Laboratory guide 1: Introduction to the DC servomotor experimental setup

Riccardo Antonello[*]　　Francesco Ticozzi[*]

March 19, 2025

1 Activity goal

The purpose of this laboratory activity is to become familiar with the DC servomotor unit used in the next laboratory activities. By the end of this activity, you should be able to implement and configure a Simulink model designed for "real-time simulation" – a simulation that runs in real time and allows direct interaction with a physical device.

2 Experimental setup

The experimental setup consists of:

1) **Quanser SRV–02 servomotor**. This is the physical device to be controlled (*plant*). It consists of a DC motor (① in Fig. 1c) with a built–in *planetary* gearbox (② in Fig. 1c) capable of driving a mechanical load attached to its output shaft (② in Fig. 1b). The mechanical load considered in this course is a simple disc inertia (i.e. *inertial load* – ① in Fig. 1e). An incremental optical encoder is directly connected to the output shaft of the SRV–02 unit for measuring the position of the mechanical load (② in Fig. 1d). A potentiometer (③ in Fig. 1d) is also connected to the shaft through a gear coupling (that includes an anti–backlash mechanism – ③ in Fig. 1b), and can be used as an alternative load position sensor. The DC motor is driven by a linear voltage driver based on a power operational amplifier (① in Fig. 1d). The simplified electrical schematic of the voltage driver is reported in Fig. 2d. It is rather immediate to deduce that the voltage driver behaves – from "Motor Input" to "Motor Current A" output – as a first order low–pass filter with DC gain

$$k_{\mathrm{drv}} \;=\; \frac{R_2}{R_1+R_2}\left(1+\frac{R_3}{R_4}\right) \;\approx\; 0.6 \tag{1}$$

and cut–off frequency

$$f_c = \frac{1}{2\pi\,(R_1//R_2)\,C_1} \;\approx\; 1.2\,\mathrm{kHz} \tag{2}$$

A *shunt resistor* is connected in series to the motor to sense the *armature current* (i.e. the current flowing through the rotor windings).

[*]Department of Information Engineering (DEI), University of Padova.
email: {antonello, ticozzi}@dei.unipd.it

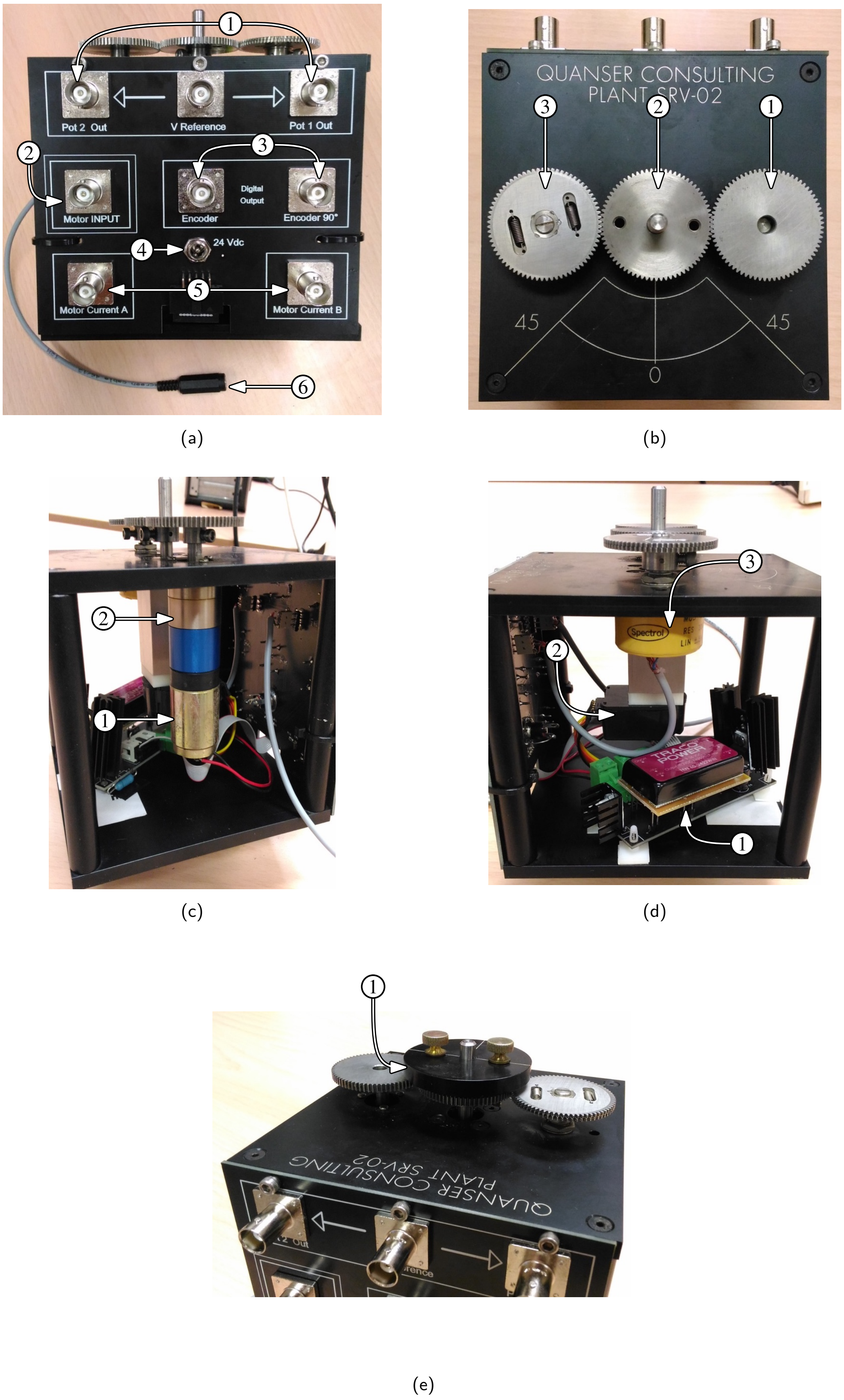
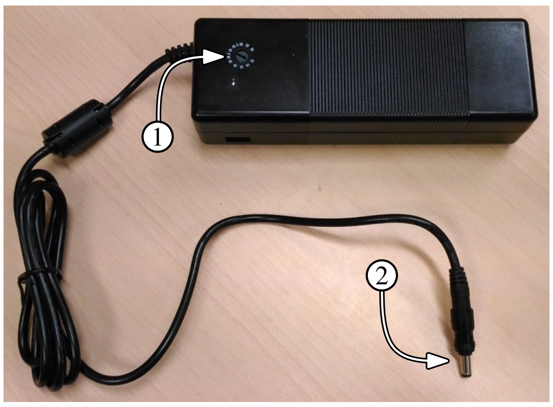
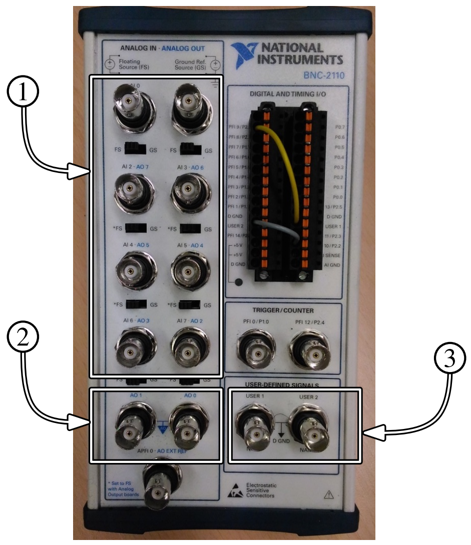


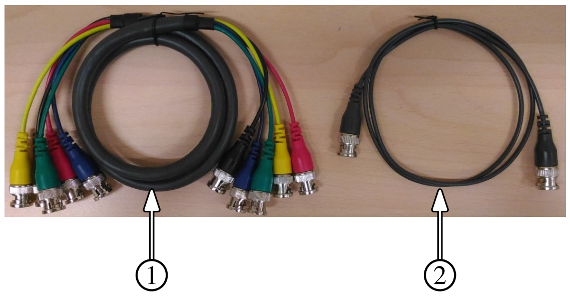
Figure 1: Experimental setup details.



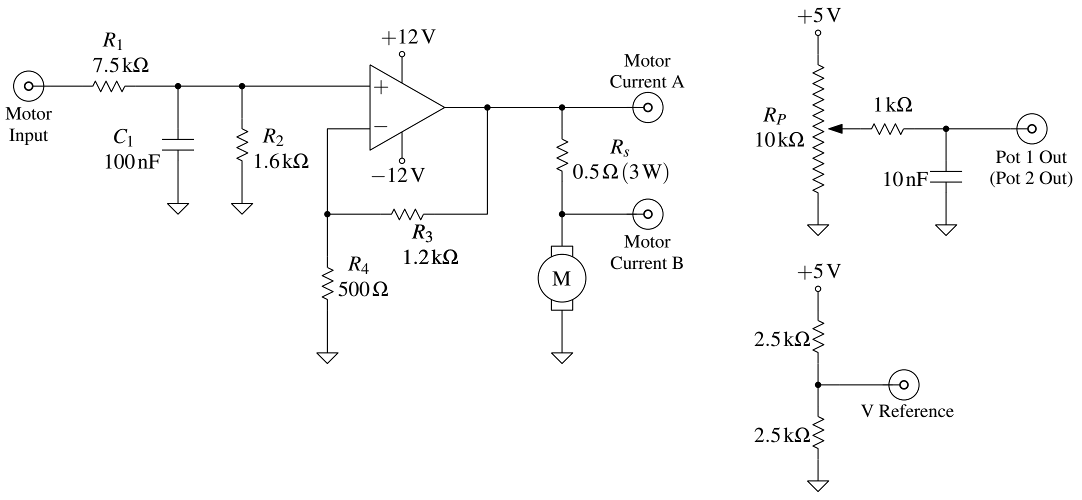
(a)



(b)



(c)



(d)

Figure 2: Experimental setup details (cont'd).

2) **National Instruments Data Acquisition Board (NI-DAQ PCI-6221 or PCIe-6321)**.

The PCI–6221 board (installed on PCFA05 $\div$ PCFA12 workstations) supports

- 16 single-ended or 8 differential analog inputs channels with 16 bit ADC resolution, maximum sample rate (per channel) of $250\,\mathrm{kS/s}$ and configurable ADC input range (available options: $\pm0.2\,\mathrm{V}$, $\pm1\,\mathrm{V}$, $\pm5\,\mathrm{V}$, $\pm10\,\mathrm{V}$).
- 2 analog output channels with 16 bit DAC resolution, fixed $\pm10\,\mathrm{V}$ output range and maximum sample rate of $833\,\mathrm{kiloS/s}$ for single–channel output mode and $740\,\mathrm{kS/s}$ for dual–channel output mode.
- 24 digital I/O channels.
- 2 general purpose 32–bit counters/timers that can be used for PWM generation, encoder counting, frequency measurement, event counting, etc.

The PCIe–6321 board (installed on PCFA01 $\div$ PCFA01 workstations) has similar specifications to PCI–6221, expect for minor differences such as

- the DAC maximum sampling rates: $900\,\mathrm{kS/s}$ for single–channel output mode and $840\,\mathrm{kS/s}$ for dual–channel output mode.
- the increased number of general purpose counters/timers: 4 instead of 2.

3) **National Instruments BNC–2110 terminal board**.

4) **BNC–terminated connection cables** (three single–channel cable + one 5–channel cable for each workbench).

5) **DC power supply** with adjustable voltage output (laptop power adapter).

6) **PC workstation** running **MATLAB/Simulink**. The **Simulink Desktop Real–Time** toolbox (SLDRT) is used to perform a “real–time simulation” that involves a direct interaction with the experimental setup.

The experimental setup parameters that are relevant for simulation and control design purposes are listed in Tab. 1. The nominal values of the parameters are deduced from accompanying data–sheets.

3 Real–time simulation

The MATLAB/Simulink environment will be used throughout the course to support all activities involved in the typical design flow of a control system. Both the implementation and experimental testing of control systems will be conducted within this environment, by resorting to the *Simulink Desktop Real–Time (SLDRT)* toolbox. SLDRT provides a real-time kernel for executing Simulink models on computers running Windows or macOS. It includes library blocks that enable Simulink models to interface with various I/O devices, supporting a wide range of I/O types, including analog and digital signals, encoder/counter inputs, PWM/frequency outputs, and communication protocols such as serial, UDP, and TCP. To run a Simulink model in real time using SLDRT, the model must first be converted into executable code. This code is then automatically loaded into memory and executed by the SLDRT kernel when the “real-time simulation begins”. MATLAB handles the entire conversion process transparently, which consists of two main stages: first, the *Simulink Coder*

DC gearmotor nominal parameters	
Motor type	brushed DC motor
Motor id	Faulhaber 2338S006S
Nominal voltage	$6\,\mathrm{V}$
Nominal output power	$3.23\,\mathrm{W}$
Motor efficiency	0.69
Armature resistance	$2.6\,\Omega$
Armature inductance	$180\,\mu\mathrm{H}$
No–load speed	$7200\,\mathrm{rpm}$
No–load current	$0.08\,\mathrm{A}$
Stall torque	$1.71\,\mathrm{N\,cm}$
Back–EMF constant	$0.804\,\mathrm{mV/rpm}$
Torque constant	$7.68 \times 10^{-3}\,\mathrm{Nm/A}$
Rotor inertia	$3.90 \times 10^{-7}\,\mathrm{kg\,m^2}$
Gearbox type	planetary gearbox
Gearbox id	Micromotor SA 23/1
Gearbox ratio	14
Gearbox efficiency	0.80
Moment of inertia of external 72–tooth gear	$1.4 \times 10^{-6}\,\mathrm{kg\,m^2}$
Moment of inertia of extra disc	$3.0 \times 10^{-5}\,\mathrm{kg\,m^2}$

Sensors data	
Sensor type	optical incremental encoder
Sensed quantity	output shaft angular position (incremental)
Sensor id	Hewlett-Packard HEDS-5540#A06
Pulses–per–rotation (ppr)[(1)]	500 (all units except "MOTORE 8" and "10") 1024 (on "MOTORE 8" and "10" units)
Sensor type	potentiometer "1"
Sensed quantity	output shaft angular position (absolute)
Sensor id	Spectrol 138-0-0-103
Resistance	$10\,\mathrm{k}\Omega$
Angle range	± 176 deg
Supply voltage	$5\,\mathrm{V}$
Sensor type	potentiometer "2"
Sensed quantity	flexible joint angular displacement (absolute)
Sensor id	Spectrol 357-0-0-103
Resistance	$10\,\mathrm{k}\Omega$
Angle range	± 170 deg
Supply voltage	$5\,\mathrm{V}$

(1) $\times 4$ times larger when adopting a quadrature encoding mode for counting the encoder pulses.

Voltage driver data	
Driver type	linear voltage driver (LM1875 power op-amp)
Driver gain	≈ 0.6
Driver cut-off frequency	$\approx 1.2\,\mathrm{kHz}$

I/O board data	
Board id	NI PCI-6221 (on PCFA05 ÷ PCFA12 workstations) NI PCIe-6321 (on PCFA01 ÷ PCFA04 workstations)
ADC range	$\pm 0.2\,\mathrm{V}$, $\pm 1\,\mathrm{V}$, $\pm 5\,\mathrm{V}$, $\pm 10\,\mathrm{V}$
ADC resolution	$16\,\mathrm{bits}$
DAC range	$\pm 10\,\mathrm{V}$
DAC resolution	$16\,\mathrm{bits}$

Table 1: Experimental setup data.

toolbox automatically translates the Simulink model into C source code; then, the generated C code is compiled using a supported C compiler.

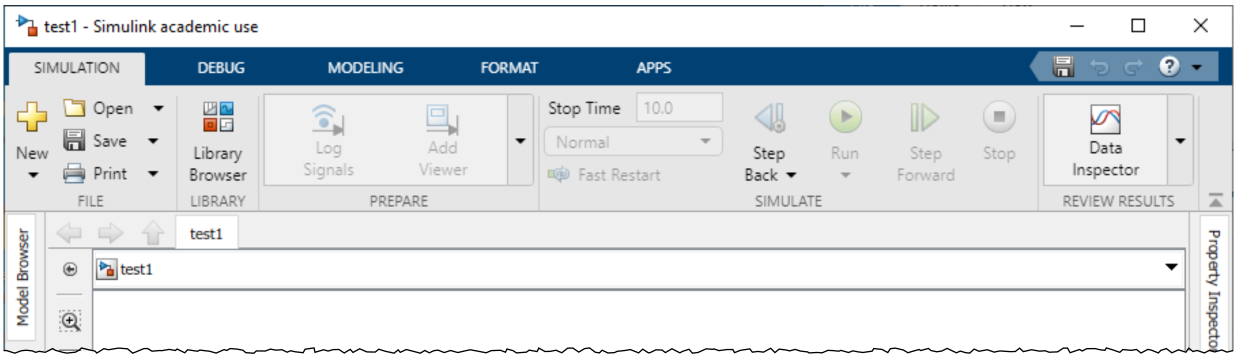
3.1 Configure a Simulink model for Simulink Desktop Real-Time

Follow these steps to set up a Simulink model for real-time execution using the Simulink Desktop Real-Time (SLDRT) toolbox:

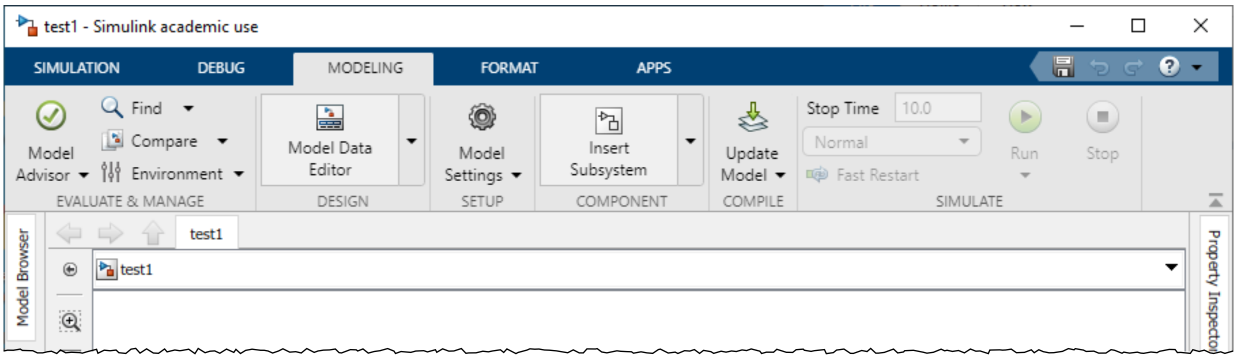
1) Set the Working Directory: change the current MATLAB working directory to your designated working folder. You can do this using the `cd` command in the MATLAB command window, or by selecting the desired directory in the *Current Directory* field on the MATLAB toolbar.

2) Create a New Simulink Model: open a new Simulink model by either selecting *New* → *Model* from the MATLAB toolbar, or by running `slLibraryBrowser` from the command window, then clicking the *New Button* in the *Simulink Library Browser* toolbar.

3) Save the Simulink Model in the Working Directory: save your model in the working directory by selecting **Save** → **Save as ...** from the **Simulation** tab on the Simulink model window. Since the Simulink Coder generates auxiliary files in the current working directory during code generation, ensure that your model file is saved in the correct directory to prevent auxiliary files from being stored in unintended locations.



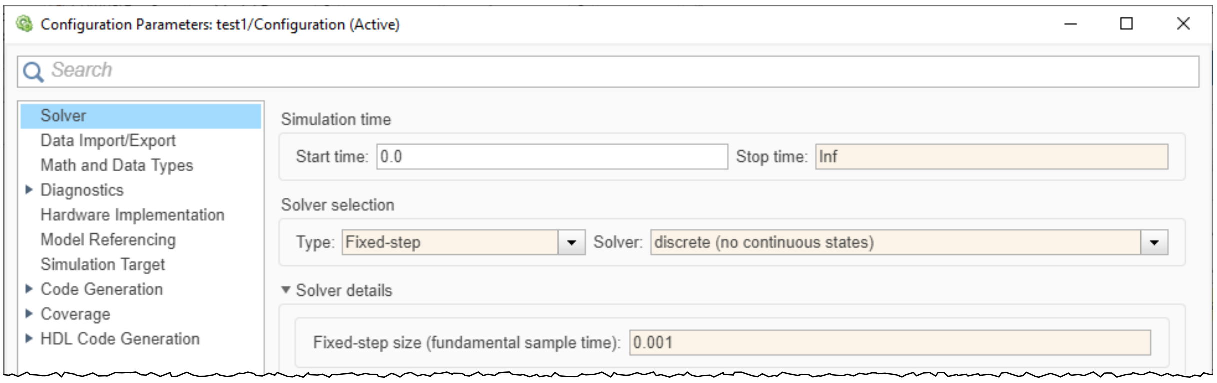
4) Set Model Parameters for Real-Time Execution with SLDRT: to configure the model for SLDRT, open the *Configuration Parameter* window by clicking **Model Settings** in the **Modeling** tab of the Simulink model window.



4-a) Click the **Solver** node. In the **Stop time** field, enter an appropriate final time. Use `Inf` if you want to run the simulation indefinitely until manually stopped.
From the **Type** list, select **Fixed-step**, as *Simulink Coder* does not support variable-step solvers.

From the **Solver** list, select an appropriate solver. If the model contains continuous-time blocks, choose a suitable ODE solver, such as `ode5 (Dormand-Prince)` If the model consists only of discrete-time blocks, select the `discrete (no continuous state)` solver.

Under **Additional options**, in the **Fixed step size (fundamental sample time)** field, enter a sample time (e.g. 0.001 for a $1\,\mathrm{ms}$ sample time). Click **Apply** to confirm the changes.


Configuration Parameters: test1/Configuration (Active)

Search

Solver
Data Import/Export
Math and Data Types
Diagnostics
Hardware Implementation
Model Referencing
Simulation Target
Code Generation
Coverage
HDL Code Generation

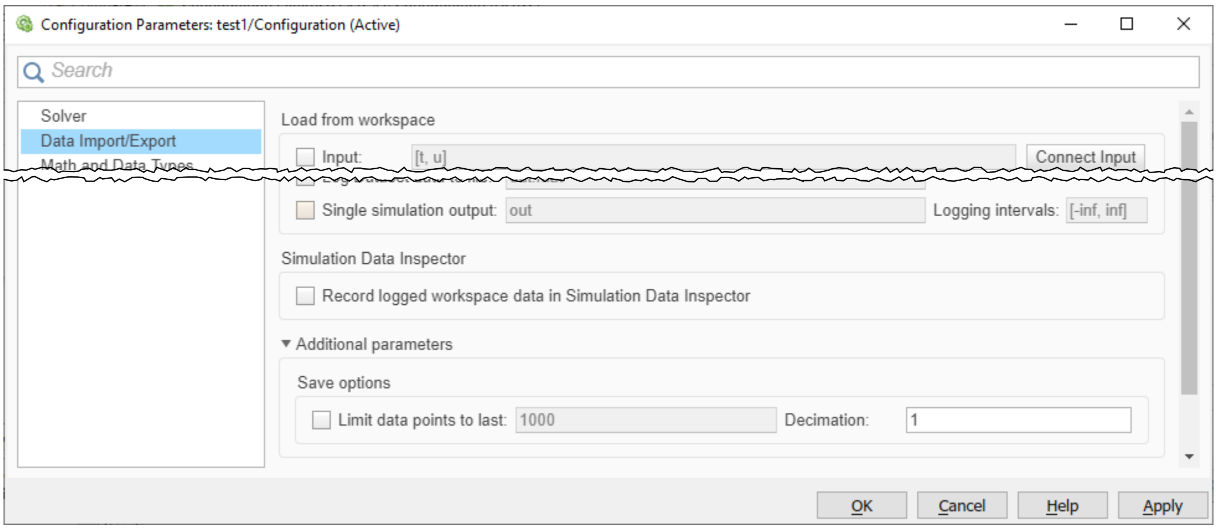
Simulation time
Start time: 0.0 Stop time: Inf

Solver selection
Type: Fixed-step Solver: discrete (no continuous states)

Solver details
Fixed-step size (fundamental sample time): 0.001

4-b) Click the **Data Import/Export** node. Uncheck **Single simulation output** to export simulation data logged by *Scope* and *To Workspace* blocks as independent variables in the MATLAB workspace. If left unchecked, a single container variable of type `Simulink.SimulationOutput` will be created.

Under **Additional parameters**, uncheck **Limit data points to last** to ensure all generated simulation data is saved. Click **Apply** to confirm changes.


Configuration Parameters: test1/Configuration (Active)

Search

Solver
Data Import/Export
Math and Data Types

Load from workspace
☐ Input: [t, u] Connect Input

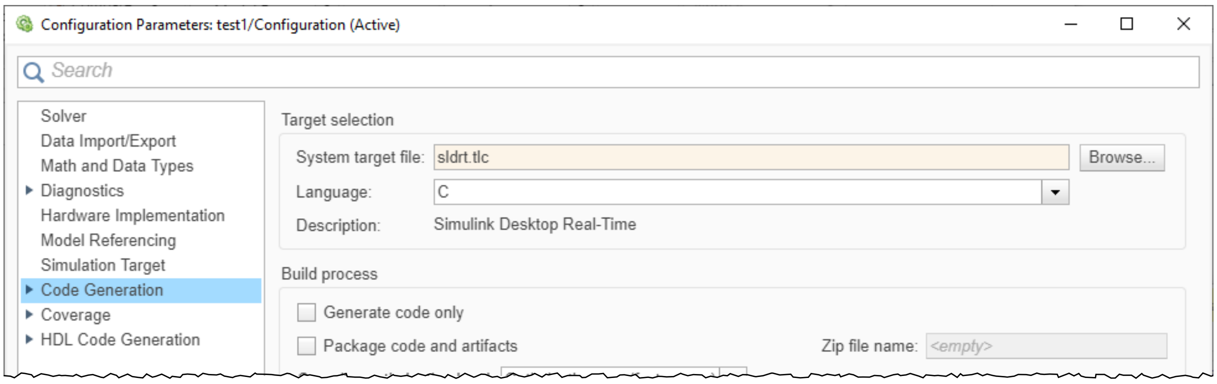
☐ Single simulation output: out Logging intervals: [-inf, inf]

Simulation Data Inspector
☐ Record logged workspace data in Simulation Data Inspector

Additional parameters
Save options
☐ Limit data points to last: 1000 Decimation: 1

OK Cancel Help Apply

4-c) Click the **Code generation** node. In the **Target selection** section, click **Browse** in the **System target file** list.


Configuration Parameters: test1/Configuration (Active)

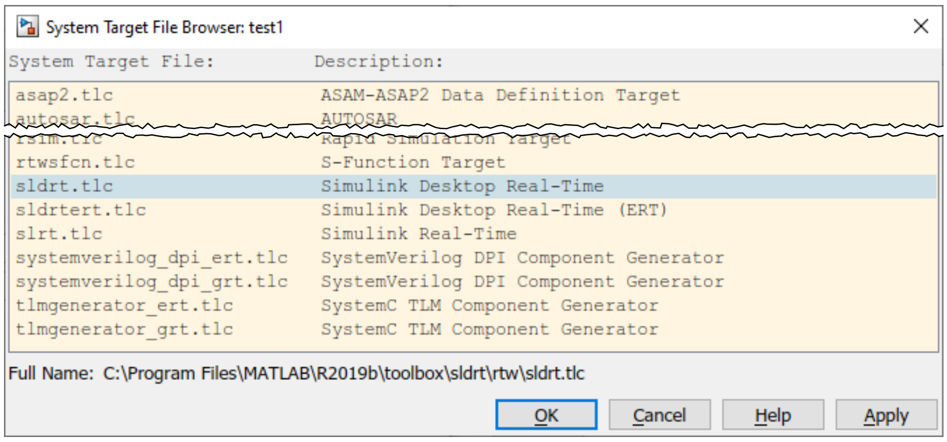
Search

Solver
Data Import/Export
Math and Data Types
Diagnostics
Hardware Implementation
Model Referencing
Simulation Target
Code Generation
Coverage
HDL Code Generation

Target selection
System target file: sldrt.tlc Browse...
Language: C
Description: Simulink Desktop Real-Time

Build process
☐ Generate code only
☐ Package code and artifacts Zip file name: <empty>

In the **System Target Browser** select `sldrt.tlc` (i.e. the *system target file* for building SLDRT applications), and click **OK**.

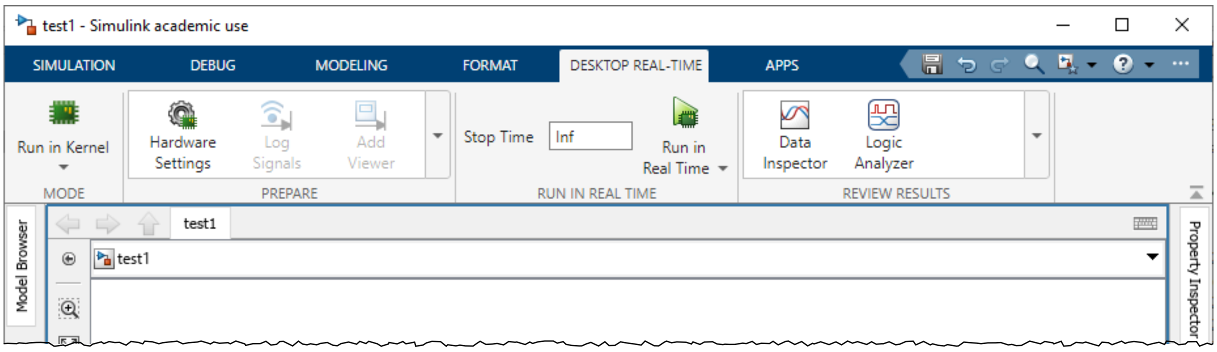

System Target File Browser: test1

```
System Target File:        Description:
asap2.tlc                  ASAM-ASAP2 Data Definition Target
autosar.tlc                AUTOSAR
rsim.tlc                   Rapid Simulation Target
rtwsfcn.tlc                S-Function Target
sldrt.tlc                  Simulink Desktop Real-Time
sldrtert.tlc               Simulink Desktop Real-Time (ERT)
slrt.tlc                   Simulink Real-Time
systemverilog_dpi_ert.tlc  SystemVerilog DPI Component Generator
systemverilog_dpi_grt.tlc  SystemVerilog DPI Component Generator
tlmgenerator_ert.tlc       SystemC TLM Component Generator
tlmgenerator_grt.tlc       SystemC TLM Component Generator
```

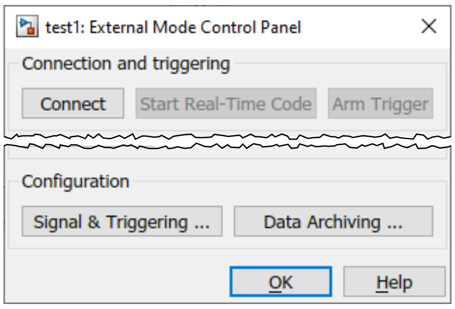
Full Name: C:\Program Files\MATLAB\R2019b\toolbox\sldrt\rtw\sldrt.tlc

4-d) Click **OK** in the *Configuration Parameter* window to close it and apply all changes.

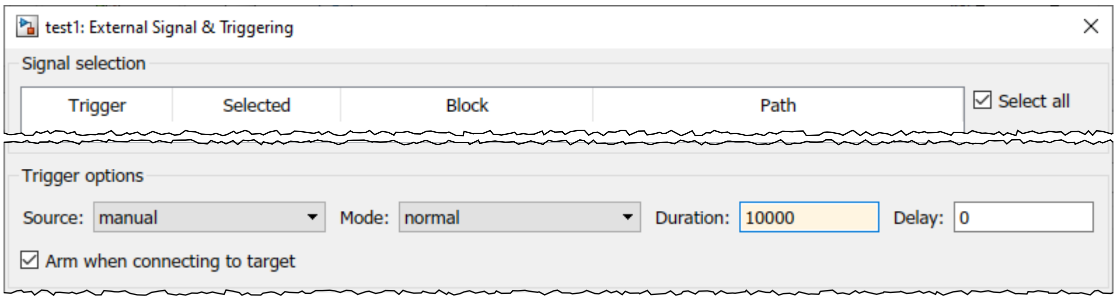
4-e) In the Simulink model window, select **Desktop Real-Time** tab. In the **Mode** section, select **Run in Kernel** to enable real-time execution of the model on the desktop computer.



5) <u>Configure Buffer Size for Data Exchange with Real-Time Application</u>: in the **Prepare** section, click on **Control Panel** to open the *External Mode Control Panel*.



- In the *External Mode Control Panel*, click **Signal & Triggering** to open the *External Signal & Triggering* window.
- In the *Trigger options* section, under the **Duration** field, enter the length (number of sample points) for the buffer used to exchange data between Simulink and the real-time application. For example, if the sample time is $0.001\,\mathrm{s}$, setting *Duration* to 10000 samples allows storing $10\,\mathrm{s}$ of simulation data (from $0\,\mathrm{s}$ to $9.999\,\mathrm{s}$).


test1: External Signal & Triggering

Signal selection

Trigger	Selected	Block	Path

☑ Select all

Trigger options

Source: manual Mode: normal Duration: 10000 Delay: 0

☑ Arm when connecting to target

- Click **OK** to save changes and close the *External Signal & Triggering* window. In the *External Mode Control Panel*, click **OK** to close the window.

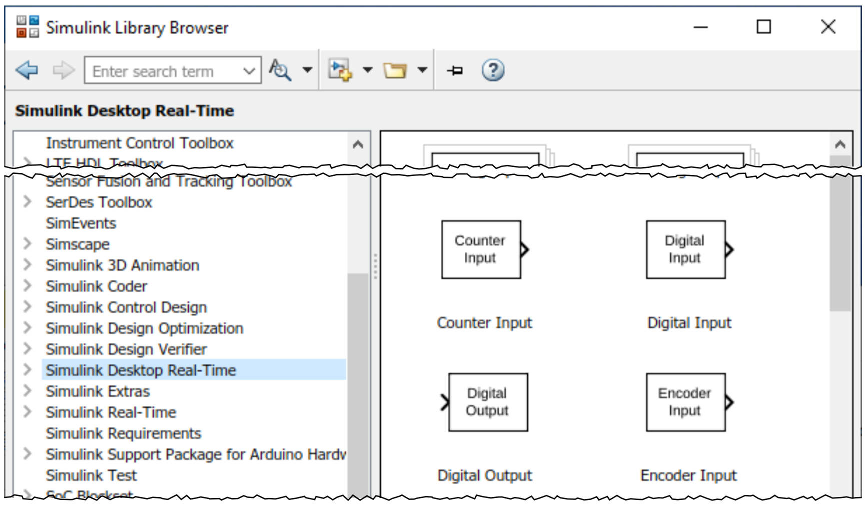
Note: when the buffer is full, it is fully erased before saving new data. This means that the buffer always holds only the last "Duration" samples produced by the simulation.

6) Save the Model with the New Configuration: on the **Simulation** tab of the Simulink model window, click **Save** to save the model with the new configuration settings.

3.2 Interface the Simulink model with the hardware

The SLDRT blockset enables the connection of a Simulink model to various I/O device, including the NI-DAQ board installed on the laboratory PC. The following steps show how to use the SLDRT blocks to interface a Simulink model with the Quanser SRV-02 servomotor:

1) Adding SLDRT Blocks to the Simulink Model: open the **Simulink Library Browser** and navigate to **Simulink Desktop Real–Time** blockset.


Simulink Library Browser

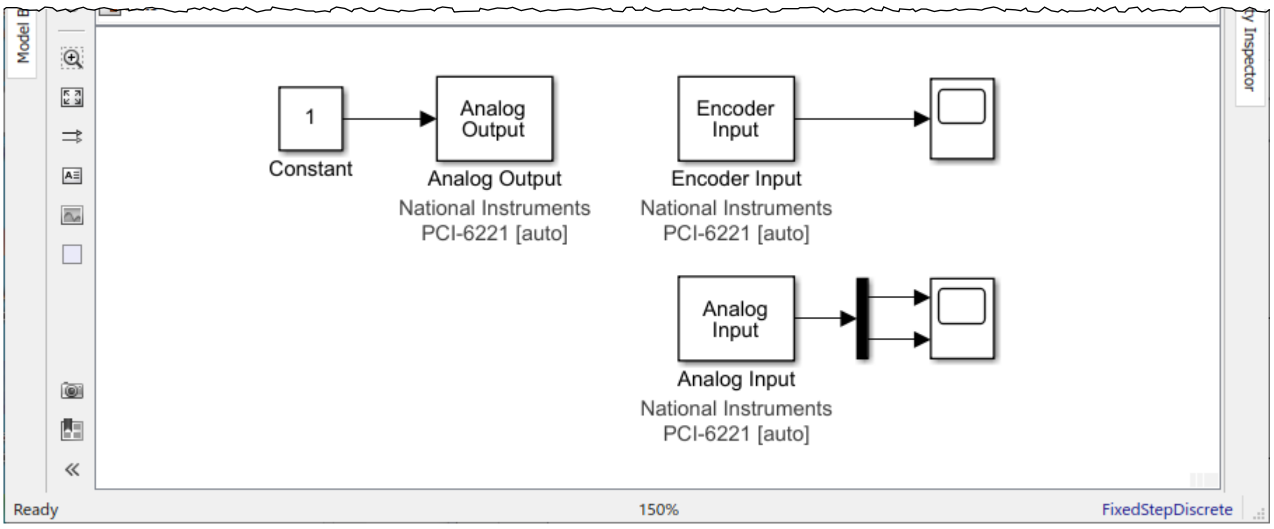
Enter search term

Simulink Desktop Real-Time

Instrument Control Toolbox
LTE HDL Toolbox
Sensor Fusion and Tracking Toolbox
SerDes Toolbox
SimEvents
Simscape
Simulink 3D Animation
Simulink Coder
Simulink Control Design
Simulink Design Optimization
Simulink Design Verifier
Simulink Desktop Real-Time
Simulink Extras
Simulink Real-Time
Simulink Requirements
Simulink Support Package for Arduino Hardv
Simulink Test
SoC Blockset

Drag and drop the following blocks into your Simulink model window:

- **Analog Input**: reads values from a specified input channel (or list of channels) of the Analog-to-Digital Converter (ADC) in the selected I/O board.
- **Analog Output**: writes values to a specified output channel (or list of channels) of the Digital-to-Analog Converter (DAC) in the selected I/O board.
- **Encoder Input**: reads data from a specified Encoder Pulse Counter integrated in the selected I/O board.



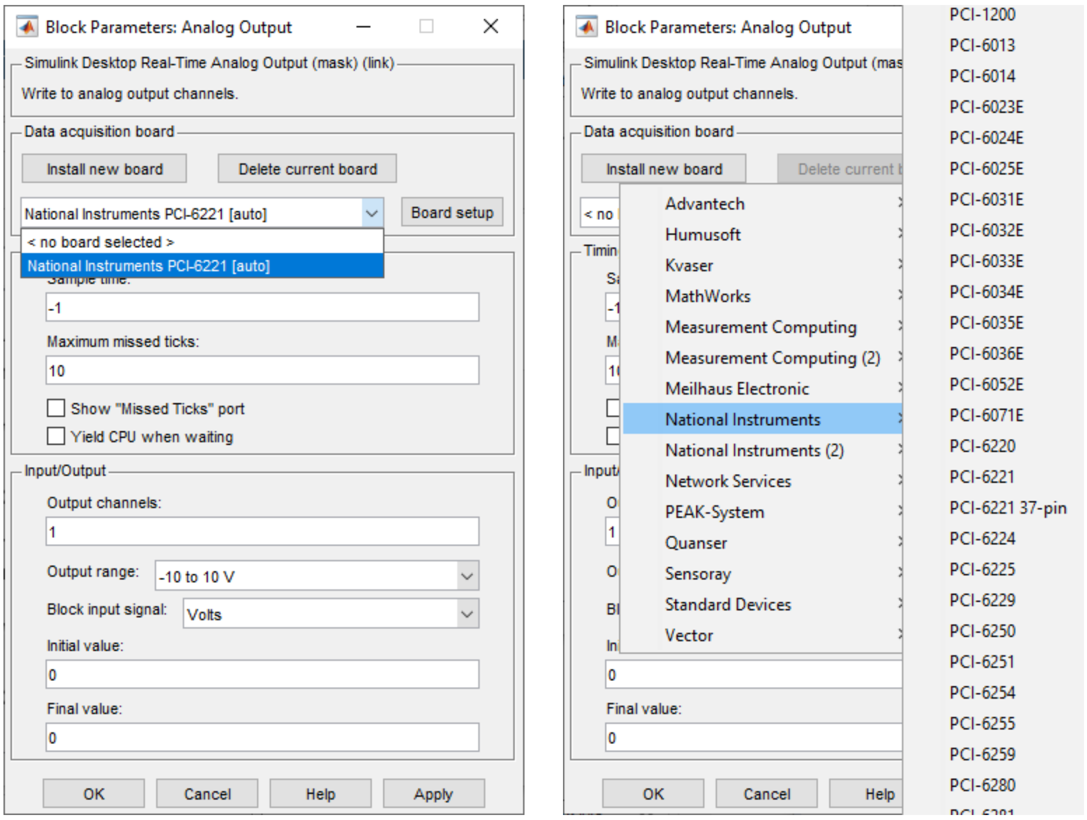
In this example:

- the **Analog Input** block acquires the "Motor Current A" and "Motor Current B" signals from the Quanser SRV-02 unit. These signals can be separated at the block output using a *Demux* block, and then displayed on a *Scope*.
- the **Analog Output** block sends a specified voltage to the Motor Input of the SRV-02 unit. Use a *Constant* block to define the voltage value.
- the **Encoder Input** block retrieves the pulse count from the incremental optical encoder of the SRV-02 unit. A *Scope* can be used to visualize the encoder output.

2) Configuring the Analog Output Block: double-click on the *Analog Output* block to open the *Block Parameters: Analog Output* dialog window.

2-a) In the *Data acquisition board* section, specify the I/O board associated with the block (either `National Instruments PCI-6221` or `PCIe-6321`, depending on your workbench):

- if the board is already installed and configured, select it in the drop–down list accessible by clicking the "▼" button, and click the **Board setup** button to check settings.
- if the board is not installed, click **Install new board** and choose the appropriate option.



Block Parameters: Analog Output

Simulink Desktop Real-Time Analog Output (mask) (link)

Write to analog output channels.

Data acquisition board

National Instruments PCI-6221 [auto]

Sample time:

-1

Maximum missed ticks:

10

☐ Show "Missed Ticks" port

☐ Yield CPU when waiting

Input/Output

Output channels:

1

Output range: -10 to 10 V

Block input signal: Volts

Initial value:

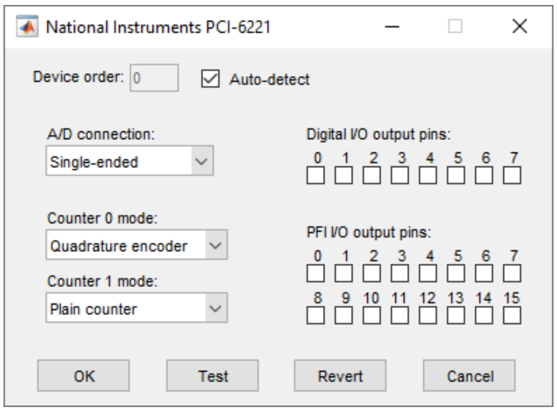
0

Final value:

0

2-b) A new window will open for board settings. In the **Counter 0 mode** drop–down list, select the `Quadrature encoder` to count both the rising and falling edges of the two encoder outputs (Note: this setting is only relevant for the Encoder Input block configured later).

Click **Test** to verify installation. If successful, *Board Test OK* will appear. Click **OK** to confirm.



National Instruments PCI-6221

Device order: 0 ☑ Auto-detect

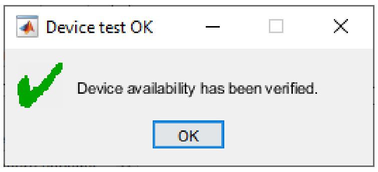
A/D connection: Single-ended

Counter 0 mode: Quadrature encoder

Counter 1 mode: Plain counter

Digital I/O output pins: 0 1 2 3 4 5 6 7

PFI I/O output pins: 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15



Device test OK

Device availability has been verified.

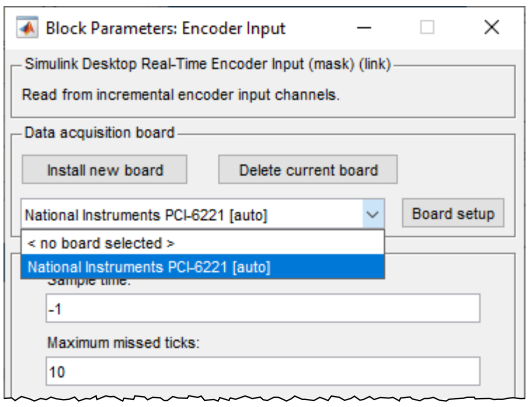
2-c) In the *Timing* section, set **Sample Time** to the desired value in seconds, or leave "-1" to inherit it from the model configuration parameters.

2-d) In the *Input/Output* section:

- specify the DAC channel by entering "1" in the **Output channels** field. This is the DAC channel physically connected to the "Motor Input" of the SRV–02 unit via the "AO 0" connector of the BNC–2110 terminal board.
- set both the **Initial value** and **Final value** to zero to ensure the motor is turned off at the start and end of the simulation.

2-e) Click the **OK** button to confirm and close the *Block Parameters: Analog Output* dialog window.

3) Configuring the Encoder Input Block: double-click on the *Encoder Input* block to open the *Block Parameters: Encoder Input* dialog window.


Block Parameters: Encoder Input

Simulink Desktop Real-Time Encoder Input (mask) (link)

Read from incremental encoder input channels.

Data acquisition board

Install new board | Delete current board

National Instruments PCI-6221 [auto] | Board setup

< no board selected >

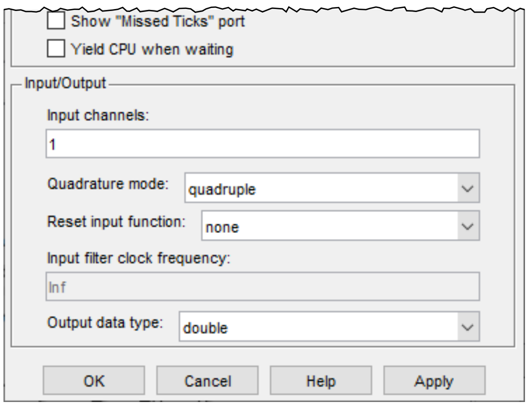
National Instruments PCI-6221 [auto]

Sample time:

-1

Maximum missed ticks:

10


☐ Show "Missed Ticks" port

☐ Yield CPU when waiting

Input/Output

Input channels:

1

Quadrature mode: quadruple

Reset input function: none

Input filter clock frequency:

Inf

Output data type: double

OK | Cancel | Help | Apply

3-a) In the *Data acquisition board* section, open the drop–down list (click on the "▼" button) and select the previously installed I/O board (either `National Instrument PCI-6221` or `National Instrument PCIe-6321`, depending on your workbench).

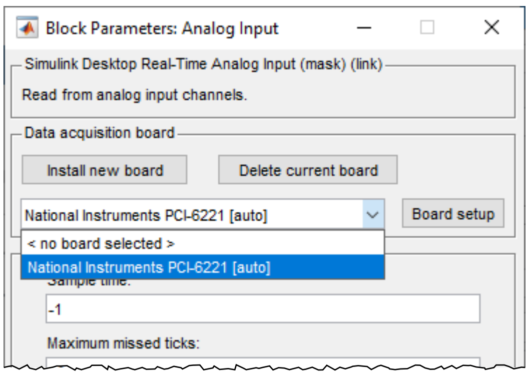
3-b) In the *Timing* section, set the **Sample Time** to the desired value, or leave "-1" to inherit from the model configuration parameters.

3-c) In the *Input/Output* section:

- Enter "1" in the **Input channels** field to use the counter 0, which is configured for counting using a quadrature encoding mode. The counter is connected to the "Encoder" and "Encoder 90°" outputs of the SRV–02 unit via the "USER 1" and "USER 2" terminals of the BNC–2110 terminal board.
- In the **Quadrature mode** drop–down list, select `quadruple` to enable counting with a quadruple encoding mode.
- In the **Reset input function** drop–down list, select `none` to reset the counter only at the beginning of the simulation. This means that the encoder initial position is assumed as the zero position reference.

3-d) Click **OK** to confirm and close the *Block Parameters: Encoder Input* dialog window.

4) Configuring the Analog Input Block: double-click on the *Analog Input* block to open the *Block Parameters: Analog Input* dialog window.


Block Parameters: Analog Input

Simulink Desktop Real-Time Analog Input (mask) (link)

Read from analog input channels.

Data acquisition board

Install new board | Delete current board

National Instruments PCI-6221 [auto] | Board setup

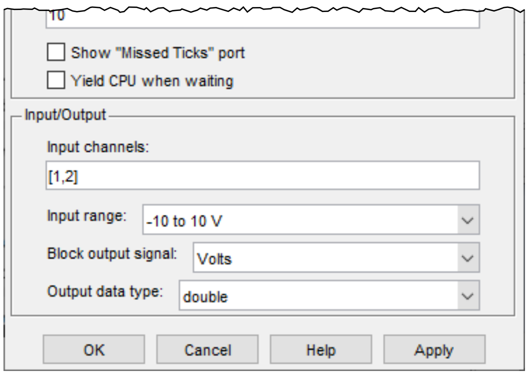
< no board selected >

National Instruments PCI-6221 [auto]

Sample time:

-1

Maximum missed ticks:


10

☐ Show "Missed Ticks" port

☐ Yield CPU when waiting

Input/Output

Input channels:

[1,2]

Input range: -10 to 10 V

Block output signal: Volts

Output data type: double

OK | Cancel | Help | Apply

4-a) In the *Data acquisition board* section, open the drop–down list (click on the "▼" button) and select the previously installed I/O board (either `National Instrument PCI-6221` or `National Instrument PCIe-6321`, depending on your workbench).

4-b) In the *Timing* section, set the **Sample Time** to the desired value, or leave "-1" to inherit from the model configuration parameters.

4-c) In the *Input/Output* section:

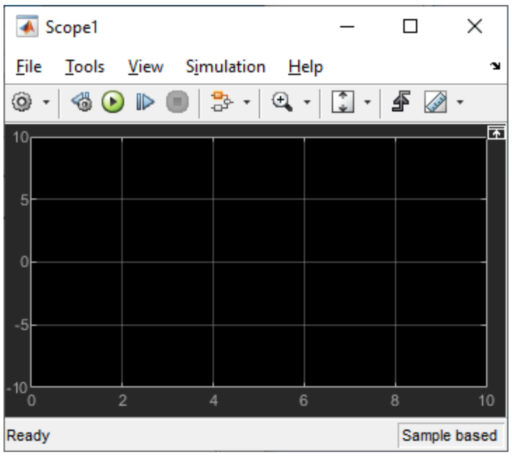
- Enter "`[1,2]`" in the **Input channels** field to use the first two ADC input channels, which are connected to the "Motor Current A" and "B" outputs of the SRV–02 unit via the "AI 0" and "AI 1" terminals of the BNC–2110 terminal board.
- In the *Input range* drop–down list, select the "`-10 to 10 V`" option to match with the $\pm 6\,\mathrm{V}$ armature voltage range.
- Click **OK** to confirm and close the *Block Parameters: Analog Input* dialog window.

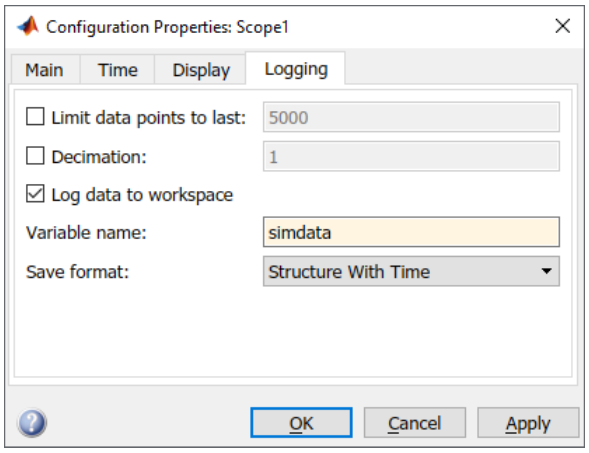
5) Save the Simulink model before proceeding. Note: it is good practice to save regularly to prevent data loss in case of an accidental MATLAB crash.

3.3 Export simulation results to MATLAB workspace

To export simulation results to the MATLAB workspace, you can use either the *To Workspace* block (found under *Simulink* $\to$ *Sinks* in the Simulink Browser), or the *Logging* feature of the *Scope* block. The latter option allows you to log data directly from a Scope block and can be enabled as follows:

1) Double-click on the *Scope* block, and click the **Configuration Properties** button in the Scope toolbar to open the *Configuration Properties* window.




Configuration Properties: Scope1

Main | Time | Display | Logging

☐ Limit data points to last: 5000

☐ Decimation: 1

☑ Log data to workspace

Variable name: simdata

Save format: Structure With Time

OK | Cancel | Apply

2) Navigate to the **History** tab. Uncheck the **Limit data points to last** checkbox.

Check the **Save data to workspace** checkbox. In the **Variable name** field, enter a name for the variable that will store the logged Scope data in the MATLAB workspace. The variable will be created automatically at the end of the simulation.

3) In the **Format** drop–down list, select the appropriate format.

4) The *Structure With Time* format is recommended, as it stores both time and signal values in a single data structure. The time and data values can be retrieved by referencing the appropriate fields of the data structure: for example, if `simdata` is the name of the variable, use the following MATLAB code to access the time and data fields:

- `t = simres.time` to retrieve the time values.

- `y = simres.signals(n).data(:,m)` to retrieve the m^{th} signal in the n^{th} set of scope axes. If the Scope block contains only a single signal and a single set of axes, you can use the shorthand: `simres.signals.data`.

Note: do not select the *Dataset* format, as it is not supported by the SLDRT.

3.4 Connecting the SRV-02 Unit to the BNC-2110 Terminal Board

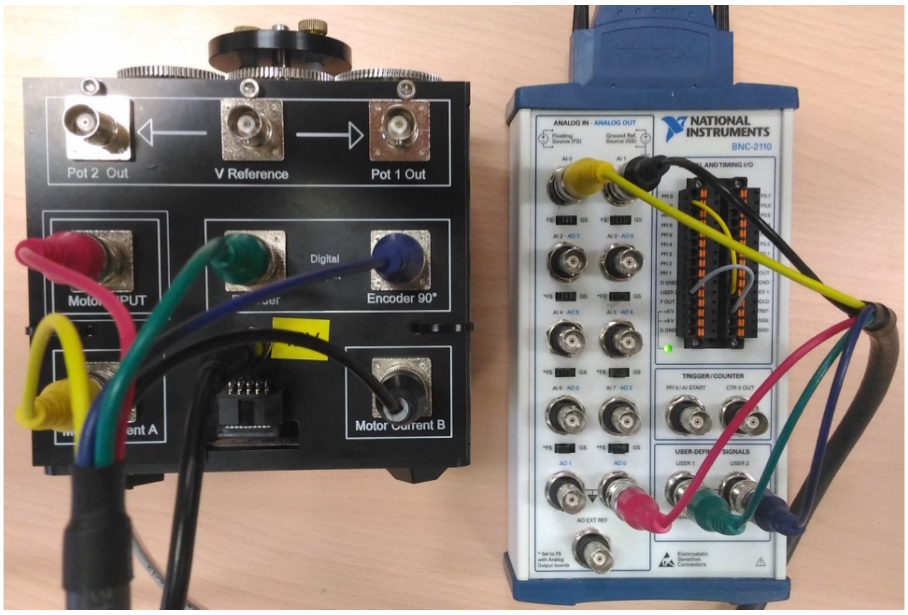
Use the BNC–terminated cables to connect the SRV–02 unit with the BNC–2110 terminal board as follows:

1) Connect the "AO 0" analog output channel of the BNC–2110 terminal board to the "Motor Input" of the SRV–02 unit. This connection allows the Simulink model to control the motor by sending voltage signals.

2) Connect the "Motor Current A" and "Motor Current B" outputs of the SRV–02 unit to the "AI 0" and "AI 1" analog input channels of the BNC–2110 terminal board. These connections allow real-time monitoring of the armature current.

3) Connect the "Encoder" and "Encoder 90°" outputs of the SRV–02 unit to the "USER 1" and "USER 2" inputs of the BNC–2110 terminal board. These signals provide the position feedback.

A direct connection is internally established between the two user–defined inputs "USER 1" and "USER 2" and the two (quadrature) inputs of counter 0 through the spring terminal block located in the *Digital and timing I/O* section of the BNC–2110 terminal board.



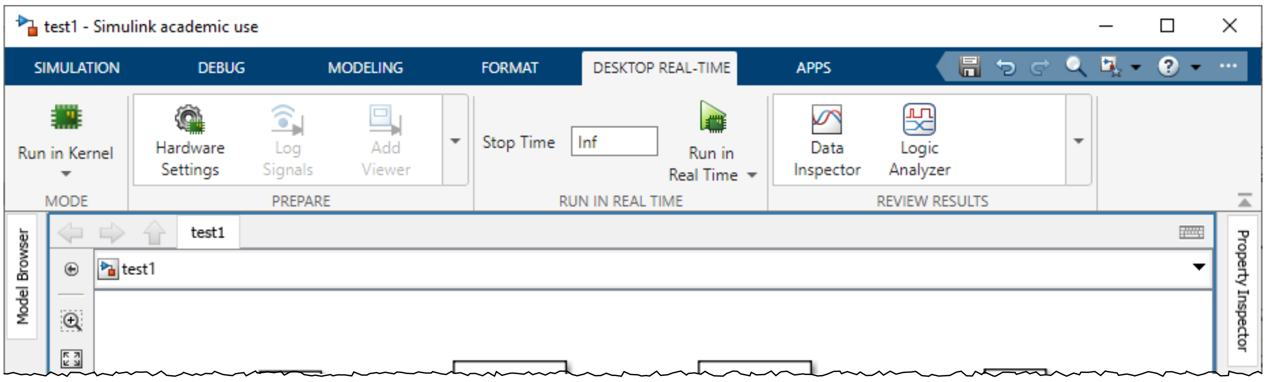
4) ⚠ **IMPORTANT: Before proceeding, verify that the regulated output voltage of the power supply adapter is set to** $12\,\mathrm{V}$. Connect the power supply to the SRV–02 unit by using the plug–in connector.

⚠ **PRECAUTION**: when the SRV–02 is turned on, **avoid accidental short-circuits** by ensuring that the voltage driver terminals do not come into contact with conductive materials (e.g., rings, metal clips, etc.).

3.5 Run the Simulink model in real–time

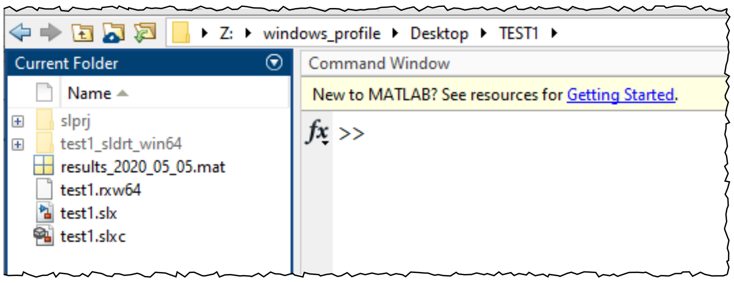
To run the Simulink model in real–time (“real–time simulation”), proceed as follows:

1) In the **Simulink Desktop Real-Time** tab of the Simulink model window, click the **Run in Real-Time** button.



This initiates the following sequence of operations:

- *Build*: the Simulink Coder is invoked to convert the Simulink model into a set of C-code source files, stored in the current working directory. Then, a C compiler compiles and links the source files to create the real-time application.

  You can manually trigger the Build process by selecting **Run in Real Time** → **Build** from the **Desktop Real-Time** tab.

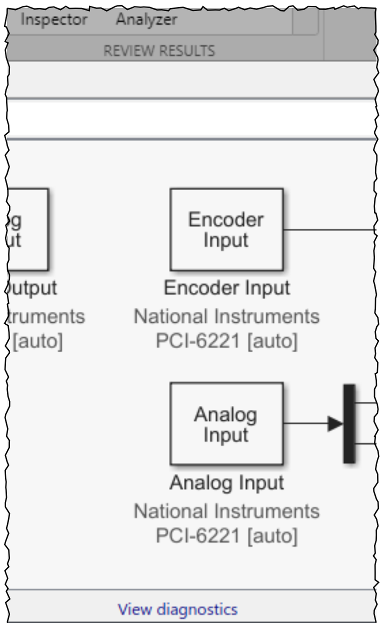
- *Connect*: the Simulink model is connected with the SLDRT kernel, enabling data exchange for commands, parameters and data logging.

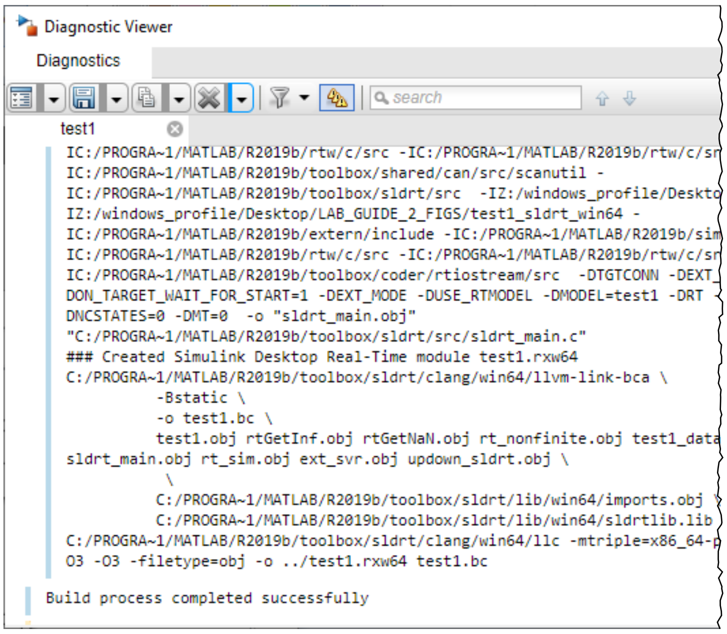
  You can manually trigger the Connect operation by selecting **Run in Real Time** → **Connect** from the **Desktop Real-Time** tab.

- *Start*: the real–time application is loaded into memory and executed by the SLDRT kernel.

  You can manually start the simulation by selecting **Run in Real Time** → **Start** on the **Desktop Real-Time** tab.

Note: you can monitor the build process in the *Diagnostic Viewer* window, accessible via the *View diagnostic* link at the bottom of the Simulink model window.





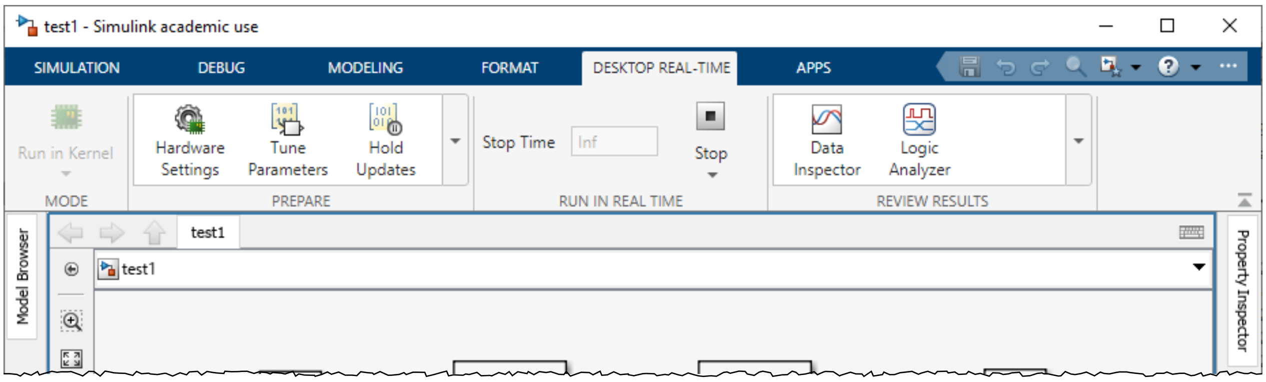
```
IC:/PROGRA~1/MATLAB/R2019b/rtw/c/src -IC:/PROGRA~1/MATLAB/R2019b/rtw/c/sr
IC:/PROGRA~1/MATLAB/R2019b/toolbox/shared/can/src/scanutil -
IC:/PROGRA~1/MATLAB/R2019b/toolbox/sldrt/src  -IZ:/windows_profile/Deskto
IZ:/windows_profile/Desktop/LAB_GUIDE_2_FIGS/test1_sldrt_win64 -
IC:/PROGRA~1/MATLAB/R2019b/extern/include -IC:/PROGRA~1/MATLAB/R2019b/sim
IC:/PROGRA~1/MATLAB/R2019b/rtw/c/src -IC:/PROGRA~1/MATLAB/R2019b/rtw/c/sr
IC:/PROGRA~1/MATLAB/R2019b/toolbox/coder/rtiostream/src  -DTGTCONN -DEXT_
DON_TARGET_WAIT_FOR_START=1 -DEXT_MODE -DUSE_RTMODEL -DMODEL=test1 -DRT -
DNCSTATES=0 -DMT=0  -o "sldrt_main.obj"
"C:/PROGRA~1/MATLAB/R2019b/toolbox/sldrt/src/sldrt_main.c"
### Created Simulink Desktop Real-Time module test1.rxw64
C:/PROGRA~1/MATLAB/R2019b/toolbox/sldrt/clang/win64/llvm-link-bca \
         -Bstatic \
         -o test1.bc \
         test1.obj rtGetInf.obj rtGetNaN.obj rt_nonfinite.obj test1_data
sldrt_main.obj rt_sim.obj ext_svr.obj updown_sldrt.obj \
          \
         C:/PROGRA~1/MATLAB/R2019b/toolbox/sldrt/lib/win64/imports.obj \
         C:/PROGRA~1/MATLAB/R2019b/toolbox/sldrt/lib/win64/sldrtlib.lib
C:/PROGRA~1/MATLAB/R2019b/toolbox/sldrt/clang/win64/llc -mtriple=x86_64-p
O3 -O3 -filetype=obj -o ../test1.rxw64 test1.bc

Build process completed successfully
```

2) While the simulation is running, you can modify the DC motor armature voltage in real–time by changing the value of the Constant block connected to the Analog Output block.

Due to static friction, the motor shaft may not rotate at low armature voltage levels. Gradually increase the voltage until the shaft begins to spin. Use the Scope to monitor the encoder pulse count and the voltage drop across the shunt resistor.

3) To stop the simulation, click the **Stop** button in the Simulink model toolbar.


test1 - Simulink academic use
SIMULATION
DEBUG
MODELING
FORMAT
DESKTOP REAL-TIME
APPS
Run in Kernel
MODE
Hardware Settings
Tune Parameters
Hold Updates
PREPARE
Stop Time
Inf
Stop
RUN IN REAL TIME
Data Inspector
Logic Analyzer
REVIEW RESULTS
Model Browser
test1
test1
Property Inspector