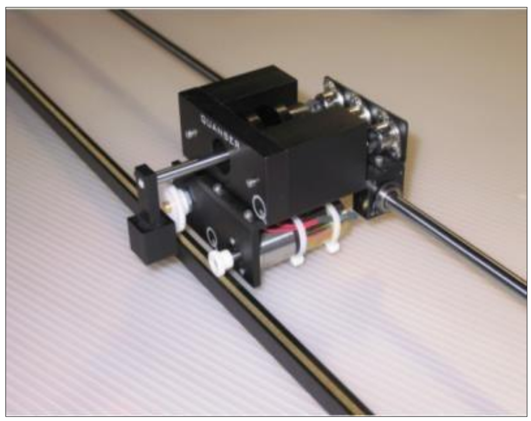
Middle East Technical University
Department of Electrical and Electronics Engineering

EE 406
Laboratory of Feedback Control Systems



Experiment #1:
Control Hardware and Software Setup, Signal Interfaces

Preliminary Work and Laboratory Manual

Initial Version: Afşar Saranlı, Emre Tuna

Group Members:

____

____

1. Objectives

In this introductory session you will become familiar with the control hardware and software that will be used through the laboratory course. This includes the Matlab-Simulink environment, the Quanser - QuaRC Windows-based real-time control software and Quanser - IP02 servo plant hardware. At the end of the session, you should 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 Simulink based signal input and output modules and run them in real-time using QuaRC.
- How to measure and display, using a Simulink diagram, the encoder, or any otheranalog input signals.
- How to apply and display, using a Simulink diagram, a varying voltage to the IP02 DC motor.
- How to construct a Simulink block (i.e., subsystem) that represents the IP02 servosystem.

2. Prerequisites

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

i) To be familiar with your IP02 main components, (e.g., actuator, sensors), your power amplifier (e.g., UPM), and your data acquisition card (e.g., Q8 or Q2), as described in References [1], [2], and [3].
ii) To be familiar with the complete wiring of your IP02 servo plant, as per dictated in Reference [1].

3. References

[1] *Quanser IP02 User Manual.*
[2] *Quanser Q2 USB Data Acquisition Device User Manual.*
[3] *Universal Power Module UPM User Manual.*
[4] *QuaRC User Manual (type* `doc quarc` *in Matlab to access).*
[5] *QuaRC Installation Manual.*

4. Experimental Setup

4.1. Main Components

This laboratory is composed of the following hardware and software components:

- **Power Module:** Quanser UPM 1503 or UPM 2405, VoltPAQ-X1
- **Data Acquisition Board:** Quanser Q8 and Q2
- **Linear Motion Servo Plant:** Quanser IP02, as shown in Figure 1 below.
- **RealTime Control Software:** The QuaRC-Simulink configuration, as detailed in the Reference [5].
- **A Computer to Run Matlab-Simulink and the QuaRC software**

The setup for this laboratory will be mostly prepared for you. Your desk setup will have the necessary components including all necessary software components pre-installed. You will follow the procedure of this laboratory session to setup and use the system.

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 IP02 as fully described in Reference [1]. When you are confident with your connections, you can power up the UPM.

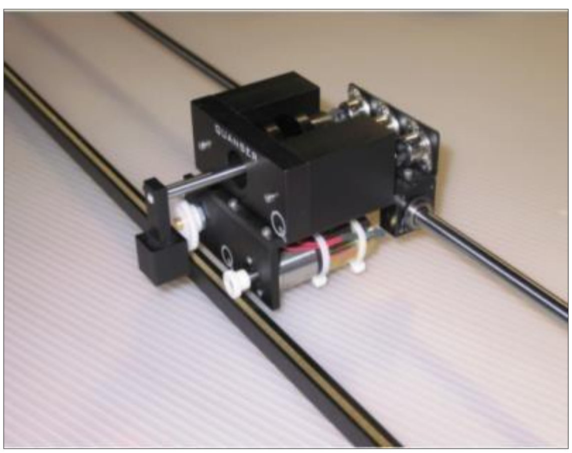


Figure 1 – The IP02 Servo Plant

5. IP02 and QuaRC Software Integration

5.1. Applying a Voltage to the DC Motor

The IP02 DC motor is driven by a motor amplifier (e.g., UPM) 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 (i.e., Q2), 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 UPM, which in turn drives the IP02 DC motor.

5.1.1. Creating the Model

Objective

Create a Simulink model similar as shown in Figure 2 to generate a desired voltage from analog output # 0 of the data acquisition card.

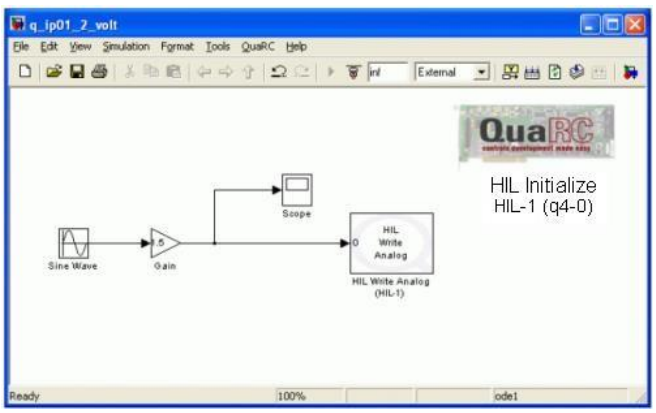


Figure 2 – Simulink diagram to apply a Sine wave to the D/A converter.

Procedure

To output a slowly-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 (Run) 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 3, expand the *QuaRC Targets* item and go to the *Data Acquisition \ Generic \ Configuration* folder.
5. Click-and-drag (Not double-click) 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 Q2 (Q8) hardware-in-the-loop DAC 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.

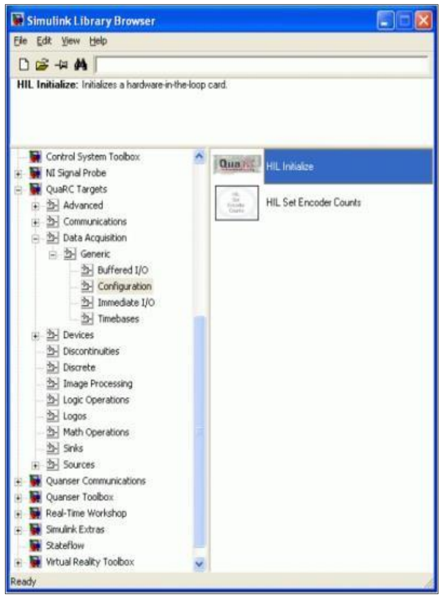


Figure 3 – QuaRC Configuration folder in the Simulink Library Browser

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 on the data-acquisition device.
   Add the "Sine Wave" block, found in the *Simulink \ Sources* folder; "Scope" block in *Simulink \ Sinks* folder and the Gain block, from the *Simulink \ Math Operations* category, into the Simulink model. Connect the blocks as shown in Figure 2.
8. Keep the default parameters for the Simulink sine wave block, that is to say, 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. (Use "." as decimal indicator. Not ",")
9. Double-click on the HIL Initialize block.
10. In the *Board type* field, select the board that is installed in your PC, and which will be used to communicate with the IP02 system. The boards that are in our setups are *q2 and q8 boards*.

Please note that the correct setting of the DAC card in your system is very important. Have your lab assistant check that you are indeed working with a Q2 board and that you have a correct setting here.

11. The default settings for the board should be fine for this experiment. For more information on this block, click on its *Help* button. Otherwise click on the OK

button and proceed. Note that *Help* button can be used to see documentation for all blocks having the same button.
12. Double-click on the HIL Write Analog block.
13. Set *Board name* to *HIL-1*. By default, *Channels* is set to 0 and the *Sample time* field is set to -1. As instructed in Reference 1, the DC motor should be 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.
14. 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 UPM 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 in your Simulink model. This should normally not be the case in your setup.

5.1.2. Compiling the Model

Objective

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

Procedure

Follow these steps to generate code from a Simulink diagram:
1. On your experiment PC, change to the temporary experiment directory (Current Matlab directory) defined by your lab assistant and save the Simulink diagram you defined in Section 5.1.1.
2. For this Simulink diagram, 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 4.

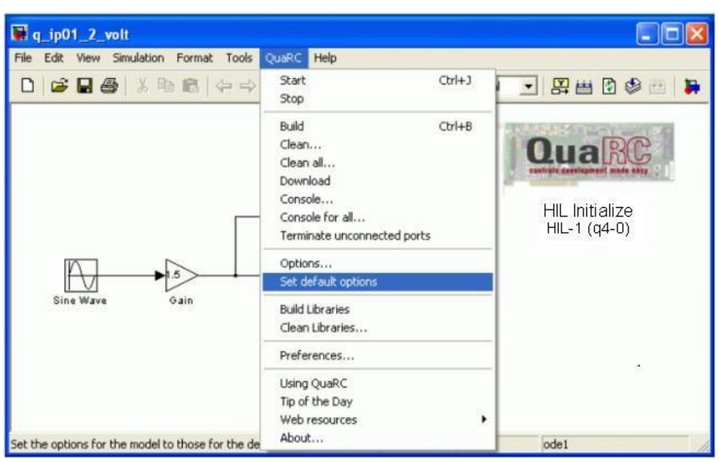


Figure 4 – Setting up a Simulink diagram to generate code using QuaRC.

3. 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* inside code generation 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. If not, ask help from your assistant.
4. Click on the *Solver* item in the left-hand pane to open the window shown in Figure5.

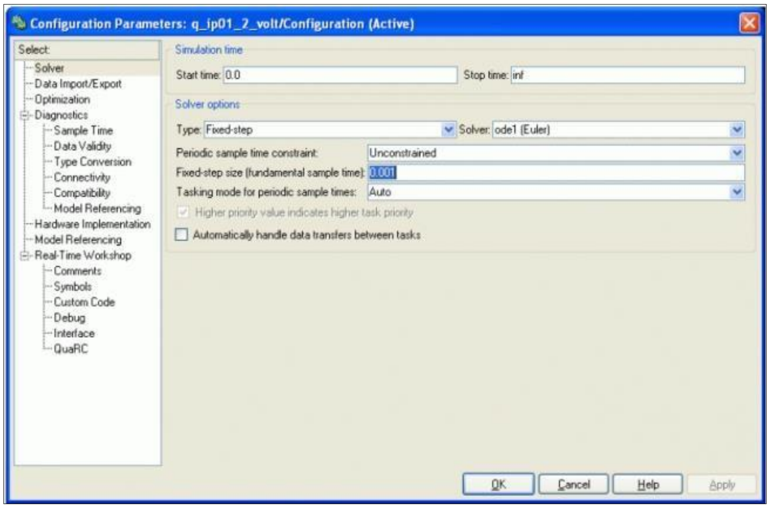


Figure 5 – Configuring the Simulink model solver.

5. 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).
6. 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 be changed to an integration method such as *ode1 (Euler)*.
7. 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.
8. Select the *QuaRC | Build* item. Various lines in the Matlab Command Window should be displayed as the model is being compiled.

REMARK: Always save your model before “Building” it to make modifications valid.

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.

5.1.3. Running the Real-Time Code

Objective

Using 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 IP02 Servo Plant. 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.**

Procedure

Now that the code has been compiled and downloaded (Build does both of them). Procedure of building can be traced by following information on Matlab command screen. Now, the model can be run in real-time. To do so and effectively command a 1.5 V sine wave to your IP02, follow the steps described below:

1. Turn ON the power of the Quanser UPM (or the equivalent power amplifier).
2. To begin executing the code, click on the *QuaRC | Start* item in the Simulink model. The IP02 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, you 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 UPM whenever you finish an experimental session.

5.2. Measuring from the IP02 Cart Encoder

5.2.1. 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 6 with QuaRC.

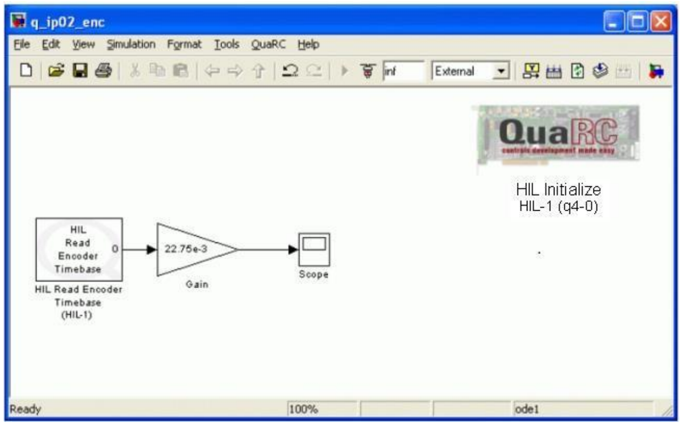


Figure 6 – Measuring from the IP02 Cart Encoder

5.2.2. Procedure

The Simulink diagram designed previously is modified to include an encoder measurement, as illustrated in Figure 6, above.

Follow this procedure to read from the IP02 encoder:

1. Load Matlab and create a new Simulink diagram.
2. From *QUARC Targets \ Data Acquisiton \ Generic \ Configuration*, drag-and-drop "*HIL Initilize*" block to the model. Make sure that its *board type* is q2_usb (q8_usb). If not, double click on *HIL Initilize* block and change *board type* to q2_usb (q8_usb) and *boardidentifier* to 0.
3. From the *QUARC Targets \ Data Acquisition \ 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 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 Encoder Timebase block to open its properties.
5. Ensure the *Board name* is set to *HIL-1*.
6. 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.

7. Add a Gain block and a Scope block from the *Simulink \ Math Operations* and *Sinks* folders in the Library Browser, respectively, into the Simulink model.
8. Connect the Scope and Gain blocks as depicted in Figure 6, 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.
9. 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.
10. Click on QuaRC | Set Default Options.
11. Save the Simulink model, e.g., exp1_part_5_2.
12. Power ON the UPM.
13. Go to *QuaRC | Build* to compile the code.
14. Click on *QuaRC | Start* to execute the code.
15. Manually move the IP02 back-and-forth along the track and examine the response in the Scope. In order to see Scope screen, just double click on scope block. 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 7.

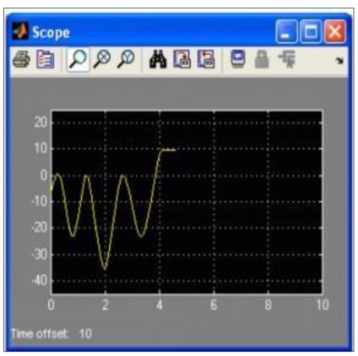


Figure 7 - IP02 cart position measured with encoder.

16. Move around the IP02 cart and remember the position reading.
17. Select the *QuaRC | Stop* item to stop the code from running.
18. 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. the voltage value read from a potentiometer.
19. Stop running the controller.
20. Note: As a general rule, do not forget to power OFF the UPM whenever you are completely finished with your experiments.

5.3. The IP02 Open-Loop Block

5.3.1. Objective

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

5.3.2. Procedure

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

1. Start a fresh Simulink diagram.
2. Draw the model in Figure 8. Necessary components are as follows:
    - *QUARC Targets \ Data Acquisition \ Generic \ Configuration \ HIL Initialize*.Make sure that its *board type* is q2_usb (q8_usb) and *board identifier* is 0.
    - *Simulink \ Sources \ Sine Wave*.
    - *Simulink \ Math Operations \ Gain*.
    - *Simulink \ Sinks \ Scope*.
    - *Simulink \ Sinks \ Display*.
    - *QUARC Targets \ Data Acquisition \ Generic \ Immediate I/O \ HIL Write Analog*. Make sure that it is assigned to HIL-1 and *channel 0* is set. If not, change them by double clicking on "*HIL Write Analog*" block. See *Board Name* and *Channels* on the opened window.
    - *QUARC Targets \ Data Acquisition \ Generic \ Timebases \ HIL Read Encoder Timebase*. Make sure that it is assigned to HIL-1 and *channel 0* is set. If not, change them by double clicking on "*HIL Read Encoder Timebase*" block. See *Board Name* and *Channels* on the opened window.

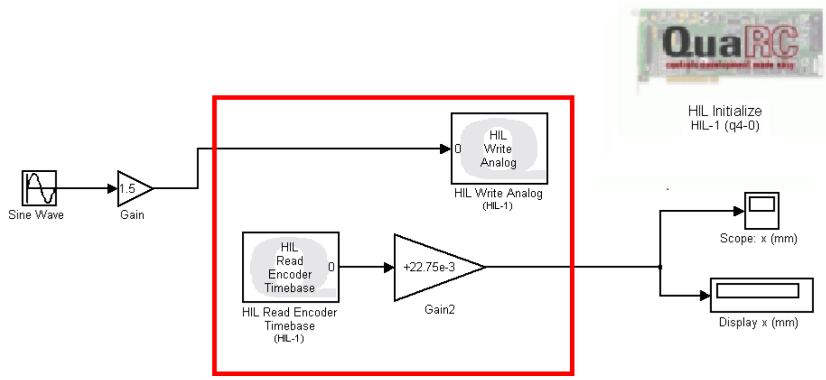


Figure 8: Initial Model (Looks complicated)

3. Now, the model is functionally ready. But it looks complicated. Even though, it may not look very complicated for the time being, learning systematic model design will be beneficial when the system gets much more complicated. So, IP02 system will be represented as a subsystem to have a clearer model.
   Chose blocks inside of the red square in Figure 8. Press either "Ctrl+g" or "Edit -> Create Subsystem" from main menu bar.
   For more information on subsystems: *Matlab Product Help* / *Simulink* / *User Guide* / *Creating a Model* / *Creating Subsystems*.
4. At this step, you have a model, but subsystem and port names are assigned as default. Change name of subsystem to "IP02".
5. Enlarge the block figure from one of its corners.
6. Go inside of the subsystem "IP02" by double clicking on it.
7. Change input port name from "In1" to "Vm (V)" and output port name from "Out1" to "x (mm)". Your subsystem should be similar to the one in Figure 9.
8. Save the subsystem and return to the main model.
9. Your main model should look like the one in Figure 10.

NOTE: Subsystems can be *reused* in future models. But never forget to have HIL Initialize block and assign correct board type to input and output blocks of the board (like q2/ HIL-1) in models. Otherwise, you will have error.

REMARK: Reasons of errors in Simulink can be easily traced. Just carefully read any error message that may pop-up.

10. Power ON the UPM.
11. Go to *QuaRC* | *Build* to compile the code (model).
12. Click on *QuaRC* | *Start* to execute the code.
13. You should be able to replicate the observations previously made in Sections on Applying a Voltage to the DC Motor and Measuring from the IP02 Cart Encoder.

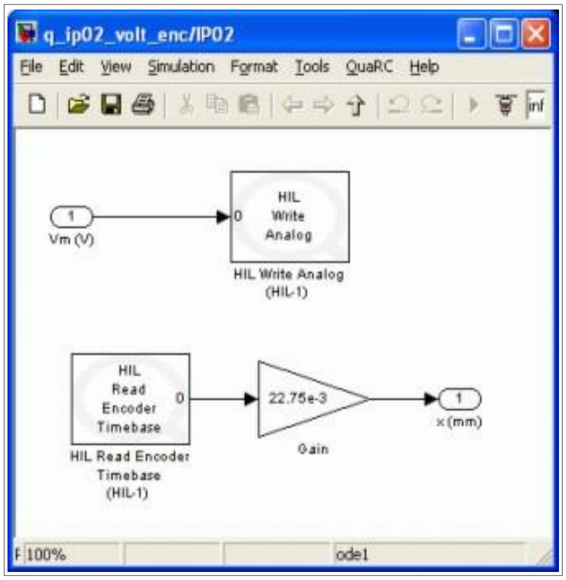


Figure 9 – Inside design of the IP02 Simulink Blok

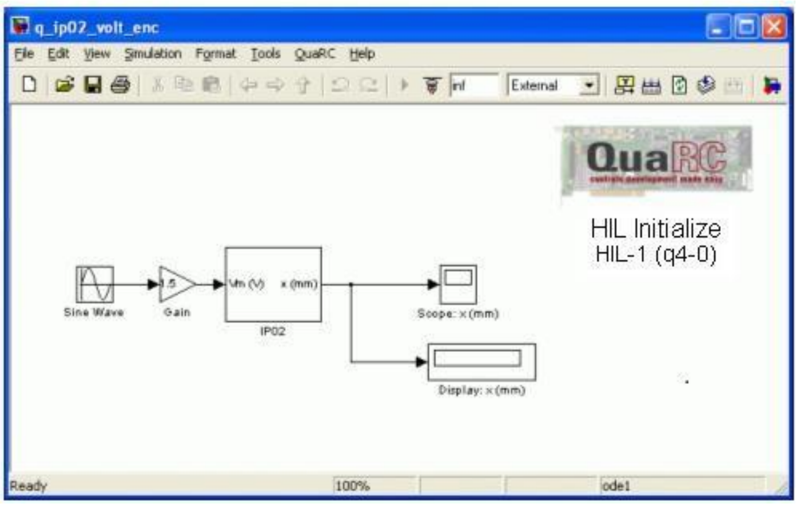


Figure 10 – Simulink model using the IP02 Subsystem

5.4. Other QuaRC Functions

5.4.1. 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 10 for example, the measured cart position is displayed numerically.

5.4.2. 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 *Configuration Properties* icon.
3. In the *Configuration Properties* window, select the *Logging* tab, as shown in Figure 11.

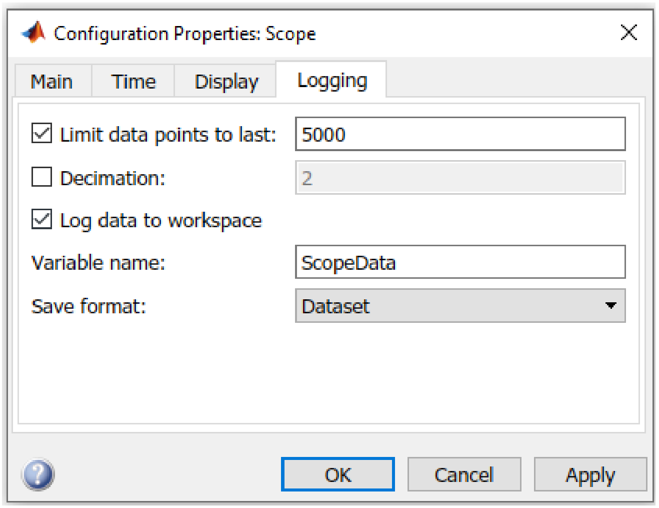


Figure 11 – Scope data history parameters

4. Select (Check) 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. Similarly, if the controller runs at 500Hz, 10 seconds of data will be displayed.
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 run 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 of Matlab.

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, create and save the following Matlab script (with appropriate corrections). Execute the script and observe the data plot.

REMARK: You can open a new M-File by clicking *File* | *New* | *Blank M-File*.

```
%% Example of Loading and Plotting 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-');
ylabel('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.

5.4.3. 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.

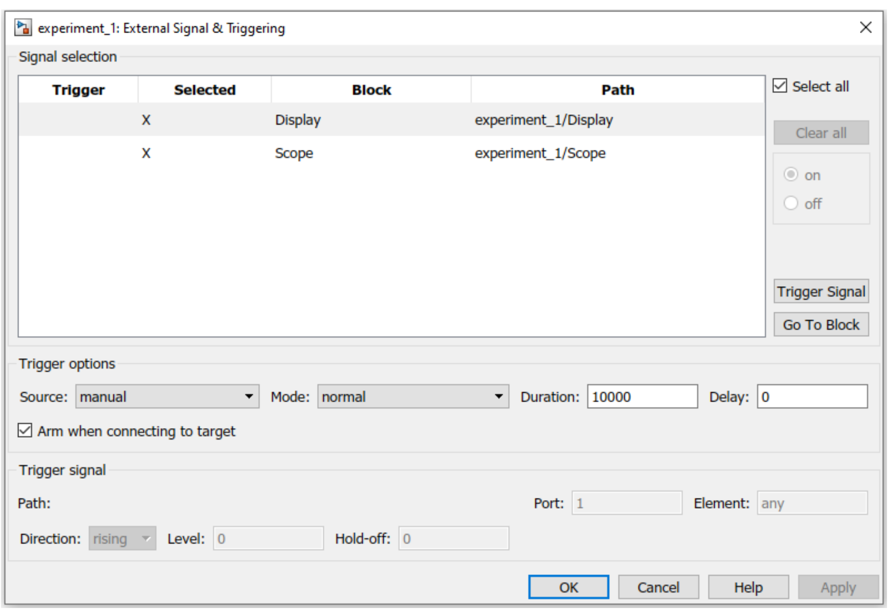


Figure 12 – Signal and triggering diagram for a Simulink diagram with a single Scope

Follow this procedure to adjust these settings:

1. In the Simulink model, select the *Code | External Mode Control Panel* item.
2. Click on the *Signal & Triggering* button and the window shown in Figure 12 will load.
3. 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 number 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

This is an introductory session. You should ensure that you have understood the principles of operation of your hardware (e.g., IP02, Q2, UPM) and QuaRC. You should also have a good understanding of how positions are measured and how the voltage is applied to the motor. Reply as a group to the following questions in the space provided.

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

2) Which D/A channel drives the UPM? What is function of the UPM?

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

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

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

6) What is the function of the QuaRC "HIL Initialize" block inside the Simulink model?

6.Signatures

Names, Student IDs, and Signatures of Group Students:

- ____
  ____
- ____
  ____

Date of Experiment

____

Name and Signature of Lab Assistant:

____