

Chapter 4

Introduction to Data Acquisition and Real-Time Control

The objectives of this laboratory session are:

1. Gain familiarity with the Rotary Motion Servo Plant SRV-02, the Data Acquisition Card NI PCI-6251, and the QuaRC Software.
2. Understand the basic input/output (I/O) connections.
3. Design a Simulink model using blocks from the QuaRC library to run the DC motor on the SRV-02.
4. Design a Simulink model using blocks from the QuaRC library to measure the various signals on the SRV-02.

4.1 Pre-Lab Assignment

Read this entire chapter as well as the references [1, 2, 3]!

Digital control of a continuous-time systems has become very popular as the price and reliability of digital computers has improved. Digital sensors (like thermocouples in thermostats) and a computer (like an integrated circuit) perform calculations to emulate the analog controllers (like bi-metal strips) they replace. Very complicated control structures can be implemented easily using a digital controller, whereas an analog controller would require complex hardware. Digital controllers are more adaptable to parameter changes than their analog counterparts. Furthermore, analog emulation and real-time control provide advanced features such as adaptive self-tuning, multivariable control, expert systems, and the ability to communicate over a local area network.

Typically, the computer replaces the cascade controller. The measured data is converted from analog to digital form by the analog-to-digital converter (ADC). This is accomplished by the data acquisition system (DAQ) which is the “eyes and ears” of the digital computer. The computer receives and manipulates the signal in digital (binary) form, and the output is then converted to analog by the digital-to-analog converter (DAC). This process is the “hands and arms” of the computer.

The full scale output is usually determined by an external reference voltage. The DAC resolution is defined as the smallest possible change in output. For an N -bit converter the resolution is given by Eq. (4.1).

$$\text{Resolution} = \frac{100}{2^N} \% \quad (4.1)$$

For example, the resolution of a 10-bit DAC is

$$\frac{100}{2^{10}} = \frac{100}{1024} = 0.09766\%$$

The resolution of an ADC is defined as the smallest detectible change in input, which is also given by (4.1).

When the computer reacts to external events as they occur, it is referred to as real-time control. An important consideration in real-time control is the update rate. During one controller cycle, three things must happen before the next cycle can begin: 1) Sensors are read (ADC inputs), 2) The microprocessor computes updated commands, and 3) The commands are converted to analog signals (DAC outputs). Between cycles, the commands outputs are held constant. If the controller is updated every 0.01 seconds, the output of the controller would look like a staircase with steps 0.01 seconds wide. As expected, the faster the dynamics of the system being controlled (pouring water versus pouring molasses), the faster the update rate must be.

Modern real-time control systems use computer aided software engineering (CASE) such as MATLAB's Real-Time Workshop to automatically generate C code from a graphical control model (like Simulink). Often the sensors or other components of the DAQ system are proprietary. Vendors now provide Simulink blocks along with their hardware in order to ease the graphical modeling of their product.

In this lab, we will focus on gaining familiarity with the Data Acquisition Card NI PCI-6251 and the QuaRC software. In addition, you will explore the effects of A/D and D/A conversion that are built into the NI PCI-6251 through which all the control signals pass.

4.2 Laboratory Procedure

The laboratory procedure consists of two main steps.

1. Wiring the equipment. For this, you will follow the wiring procedure detailed in Section 6 of [1] with some minor changes. The Data Acquisition Card we have is an NI PCI-6251 instead of a Quanser Q4 or Q8. As a result, we use an appropriate Quanser-NI Terminal Board to access NI PCI-6251, which is slightly different from the Quanser Q8 terminal board mentioned in [1]. The minor changes in the wiring due to this difference are pointed out in subsection 4.2.1.
2. Building and running a QuaRC code using Simulink to run the SRV-02 as well as to measure various signals from the SRV-02. For this, you will follow [3] with some minor changes. These changes are again due to the difference in our Data Acquisition Card. Some of the QuaRC blocks mentioned in [3] are not available for NI PCI-6251, and hence they will have to be replaced by those blocks that have similar functions and are available for NI PCI-6251. These changes are outlined in subsection 4.2.2. For more information, see [4] or type "doc quarc" in the Matlab prompt and go to User's guide → QuaRC Data Acquisition card Support.

4.2.1 Wiring Procedure

This experiment requires four items to be connected:

1. Rotary Motion Servo Plant SRV-02
2. Quanser-NI Terminal Board
3. Universal power module (UPM)
4. Data Acquisition Card NI PCI-6251 – This card is installed within the PC, hence it is not readily visible, but it can be accessed through the NI cable with a 68-pin blue connector lying on the bench.

The proper connection of these elements is crucial—and particularly confounding in the tight spaces on the lab bench. Follow the steps below to wire the equipment:

1. Turn off UPM using the “kill switch” located on the back of the UPM.
Important: This is the only fail-safe method to cut power to the SRV-02 system. Whenever you run an experiment, one team member must be assigned to operate this power switch. Mistakes *do* happen, so to protect yourself, your teammates, and the equipment, always have someone ready to cut power in the event of an accident.
2. Turn off the PC.
3. Follow the steps in subsections 6.1 and 6.2 of [1] to wire the equipment with the following changes:
 - (a) As mentioned above, Quanser-NI Terminal Board is slightly different from the one in [1]. However, similar to the one in [1], Quanser-NI Terminal Board has all the analog I/O channels and the encoder input channels mentioned in [1], although the numbers and the locations of these channels are different from the one shown in [1].
 - (b) Another minor difference is that Quanser-NI Terminal Board will be connected to NI PCI-6251 using the NI cable with a 68-pin blue connector lying on the bench. Figure 4.1, which is to replace Figure 21 in [1], shows Quanser-NI Terminal Board and the connections.
 - (c) Finally, make sure that the jumper on the analog input channels #0-3 of the Quanser-NI Terminal Board is set to “D”, which stands for differential (In QuaRC, the analog mode is set to the default mode for the Data Acquisition Card, which is differential for NI cards. As a result, the jumper on the Quanser-NI Terminal Board must be configured to match the default analog mode of NI PCI-6251).
4. Have the instructor check your wiring.
5. Turn on the PC. Make sure the LED on the terminal board is illuminated. If it is not, then the fuse needs replacement. Contact the instructor.

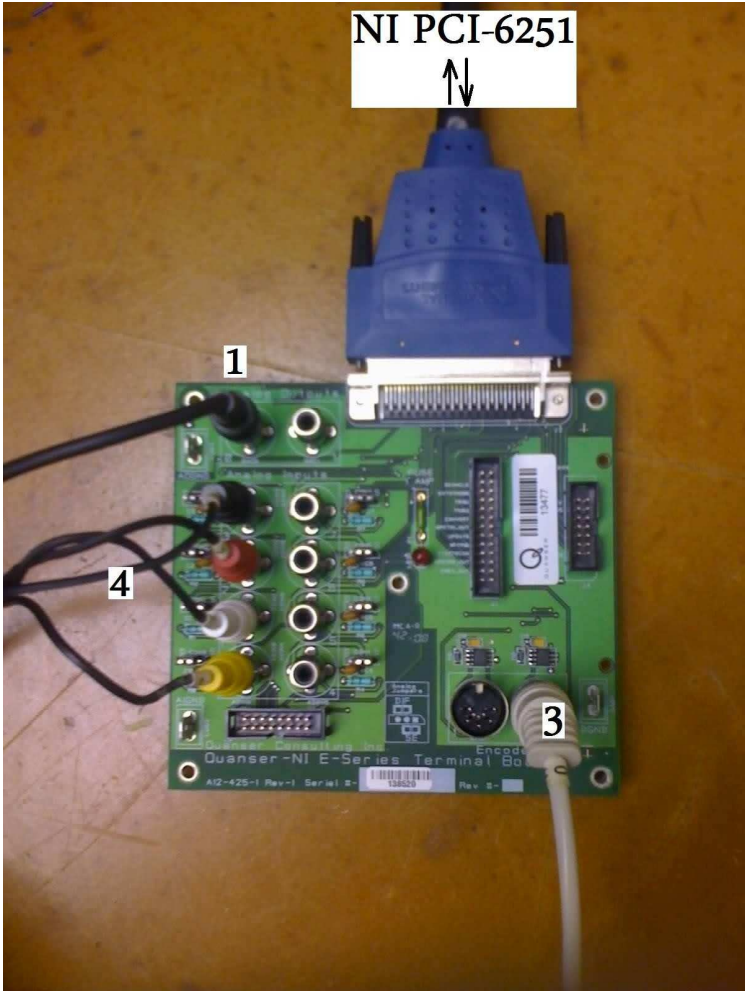


Figure 4.1: Quanser NI terminal Board.

4.2.2 Building and running a QuaRC code using Simulink

1. Log in the account “nobody” which is not password protected.
2. Launch MATLAB and start Simulink.
3. Create a new folder “EE351L” in “My Documents” and save all files you will create throughout the experiment in this “EE351L” folder.
4. To apply a voltage to the DC motor on SRV-02, follow the steps in subsection 4.1 of [3] with the following changes:
 - (a) Set the “Board Type” field in the HIL Initialize Block to “ni_pci_6251” instead of “q4” or “q8”.
 - (b) Use a “HIL Write Analog Block” instead of a “HIL Write Block”.

Important: To protect the equipment, do not set the amplitude and the frequency in the “Signal generator” to values higher than the ones indicated in the manuals.
5. To measure the potentiometer voltage on SRV-02, follow the steps in subsection 4.2 of [3] with the following (additional) changes:
 - (a) Use a “HIL Read Analog Timebase Block” instead of a “HIL Read Timebase Block”. Set the “Samples in Buffer” field to 1 in the “HIL Read Analog Timebase Block”.
 - (b) The potentiometer we have is model 138 instead of model 132. It has an electrical range $345^\circ \pm 4^\circ$ according to [1]. Therefore, use the scaling factor $-345/10$ instead of $-352/10$ in potentiometer calibration.
6. To measure the tachometer voltage on SRV-02, follow the steps in subsection 4.3 of [3] with the following (additional) changes:
 - (a) Set the “Analog channels” field in the “HIL Read Analog Timebase” block to [0,1,2,3] instead of [0,2]. Adding the analog channels #1 and #3 seems unnecessary since you will not use them. However, for some unknown reason, the tachometer voltage does not seem to show up at the analog channel #2 otherwise.
 - (b) The SRV-02 is in the high-gear configuration, therefore, use the corresponding scaling factor for tachometer calibration.
7. To measure the encoder signal on SRV-02, follow the steps in subsection 4.4 of [3] with the following (additional) changes:
 - (a) An encoder channel is not available in the “HIL Read Analog Timebase” block. Therefore, use an “HIL Read Encoder” block from the QuaRC Targets library under Data Acquisition → Generic → Immediate I/O.

Tip: Use Figure 4.2 as a guide to build your Simulink models for this experiment.

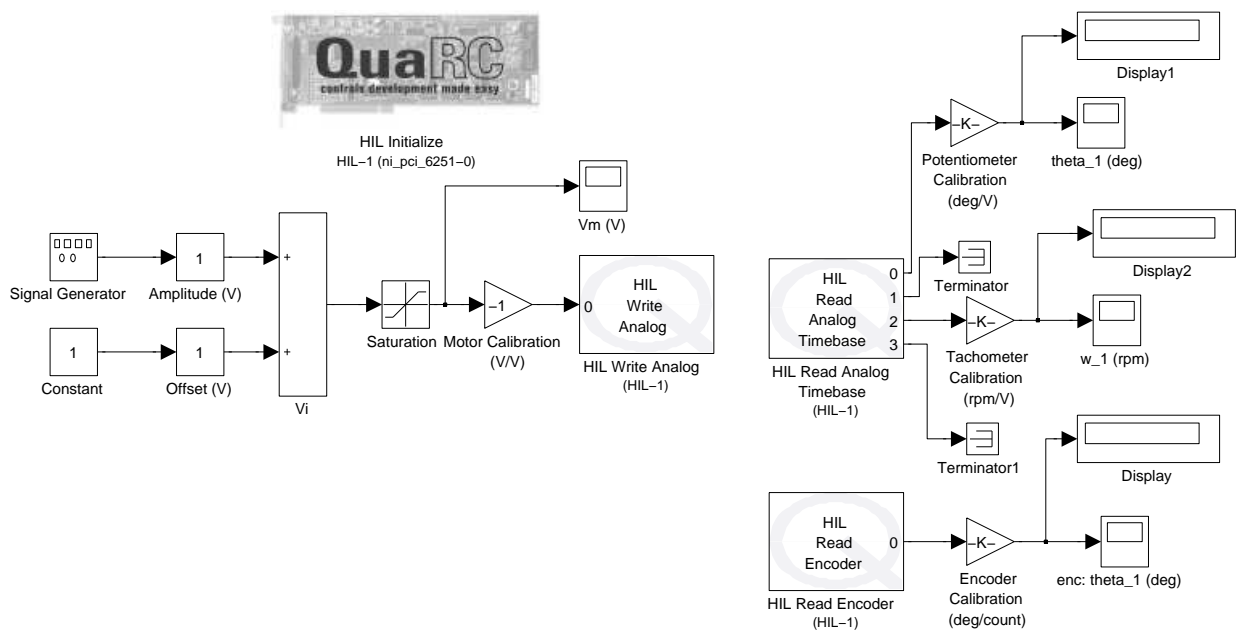


Figure 4.2: Simulink model to send voltage to the SRV-02 as well as to obtain various readings from the SRV-02.

4.3 In-Lab Assignment and Post-Lab Report

Assume that the encoder gives you perfect differential position information. You should actually convince yourself that the encoder measurements are quite accurate by rotating the gear 360° manually and comparing your visual reading with the calibrated encoder reading. On the other hand, the potentiometer voltage range is not exactly $\pm 5V$ (you can actually measure the potentiometer with a multimeter), and its electrical range is not exactly 345° .

1. Think about a way of re-calibrating the potentiometer voltage so that the position information obtained via the potentiometer tracks very closely the position information via the encoder (up to a constant, of course, since potentiometer gives absolute position information whereas the encoder gives differential position information). Before doing this assignment, you should have reviewed [1] which explains how the potentiometer and the encoder work.
2. Estimate the rate based on the encoder measurement. Compare the rate obtained from the tachometer measurement with the estimated rate based on the encoder measurement. Are they sufficiently close, or do the tachometer measurements need to be re-calibrated as well?

Tip: For this assignment, you can apply a constant voltage 1V (with a saturation ± 10 V/s) as shown in Figure 4.2. As a result, the gears will start rotating at a constant rate. Do not apply arbitrary voltages with high amplitudes or high frequencies!

Save all of your data and figures. Submit a typed and well-organized lab report to the instructor by the beginning of the next lab session. In your report, explain your reasoning and what you did clearly but concisely. Include at least one figure illustrating that the re-calibrated potentiometer voltage closely tracks the position information via the encoder. Include at least one figure comparing the rate obtained from the tachometer measurement with the estimated rate based on the encoder measurement. In both questions, you can superimpose two trajectories on top of each other and/or plot the difference, etc.

Tip: Use the “subplot” command in MATLAB to save space.

Bibliography

- [1] Quanser SRV-02 User Manual.
- [2] Quanser UPM User Manual.
- [3] Quanser Rotary Experiment #00: QuaRC Integration Student Manual.
- [4] QuaRC Data Acquisition Card Support.