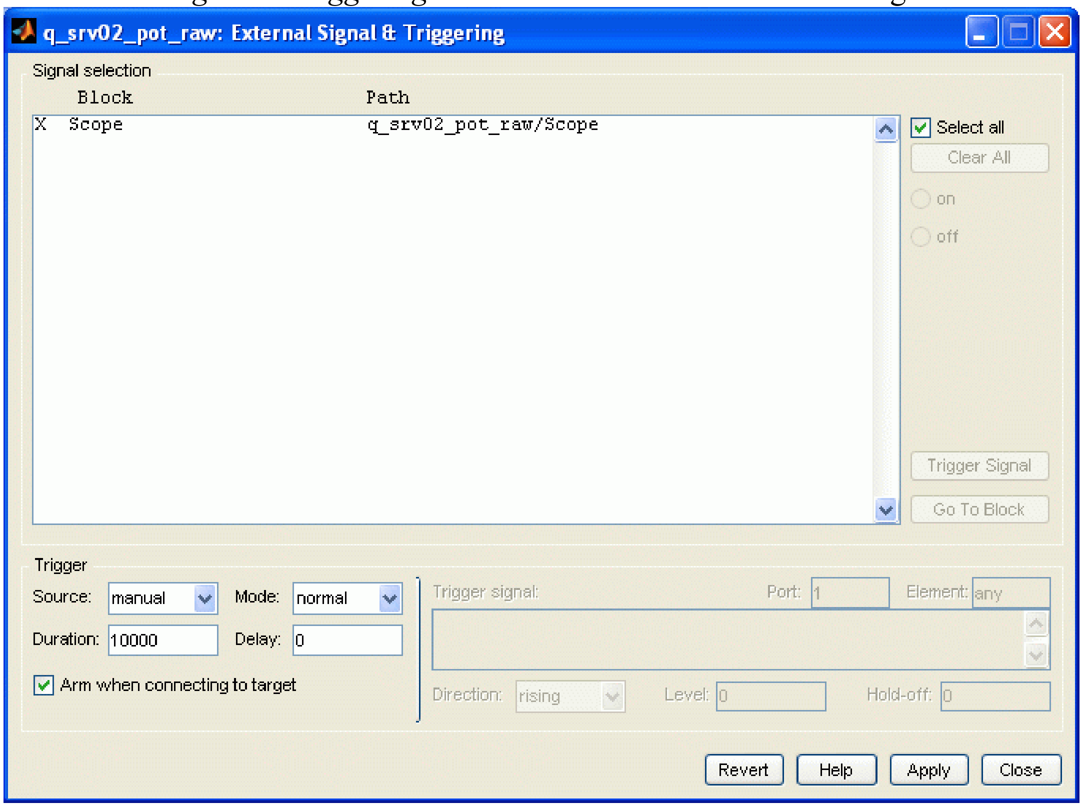

Figure 18: Signal and Triggering window for a Simulink diagram with a single Scope.

3. As shown in Figure 18, ensure the *Arm when connection to target* option is selected so that the scope plots information when the QUARC controller is running.
4. The *Duration* specifies the amount of points that are plotted. If the buffer is set to 10,000 points and the controller runs at a sampling rate of 1 kHz, i.e. sampling interval of 0.001 s, then the scope will plot up to 10 seconds of data. Thus to view up to 20 seconds the *Duration* would have to be changed to 20 seconds.
5. Make sure you click on the *Apply* button to commit the changes.
6. Click on *Close* when done.

5. References

[1] DAQ User Manual
[2] QUARC User Manual (type `doc quarc` in Matlab to access).
[3] QUARC Installation Manual.
[4] Amplifier User Manual.
[5] SRV02 User Manual.