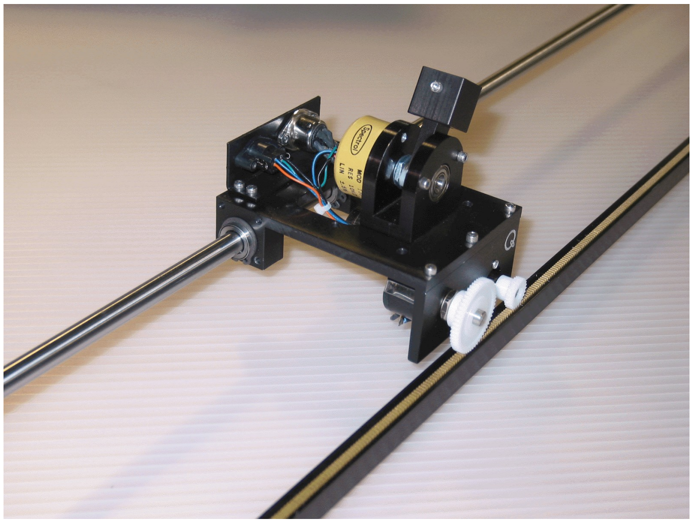
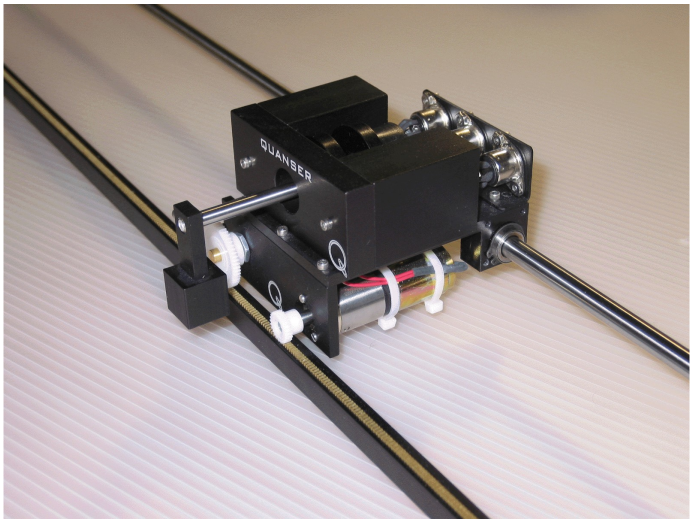
Linear Motion Servo Plants: IP01 or IP02

Linear Experiment #0: Integration with QUARC

IP01 and IP02





Student Handout

Table of Contents

1. Objectives

In this introductory session you will become familiar with QUARC and the IP01 or IP02 servo experiment. At the end of the session you will know the following:

- The main components comprising this basic setup (e.g. data acquisition board, power amplifier, real-time control software).
- How to wire up the system.
- How to create a Simulink based controller and run it in real-time using QUARC (i.e. Quanser's Windows-based real-time control software).
- How to measure and display, using a Simulink diagram, the encoder and/or potentiometer signals.
- How to apply and display, using a Simulink diagram, a varying voltage to the IP01 or IP02 DC motor.
- How to construct a Simulink block (i.e. subsystem) that represents the IP01 or IP02 servo system.

2. Prerequisites

To successfully carry out this introductory laboratory, the prerequisites are:

i) To be familiar with your IP01 or IP02 main components, (e.g. actuator, sensors), your power amplifier (e.g. VoltPAQ), and your data acquisition card (e.g. Q4 or Q8), as described in References [1], [2], and [3].
ii) To be familiar with the complete wiring of your IP01 or IP02 servo plant, as per dictated in Reference [1].

3. References

[1] *IP01 and IP02 User Manual.*
[2] *DAQ User Manual*.
[3] *Universal Power Module User Manual.*
[4] *QUARC User Manual* (type `doc quarc` in Matlab to access).
[5] *QUARC Installation Manual*.

4. Experimental Setup

4.1. Main Components

To setup this preparatory laboratory, the following hardware and software are required:

- **Power Module:** Quanser VoltPAQ, or equivalent.
- **Data Acquistion Board:** Quanser Q2-USB, Q8-USB, QPID or equivalent.
- **Linear Motion Servo Plant:** Quanser IP01 or IP02, as shown in Figures 1 and 2, respectively.
- **Real Time Control Software:** The QUARC-Simulink configuration, as detailed in the Reference [5], or equivalent.

For a complete and detailed description of the main components comprising this setup, please refer to the manuals corresponding to your configuration.

4.2. Wiring

To wire up the system, please follow the default wiring procedure for your IP01 or IP02 as fully described in Reference [1]. When you are confident with your connections, you can power up the amplifier.

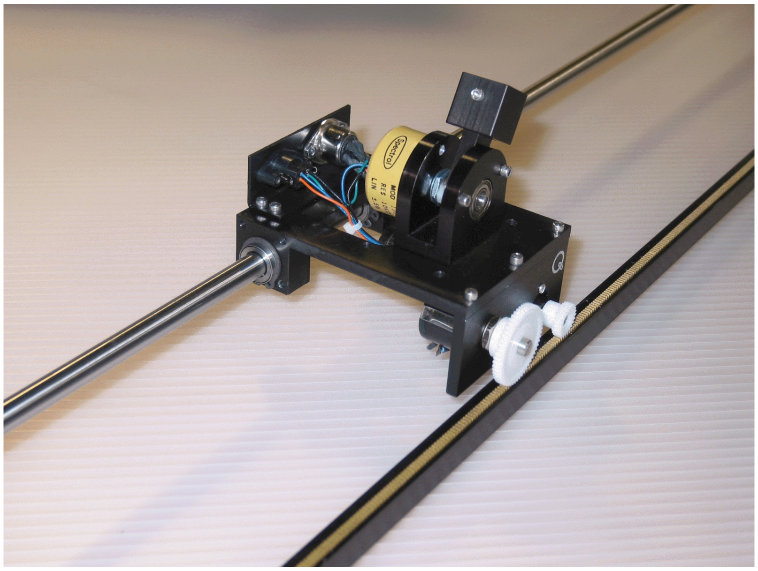


Figure 1 IP01 System

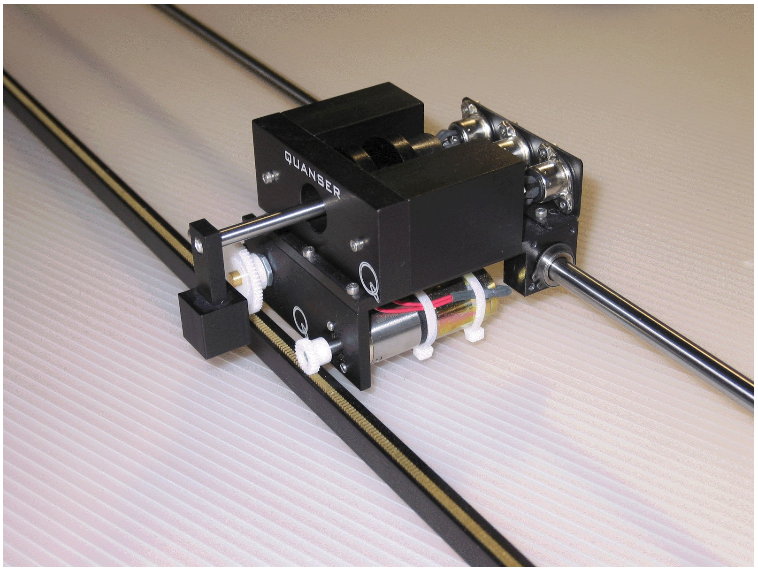


Figure 2 IP02 System

5. IP01 or IP02 and QUARC Integration

5.1. Applying a Voltage to the DC Motor

The IP01 or IP02 DC motor is driven by an amplifier (e.g. VoltPAQ) suitable to deliver the desired power to it. In order to drive a voltage to the motor, you need to output the desired voltage to the desired D/A (i.e. Digital-to-Analog) channel of your data acquisition card (e.g., Q8, Q4), that is to say the D/A channel connected to the power amplifier. From the wiring you performed, following the instructions described in [1], it is known that D/A channel #0 on the data acquisition card drives the amplifier, which in turn drives the IP01 or IP02 DC motor.

5.1.1. Creating the Model

5.1.1.1. Objective

Create a Simulink model similar as shown in Figure 3 to generate a desired voltage from analog output # 0 of the data acquisition card.

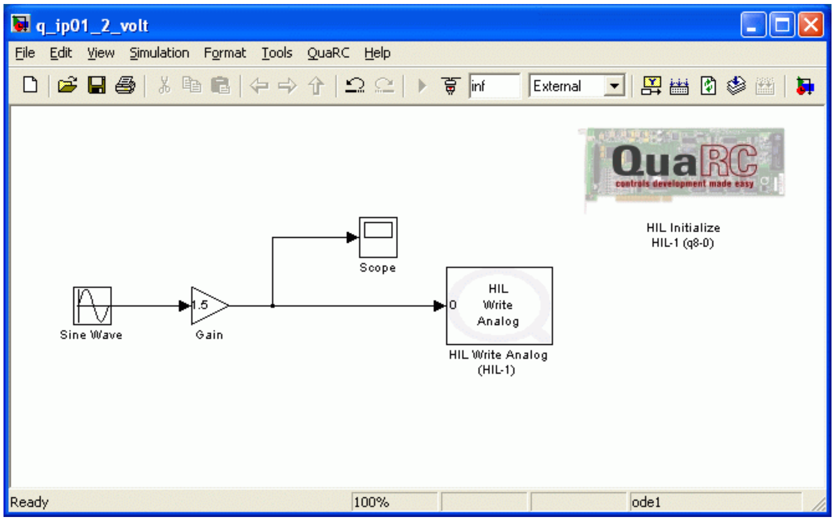
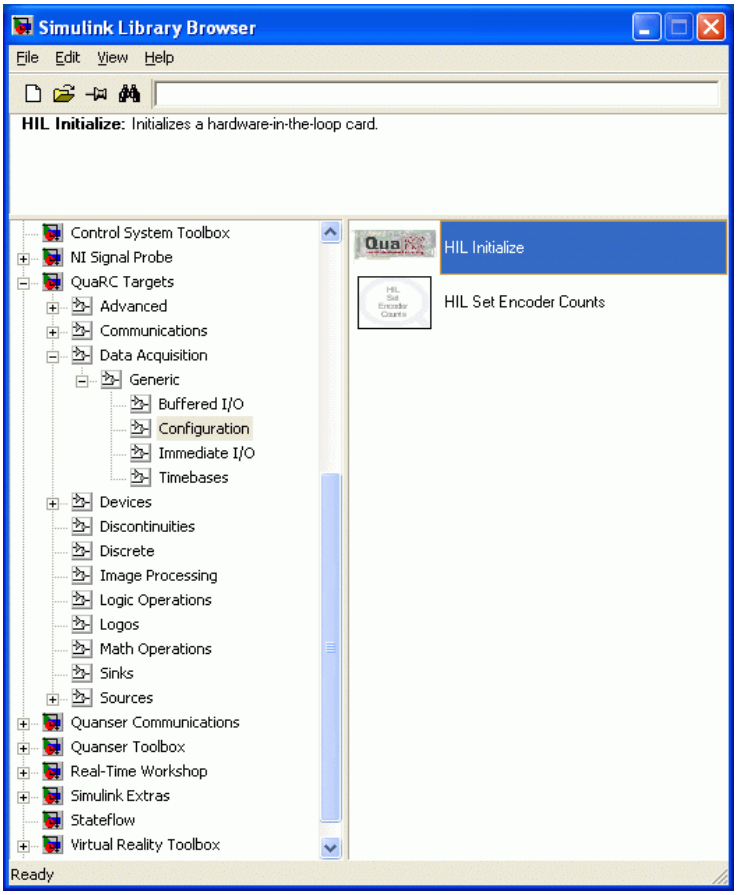


Figure 3 Diagram to Apply Sine Wave Voltage to the D/A

5.1.1.2. Procedure

To output a slowing-varying sine wave voltage of amplitude 1.5 V from the Digital-to-Analog Converter (DAC) of the board that you have installed, follow the steps described below:

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 4, 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 Q4 or Q8 hardware-in-the-loop 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.



HIL Initialize: Initializes a hardware-in-the-loop card.

Figure 4: 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.

Add the Sine Wave block, found in the *Simulink \ Source* folder, and the Gain block, from the *Simulink \ Math Operations* category, into the Simulink model. Connect the blocks as shown in Figure 3.

8. Keep the default parameters for the Simulink sine wave block, that is to say, an amplitude of 1, a frequency of 1 rad/s, and zero bias and phase. Therefore, the sine wave amplitude, as fed to the analog output block, is set by the subsequent gain block to 1.5 V.
9. Double-click on the HIL Initialize block.
10. In the *Board type* field, select the the board that is installed in your PC and which will be used to communicate with the IP01_2 system, e.g. *q4* or *q8*.

    ⚠ **Please have your lab assistant confirm the type of data acquisition card (e.g., Q8, Q4) that is installed in the computer and that you are using. Ensure the *HIL Initialize* block is configured for the data-acquisition device installed in your PC and that you will be using.**
11. If more than one board of same type 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 should 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 1, 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.

⚠ **CAUTION:**

If a "To Load" cable (as described in Reference [1]) with a gain other than one is being used (to connect the amplifier to the DC motor), a Gain block should be inserted before the Analog Output block with a gain of "1 / cable gain" in order to obtain overall closed-loop unity gain.

5.1.2. Compiling the Model

5.1.2.1. Objective

To compile the designed model into real-time code that will be executable by QUARC.

5.1.2.2. Procedure

Follow these steps to generate code from a Simulink diagram:

1. In the Simulink diagram designed in Section 5.1.1, 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). This is shown in Figure 5.

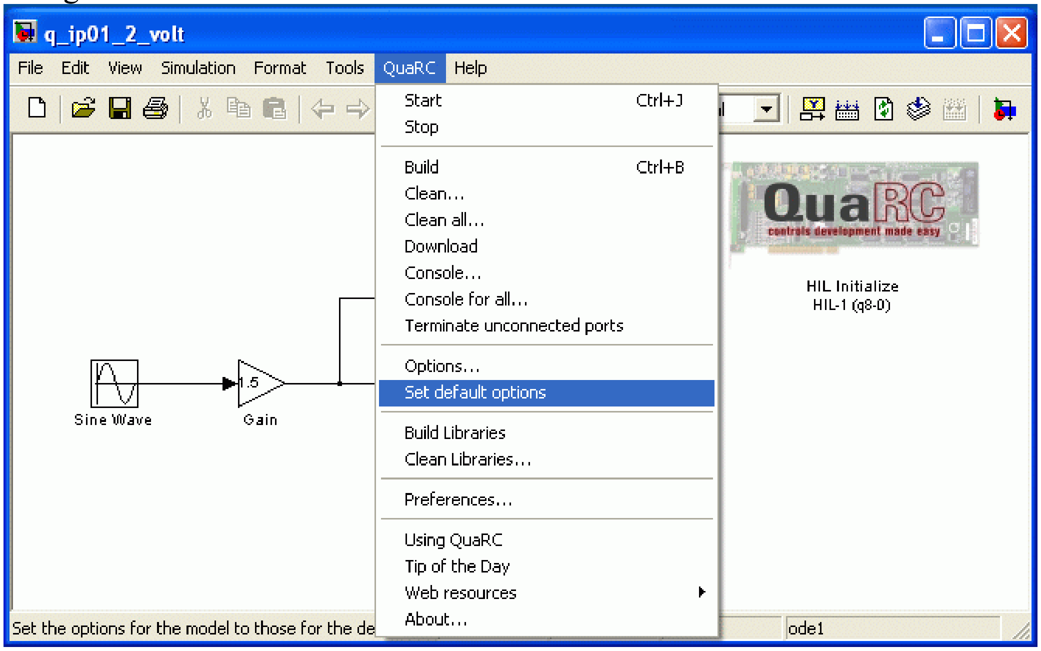
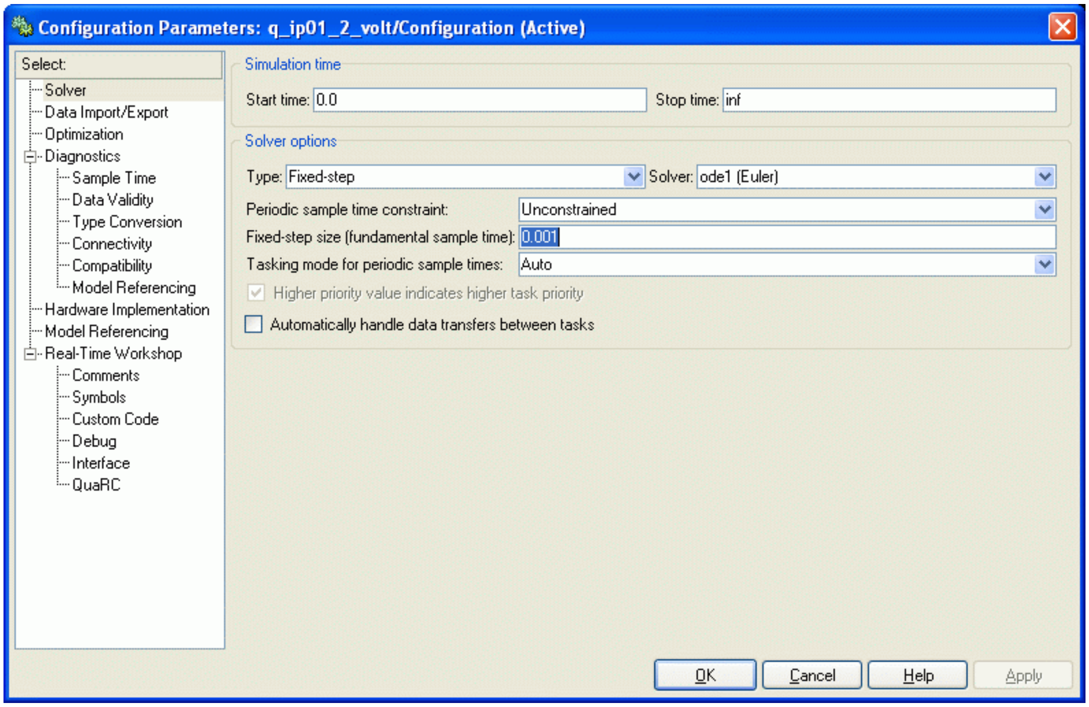


Figure 5: Setting up a Simulink diagram to generate code using QuaRC.

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 to open the window shown in Figure 6.



Configuration Parameters: q_ip01_2_volt/Configuration (Active)

Simulation time

Start time: 0.0 Stop time: inf

Solver options

Type: Fixed-step Solver: ode1 (Euler)

Periodic sample time constraint: Unconstrained

Fixed-step size (fundamental sample time): 0.001

Tasking mode for periodic sample times: Auto

☑ Higher priority value indicates higher task priority

□ Automatically handle data transfers between tasks

Figure 6: Configuring the Simulink model solver.

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. Click on the OK button to close the Configuration Parameters window.
7. Select the *QUARC* | *Build* item. Various lines in the Matlab Command Window should be displayed as the model is being compiled.
8. 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.
9. Note that the executable and associated code folder may be removed from the current directory by clicking on *QUARC* | *Clean* item.

5.1.3. Running the Real-Time Code

5.1.3.1. Objective

To use QUARC to start and stop the real-time code.

⚠ ***CAUTION*:**
Running the previously designed Simulink model with QUARC will effectively command a 1.5-Volt sine wave to the DC motor of your IP01 or IP02. The generated voltage will go from the card analog output channel 0 through the power amplifier and finally to the motor. This corresponds to an open-loop test since no feedback signal is used. **Therefore, before starting the model (and sending the voltage), ensure that the cart is located around the track mid-stroke position and is free to move in both directions.**

5.1.3.2. Procedure

Now that the code has been compiled and downloaded, it can be run in real-time. To do so and effectively command a 1.5 V sine wave to your IP01 or IP02, follow the steps described below:

1. Turn ON the power amplifier.
2. To begin executing the code, click on the *QUARC* | *Start* item in the Simulink model. The IP01_2 should begin moving back-and-forth along the track. 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. Changing the value of the Gain block on-line will change the amplitude of the commanded voltage and will directly affect the speed of the motor/cart. **Be aware that a larger voltage will not only result in an increase the cart velocity but also the distance travelled!** Changing the sign of the Gain block value will change the cart direction. Similarly, you can change the frequency of the signal by varying the Frequency parameter of the Sine Wave block.
4. Select the *QUARC* | *Stop* item to stop the code from running (or click on the *Stop* button in the Simulink model tool bar).
5. Power OFF the amplifier if no more experiments will be ran in this session.

5.2. Measuring from the IP01 Cart Potentiometer

5.2.1. Objective

To monitor and measure the actual position of the IP01 cart, as sensed by the cart potentiometer using a Simulink model similar to as shown in Figure 7 with QUARC.

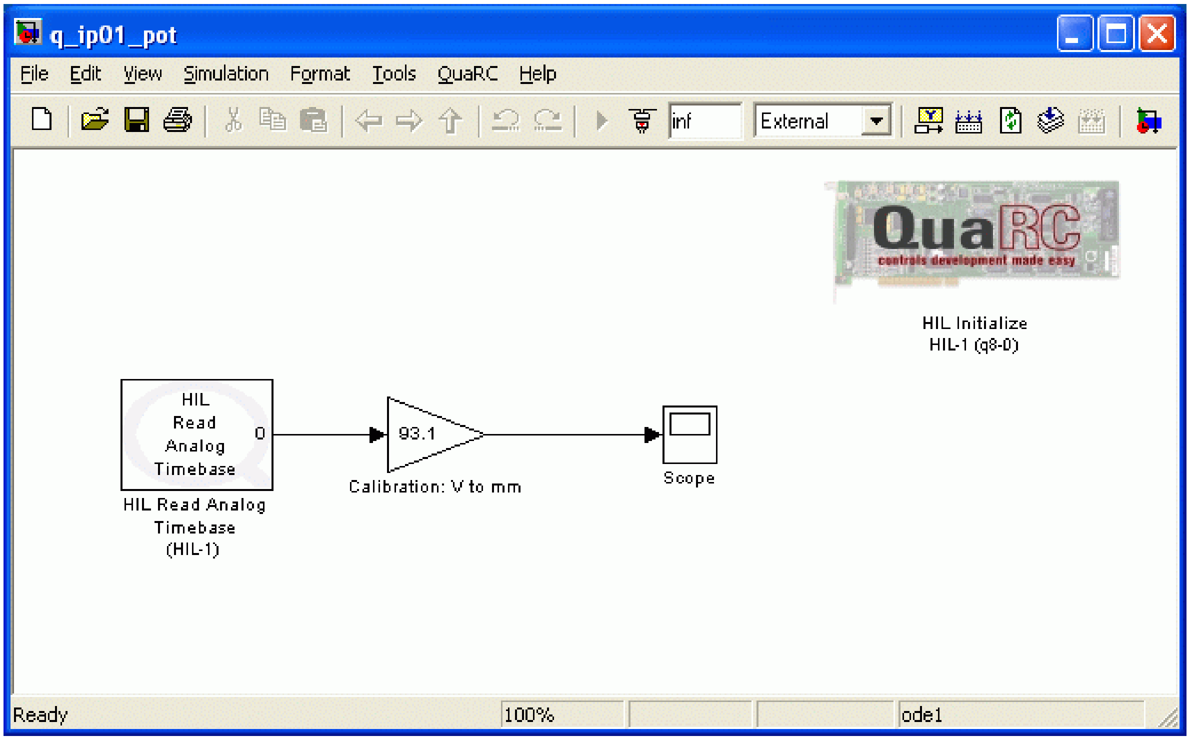


Figure 7: Simulink model used to obtain readings from the potentiometer sensor.

5.2.2. Procedure

To create a model to measure and display the IP01 cart position, follow the steps described below:

1. Create a new Simulink diagram.
2. As discussed in Section 5.1.1, add and configure the HIL Initialize block according to the DAQ that will be used (if needed, see your TA for assistance).
3. From the *Generic \ Timebases* category in the Simulink Library Browser, click-and-drag the *HIL Read Analog Timebase* block into the Simulink diagram. This block can be configured 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.

4. Double-click on the HIL Read Analog Timebase block to open its properties.
5. Set *Board name* to *HIL-1*.
6. Recall that, as instructed in Reference [1], the potentiometer is connected to Analog Input #0 on the hardware-in-the-loop board. Therefore the default setting of *Analog* c*hannels* set to 0 is fine.
7. Add a Gain block from the *Simulink \ Math Operations* library.
8. The potentiometer generates an analog signal proportional to the position pinion angle of rotation, which is in turn directly proportional to the cart position on the rack. As established in Reference [1], the IP01 cart potentiometer sensitivity is 93.1 mm/V. Therefore, such a calibration constant can be applied to the potentiometer voltage, measured from analog channel # 0, in order to obtain the cart position in mm. Enter the calibration factor of +93.1 mm/V in the Gain block following the HIL Read Analog Timebase block. The positive sign of the calibration factor should ensure that a motion of the cart towards the right-hand side of the track, which results from a positive motor input voltage as previously seen, generates an increasing position measurement from the cart potentiometer.
9. 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,
10. Double-click on the Scope icon.
11. Save the Simulink model as q_ip01_pot or something.
12. Power ON the amplifier.
13. Go to *QUARC | Build* to compile the code.
14. Click on *QUARC | Start* to execute the code. Manually move around the IP01 back-and-forth. The Scope should display the potentiometer readings similarly as shown in Figure 8.

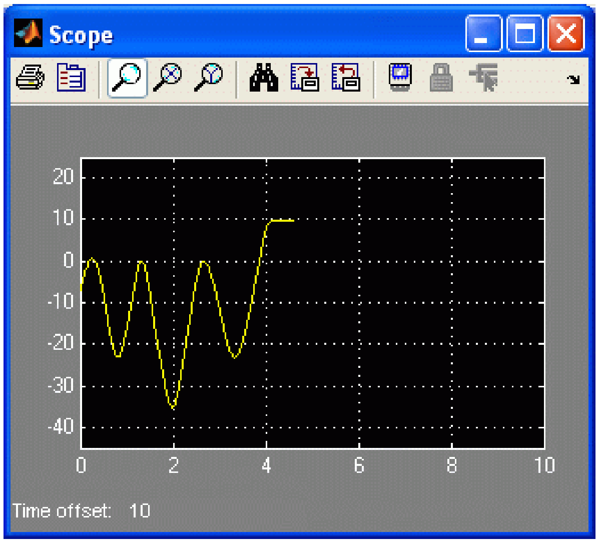


Figure 8: Reading IP01 cart position from the potentiometer.

NOTE: If the scope is not displaying any data and the SRV02 is rotating, then see Section 5.6.3 to re-configure the RTW Signal & Triggering properties.

15. Vary the sine wave frequency between 0.1 and 1 Hz and the Gain value between 0 and 2 and observe how the parameters are applied “on-the-fly”. You may also notice the discontinuity present in the potentiometer.
    Note: You can adjust the potentiometer pinion rotation relative to the rack such that the zero measurement corresponds to the desired "zero" position of the cart on the track.
16. As discussed in Reference [1], the potentiometer is an absolute sensor. To demonstrate this, set the Slider Gain to 0 V.
17. Manually rotate the gears such that the potentiometer is reading a value other than 0 V.
18. Stop the code by clicking on QUARC | Stop.
19. Start the code again, i.e. select QUARC | Start and examine how the voltage reading is the same as previously set.
20. Stop the code again.

5.3. The IP01 Open-Loop Block

5.3.1. Objective

To create a Simulink subsystem to represent the IP01 servo plant. Such a subsystem, called the IP01 block, can conveniently be used, as is, in subsequent experiments involving the IP01.

5.3.2. Procedure

To create and use a Simulink subsystem symbolizing the IP01, follow the steps described below:

Step 1. Start a fresh Simulink diagram and create a model that contains a subsystem that has the IP01 motor voltage as input and the potentiometer reading (in mm) as output, as shown in Figure 9, below. This is done by combining the models designed previously in Sections Applying a Voltage to the DC Motor and Measuring from the IP01 Cart Potentiometer.

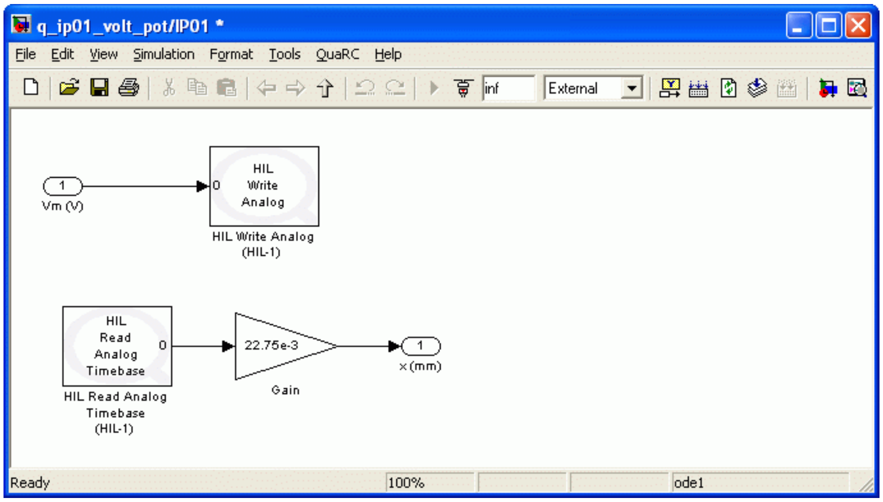


Figure 9 IP01 Subsystem

Step 2. Figure 10, below, provides an example of model using the IP01 block. Create and run in real-time the model represented in Figure 10. You should be able to replicate the observations previously made in Sections Applying a Voltage to the DC Motor and Measuring from the IP01 Cart Potentiometer.

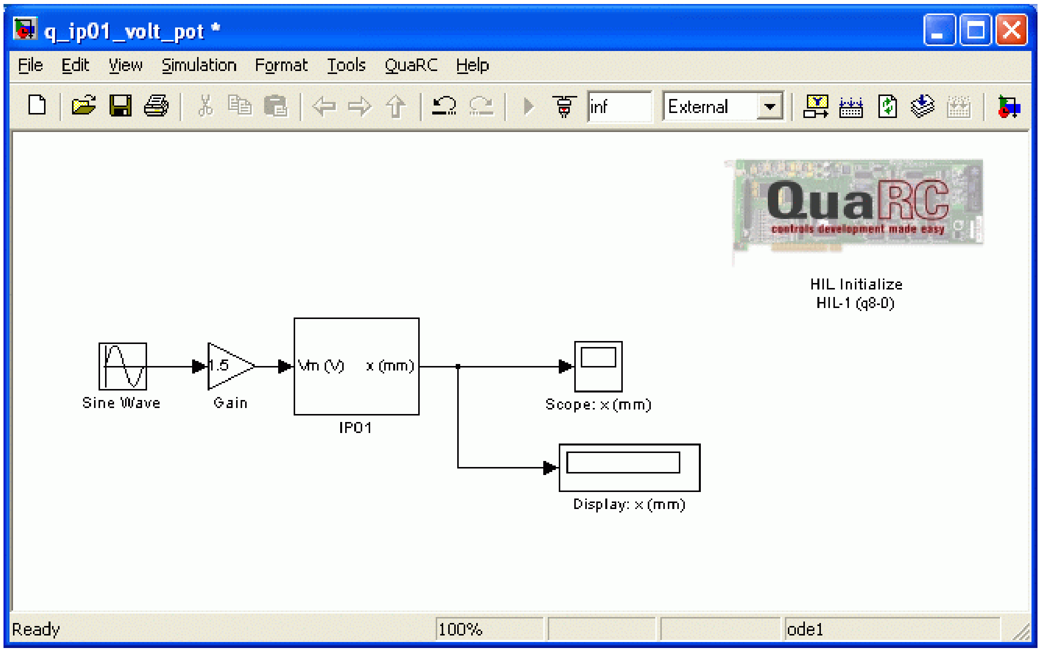


Figure 10 Model using the IP01 Subsystem

5.4. Measuring from the IP02 Cart Encoder

5.4.1. Objective

To monitor and measure the actual position of the IP02 cart, as sensed by the cart encoder, using a Simulink diagram similar to as shown Figure 11 with QUARC.

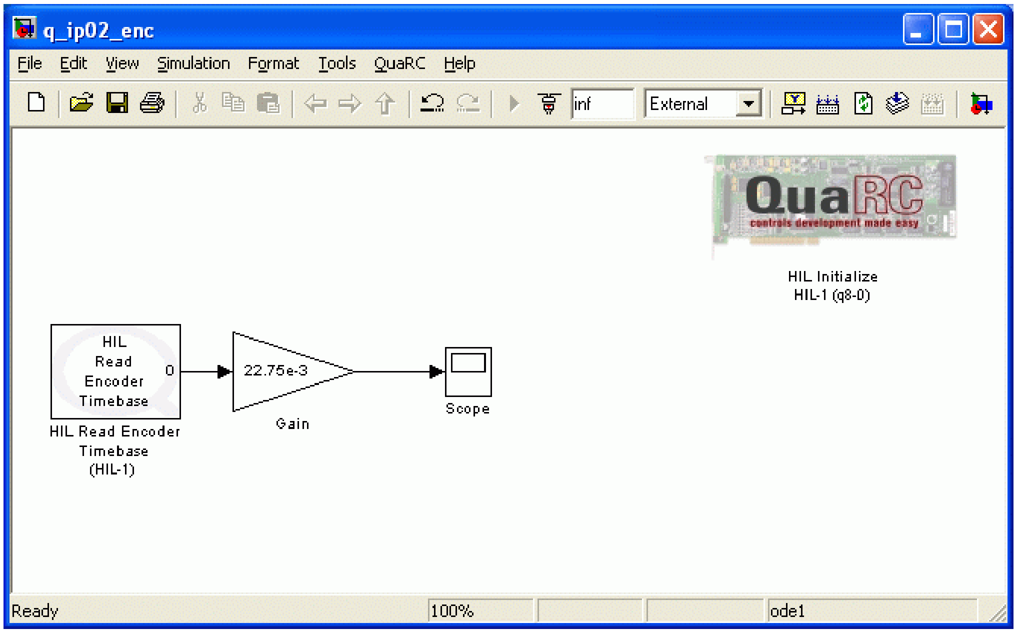

q_ip02_enc
File Edit View Simulation Format Tools QuaRC Help
inf
External
QuaRC controls development made easy
HIL Initialize
HIL-1 (q8-0)
HIL Read Encoder Timebase
0
22.75e-3
Gain
Scope
HIL Read Encoder Timebase (HIL-1)
Ready
100%
ode1

Figure 11 Measuring from the IP02 Cart Encoder

5.4.2. Procedure

The Simulink diagram designed previously is modified to include an encoder measurement, as illustrated in Figure 11, above.

Follow this procedure to read from the IP02 encoder:

1. Load Matlab and create a new Simulink diagram.
2. From the *Generic* \ *Timebases* category in the Simulink Library Browser, click-and-drag the *HIL Read Encoder Timebase* block into the Simulink diagram. This block can be configured to read the encoder input channel 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.

3. Double-click on the HIL Read Encoder Timebase block to open its properties.
4. Ensure the *Board name* is set to *HIL-1*.
5. Recall that, as instructed in Reference [1], the encoder is connected to Encoder Input #0 on the hardware-in-the-loop board. By default the *Encoder channels* field is already set to [0], so no changes are required . However keep in mind that the channel to read the encoder can be changed.
6. Add a Gain block and a Scope block from the *Math Operations* and *Sinks* folders in the Library Browser, respectively, into the Simulink model.
7. Connect the Scope and Gain blocks as depicted in Figure 11, above. Thus connect the Gain block to the output labeled *0* of the HIL Read Encoder Timebase block and connect the output of this gain block to the input of the Scope block.
8. The encoder generates an integer number of counts proportional to the position pinion angle of rotation, which in turn is directly proportional to the cart position on the rack. As established in Reference [1], the IP02 cart encoder resolution is 22.75 μm/count. Therefore, in order to obtain the cart position in mm, enter the calibration factor of +22.75e-3 mm/count in the Gain block. The positive sign of the calibration factor should ensure that a motion of the cart towards the right-hand side of the track, which results from a positive motor input voltage as previously seen, generates an increasing position measurement from the cart encoder.
9. Click on QUARC | Set Default Options.
10. Save the Simulink model, e.g. q_ip02_enc.
11. Power ON the amplifier.
12. Go to *QUARC | Build* to compile the code.
13. Click on *QUARC | Start* to execute the code.
14. Manually move the IP02 back-and-forth along the track and examine the response in the Scope. The measurement will be very small. To zoom up on the signal, click on the *Autoscale* icon in the scope or adjust the range of the y-axis manually. The Scopes should appear similarly as shown in Figure 12.

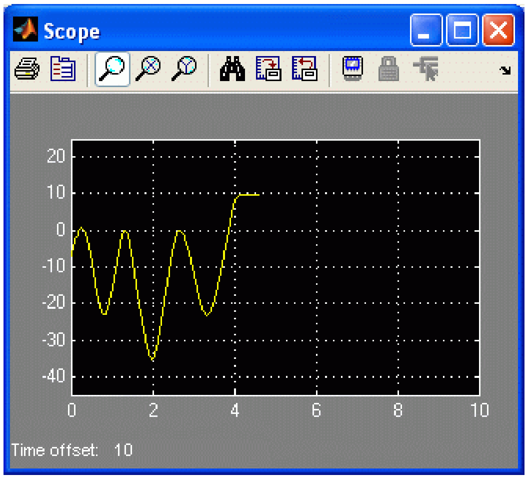


Figure 12: IP02 cart position measured with encoder.

15. Move around the IP02 cart and remember the position reading.
16. Select the *QUARC* | *Stop* item to stop the code from running.
17. Begin running the controller again. Notice in the Scope that the encoder is now reading 0 mm. 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.

5.5. The IP02 Open-Loop Block

5.5.1. Objective

To create a Simulink subsystem to represent the IP02 servo plant. Such a subsystem, called the IP02 block, can conveniently be used, as is, in subsequent experiments involving the IP02.

5.5.2. Procedure

To create and use a Simulink subsystem symbolizing the IP02, follow the steps described below:

Step 1. Start a fresh Simulink diagram and create a model that contains a subsystem that has the IP02 motor voltage as input and the encoder reading (in mm) as output. Such a subsystem (a.k.a. block) can be seen in Figure 13, below, as combining the models designed previously in Sections Applying a Voltage to the DC Motor and Measuring from the IP02 Cart Encoder.

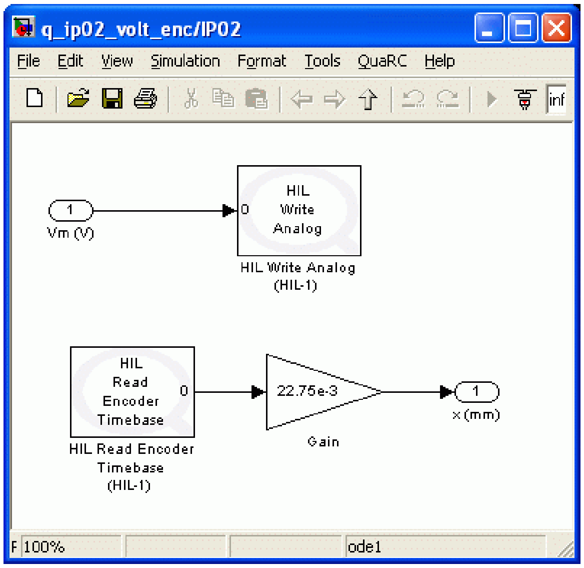

q_ip02_volt_enc/IP02
File Edit View Simulation Format Tools QuaRC Help
inf
1
Vm (V)
0
HIL Write Analog
HIL Write Analog (HIL-1)
HIL Read Encoder Timebase
0
HIL Read Encoder Timebase (HIL-1)
22.75e-3
Gain
1
x (mm)
100%
ode1

Figure 13 IP02 Subsystem

Step 3. Figure 14, below, provides an example of model using the IP02 block. Create and run in real-time the model represented in Figure 14. You should be able to replicate the observations previously made in Sections Applying a Voltage to the DC Motor and Measuring from the IP02 Cart Encoder.

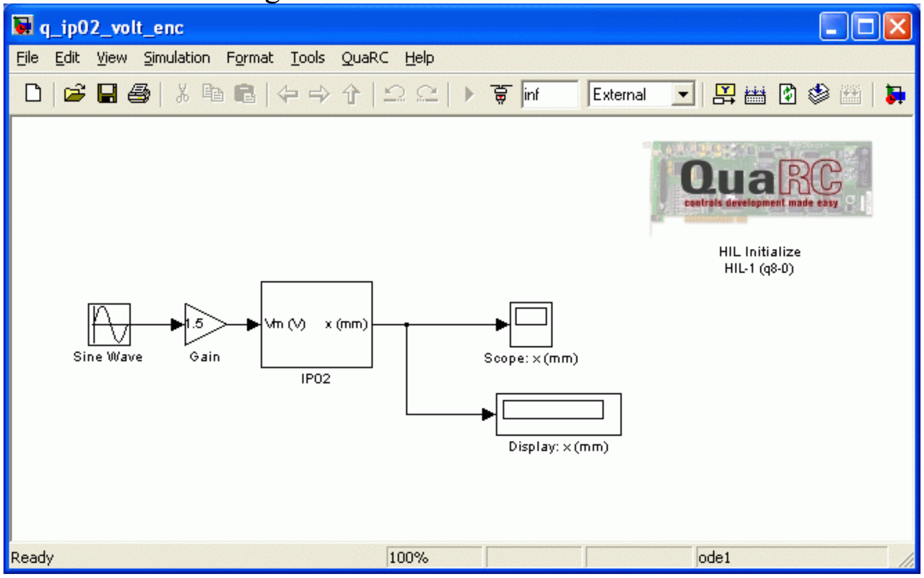


Figure 14 Model using the IP02 Subsystem

5.6. Other QUARC Functions

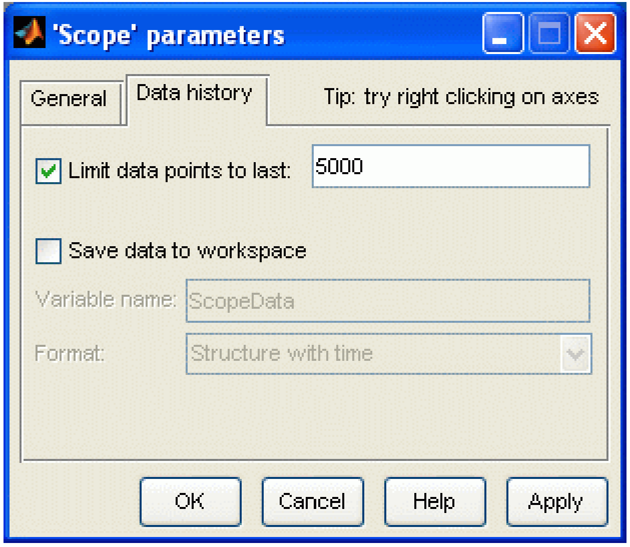
5.6.1. Digital Display

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 the Simulink diagrams shown in Figure 12 and Figure 14 for example, the measured cart position is displayed numerically.

5.6.2. Saving Scope Data

The scopes in the Simulink model can be configured to save variables in the Matlab workspace. For instance, to configure the Scope sink used in the previous Simulink models perform the following:

1. Double-click on the Scope sink.
2. Click on *Parameters* icon.
3. In the *Parameters* window, select the *Data History* tab, as shown in .



Tip: try right clicking on axes

☑ Limit data points to last: 5000

□ Save data to workspace

Variable name: ScopeData

Format: Structure with time

Figure 15: Scope data history parameters.

4. Select the *Save data to workspace* check box.
5. Change the *Variable* name field to a desired variable. For example, when measuring the cart the variable could be called *data_xc*.
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 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 is a two-dimensional array with 5000 elements. The first vector is the running time and the second vector is, for example, the position of the cart. 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 = data_xc(:,1);
% Cart position (mm)
xc = data_xc(:,2);
%
%% Plot response
plot(t,xc,'r-');
label('x_c (mm)')
xlabel('time (s)');
title('\bf IP02 Position Response');
```

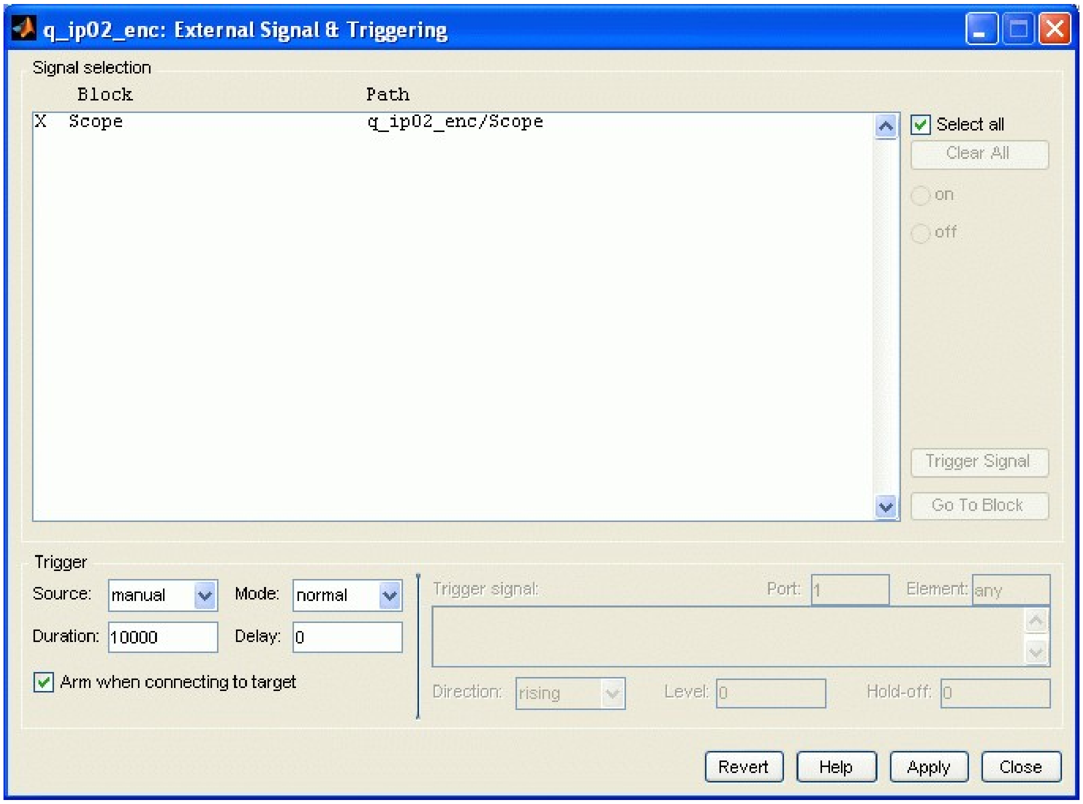
Text 1: Sample script used to plot saved variable *data_xc* in a Matlab figure.

5.6.3. 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 Error: Reference source not found will load.


q_ip02_enc: External Signal & Triggering

Signal selection

	Block	Path
X	Scope	q_ip02_enc/Scope

☑ Select all

Trigger

Source: manual Mode: normal

Duration: 10000 Delay: 0

☑ Arm when connecting to target

Trigger signal: Port: 1 Element: any

Direction: rising Level: 0 Hold-off: 0

Figure 16: Signal and Triggering window for a Simulink diagram with a single Scope.

3. As shown in Figure 16, 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.

6. Knowledge Test

As this is an introductory session, there is no report to write. You should however ensure that you have understood the principles of operation of your hardware (e.g. IP01 or IP02, Q2-USB or Q8-USB or QPID, VoltPAQ) and QUARC. You should also have a good understanding of how positions are measured and how the voltage is applied to the motor. In order to ensure that you understand the complete system, please assure that you can reply to the following questions correctly (and confidently!):

1) How many actuators and how many sensors does the IP01 system have?

2) How many actuators and how many sensors does the IP02 system have?

3) Which D/A channel drives the amplifier? the DC motor?

4) Which analog input channel measures the IP01 cart potentiometer voltage?

5) Which encoder channel measures the IP02 cart encoder counts?

6) Why is the sign of the calibration constant important?

7) Why does the cable gain have to be compensated for in the motor model?