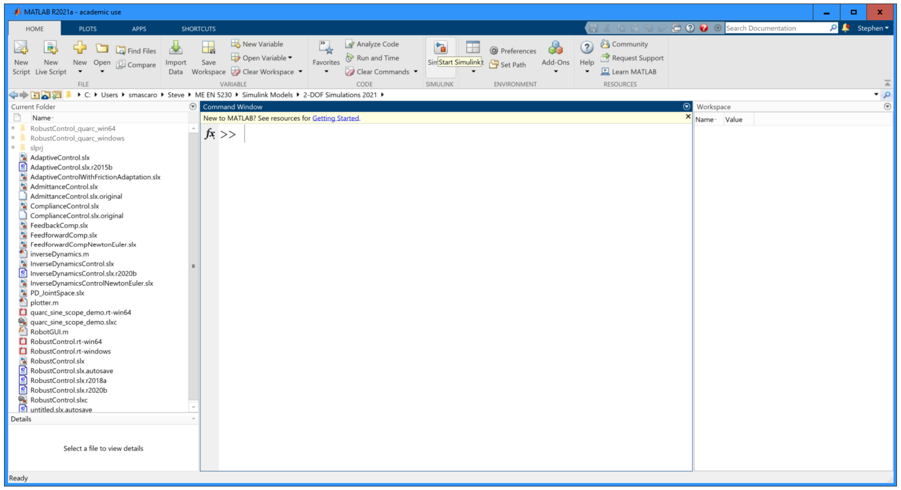
MATLAB/Simulink/QUARC Real-Time Windows Control Tutorial

(tested on Windows 10/MATLAB R2021b/QUARC 2021 SP1)

ME EN 5230/6230, CS 6330, ECE 6651 – Intro to Robot Control –Spring 2023

Following are basic instructions for creating a Windows-based Real-Time Controller using MATLAB/Simulink/QUARC. This method allows you create a controller in Simulink (using block diagram techniques) and then compile it to a Windows executable app that can run in real-time.

1. You can open the Simulink by typing "simulink" at the MATLAB command prompt or clicking on the Simulink icon on the Home menu bar.



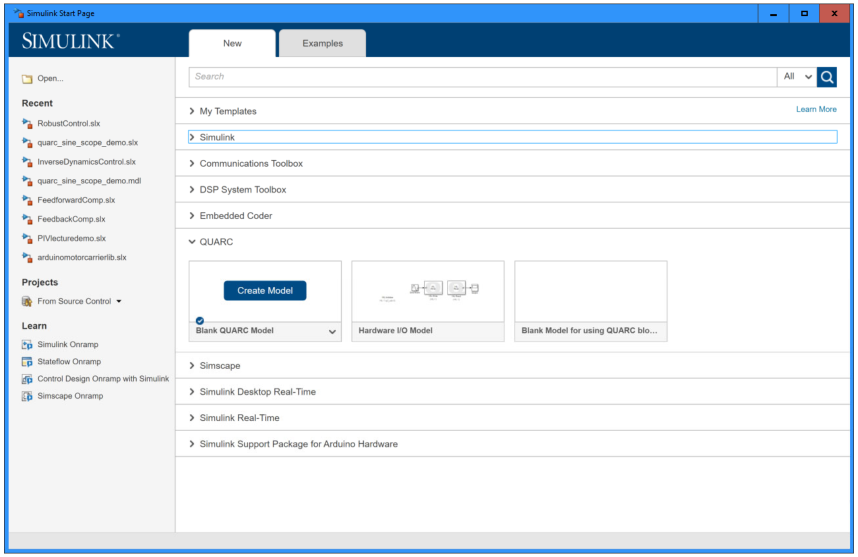
New to MATLAB? See resources for Getting Started.

```
fx >>
```

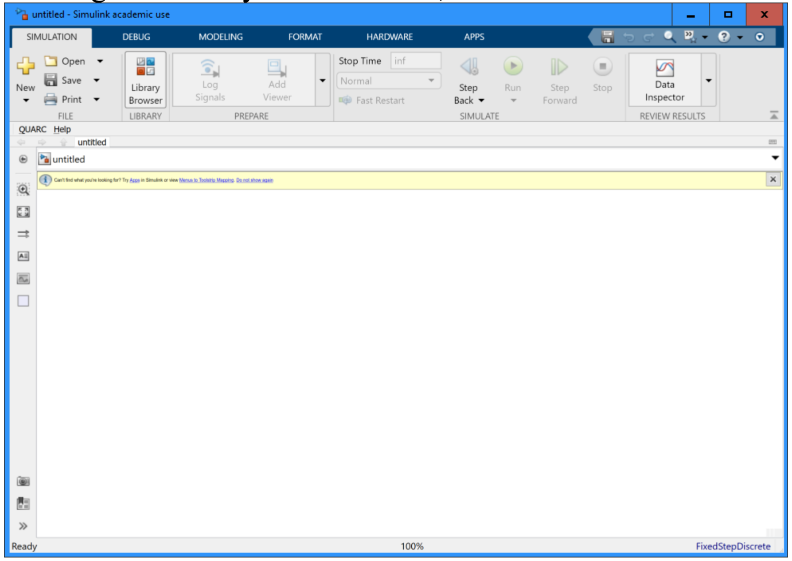
RobustControl_quarc_win64
RobustControl_quarc_windows
slprj
AdaptiveControl.slx
AdaptiveControl.slx.r2015b
AdaptiveControlWithFrictionAdaptation.slx
AdmittanceControl.slx
AdmittanceControl.slx.original
ComplianceControl.slx
ComplianceControl.slx.original
FeedbackComp.slx
FeedforwardComp.slx
FeedforwardCompNewtonEuler.slx
inverseDynamics.m
InverseDynamicsControl.slx
InverseDynamicsControl.slx.r2020b
InverseDynamicsControlNewtonEuler.slx
PD_JointSpace.slx
plotter.m
quarc_sine_scope_demo.rt-win64
quarc_sine_scope_demo.slxc
RobotGUI.m
RobustControl.rt-win64
RobustControl.rt-windows
RobustControl.slx
RobustControl.slx.autosave
RobustControl.slx.r2018a
RobustControl.slx.r2020b
RobustControl.slxc
untitled.slx.autosave

Select a file to view details

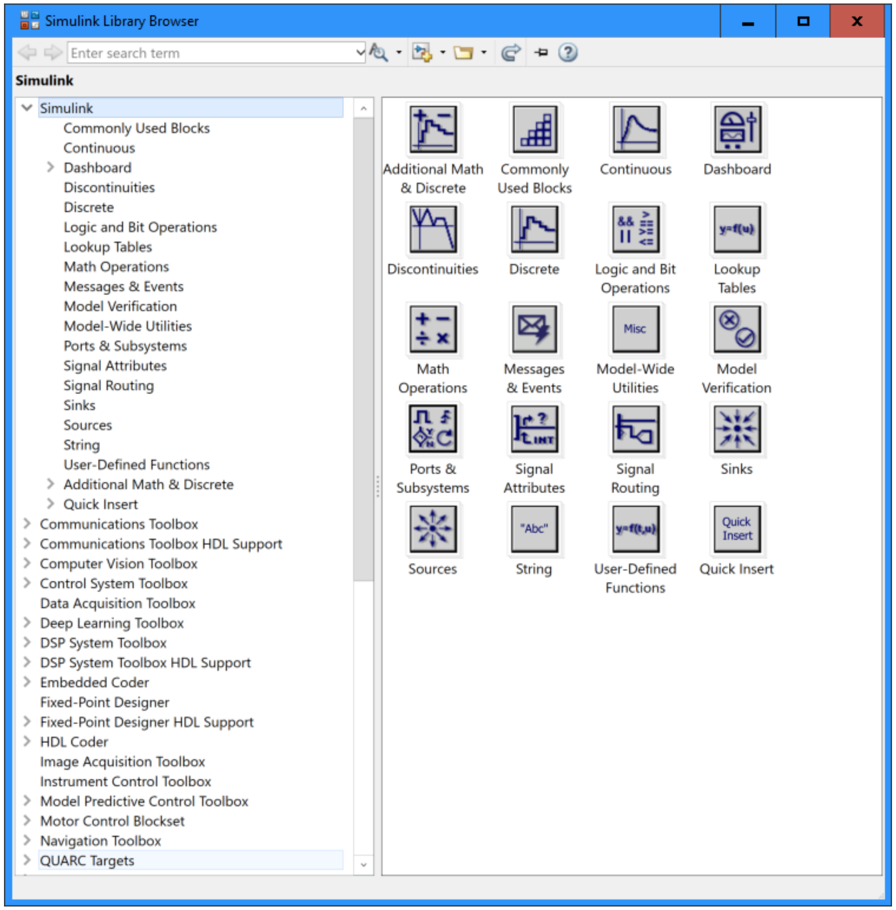
2. Use the icons in the Simulink Start Page to open up the Simulink model that you wish to run in real time (or create a new blank model). You can also open an existing Simulink model by double-clicking on the filename in the MATLAB window. If you are planning to do Real-Time Control, you can select "Blank QUARC Model," which will automatically preconfigure some of the model configuration parameters for you.



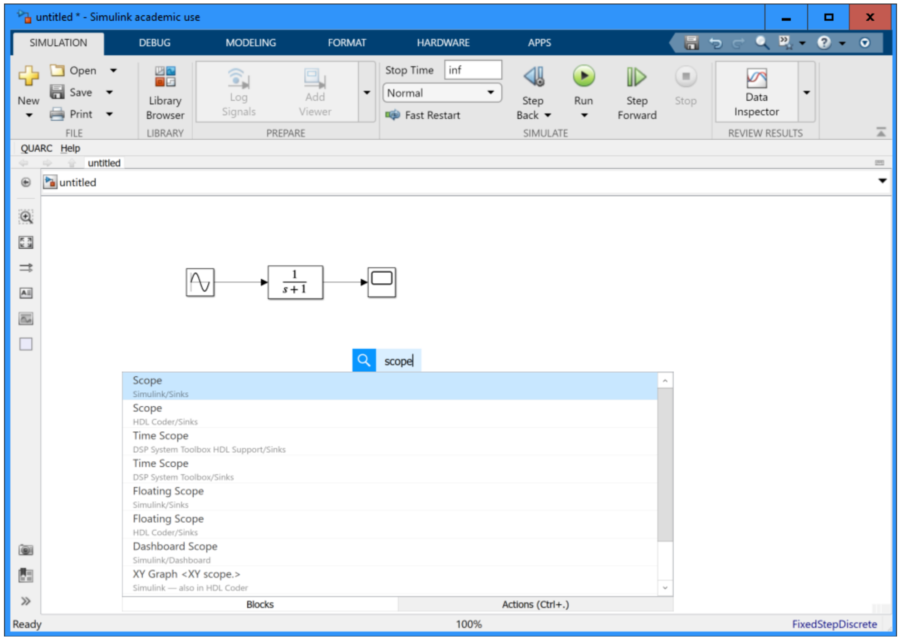
You will probably also want to open the Simulink Library Browser, by navigating to the Simulation tab of your Model window and clicking the Library Browser icon, or use the Ctrl+Shift+L shortcut.



Can't find what you're looking for? Try Apps in Simulink or view Menus to Toolstrip Mapping. Do not show again



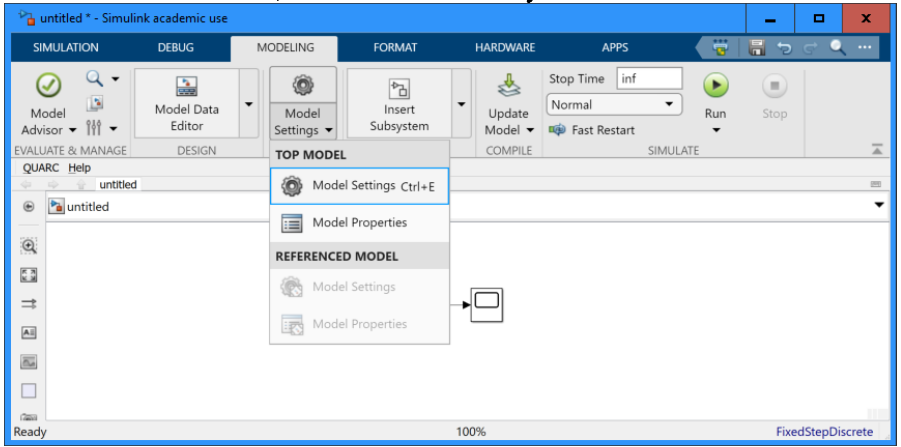
From the “Simulink Library Browser”, you can drag and drop various blocks into your model. For example, by looking in the Sinks, Sources, and Continuous folders of the library, you can create the model shown below with a Sine Wave, Transfer Fcn, and Scope block. As a short cut to search the library, you can double-click anywhere in your model and type in the name of the block you want.

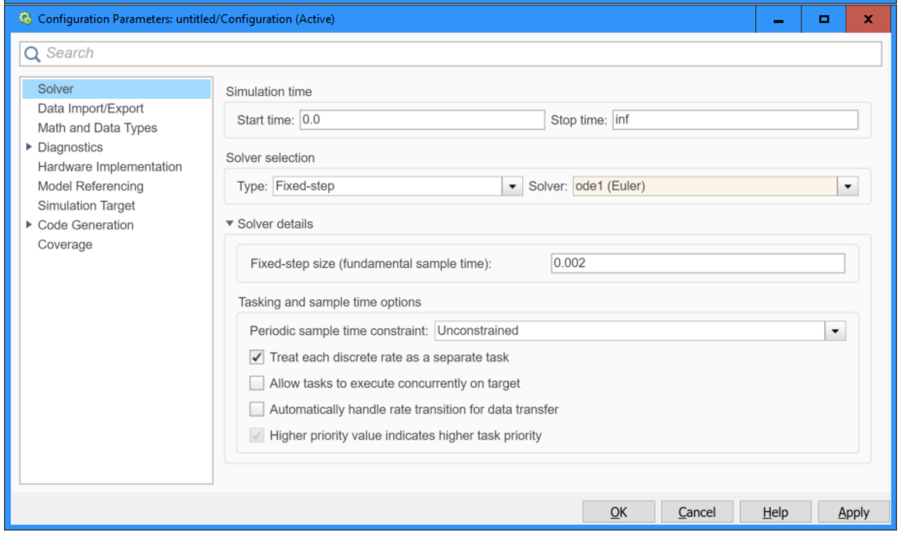


3. Under the Modeling tab, click on the Model Settings icon or use the Ctrl+E shortcut to open the Configuration Parameters window. On the Solver tab as shown below, do the following:
   a. Select “Fixed-step” under the Solver selection
   b. Select a continuous Solver such as ode1 to ode5. QUARC recommends ode1 (Euler).[1]

[1] A higher order solver like ode5 would result in more precise numerical approximations, but yields more computation in real-time code, which would limit the sampling rate you can achieve. Feel free to experiment with this.

c. Under Solver details, select a fixed step size (determines the sampling rate). QUARC recommends a sample time of 0.002 s, but 0.001 is usually the best we can do.




Configuration Parameters: untitled/Configuration (Active)

Solver

Simulation time

Start time: 0.0 Stop time: inf

Solver selection

Type: Fixed-step Solver: ode1 (Euler)

Solver details

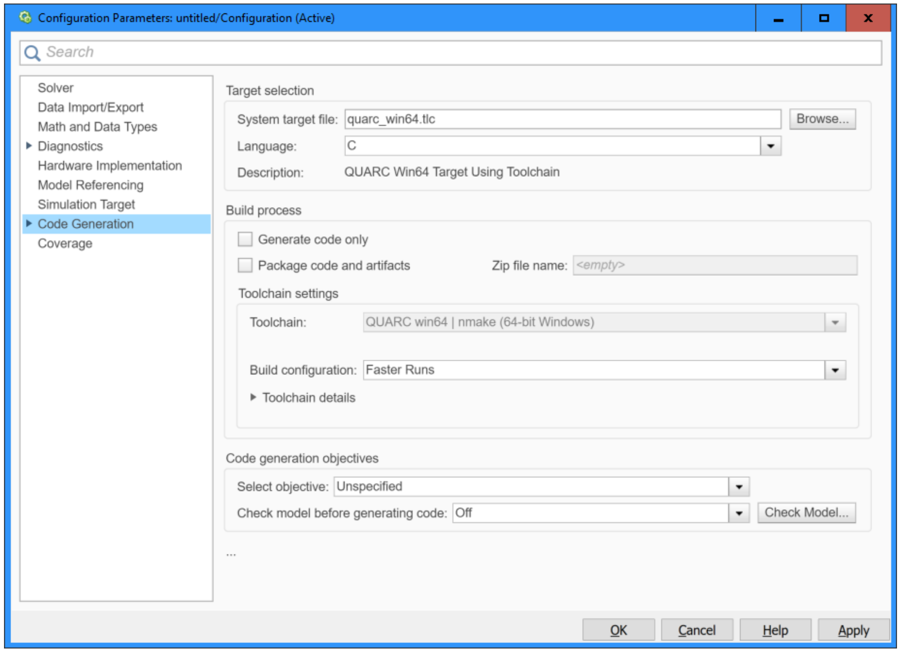
Fixed-step size (fundamental sample time): 0.002

Tasking and sample time options

Periodic sample time constraint: Unconstrained

☑ Treat each discrete rate as a separate task
□ Allow tasks to execute concurrently on target
□ Automatically handle rate transition for data transfer
☑ Higher priority value indicates higher task priority

d. While still in the Configuration Parameters, select the "Code Generation" tab and make sure that System target file is "quarc_win64.tlc".


Configuration Parameters: untitled/Configuration (Active)

Code Generation

Target selection

System target file: quarc_win64.tlc

Language: C

Description: QUARC Win64 Target Using Toolchain

Build process

□ Generate code only
□ Package code and artifacts Zip file name: <empty>

Toolchain settings

Toolchain: QUARC win64 | nmake (64-bit Windows)

Build configuration: Faster Runs

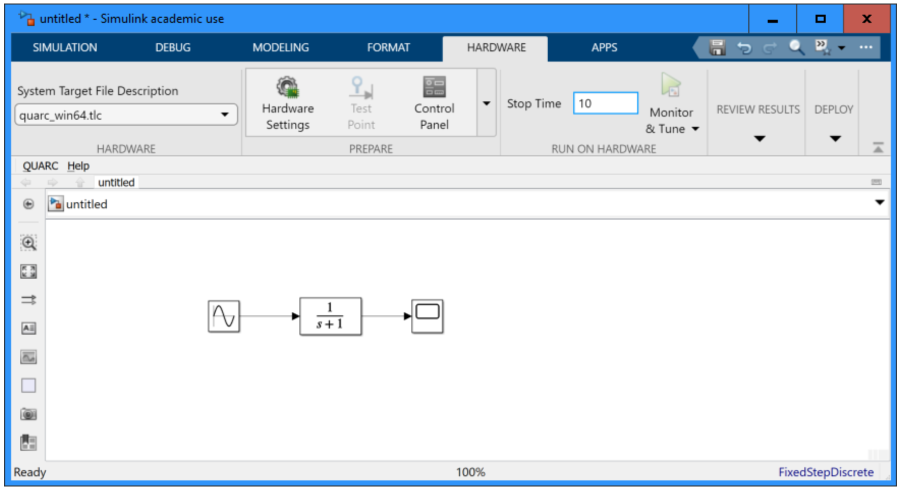
Toolchain details

Code generation objectives

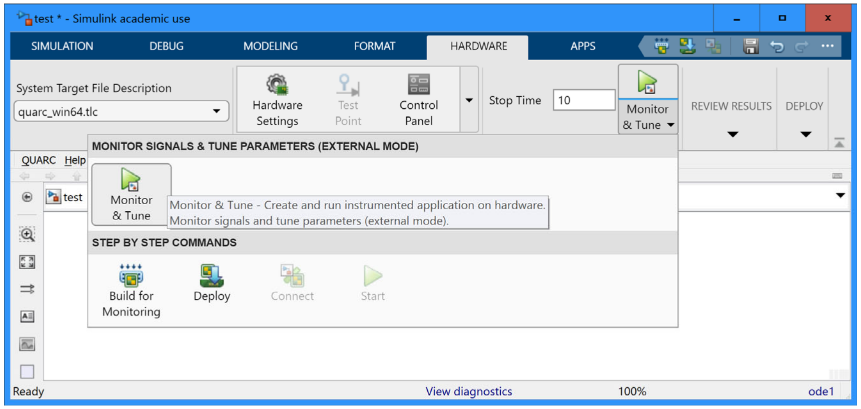
Select objective: Unspecified

Check model before generating code: Off

4. In order to run the simulation in Real-Time, you should navigate to the Hardware tab of your model window and use the "Run on Hardware" commands.
    a. You may also wish to specify the duration of the simulation by entering a Simulation Stop Time (e.g. 10 seconds) in the box just to the left of that. If you leave the Simulation Stop Time as "inf", the simulation will run indefinitely.
    b. Before you run your model, it's usually a good idea to save any changes you have made.

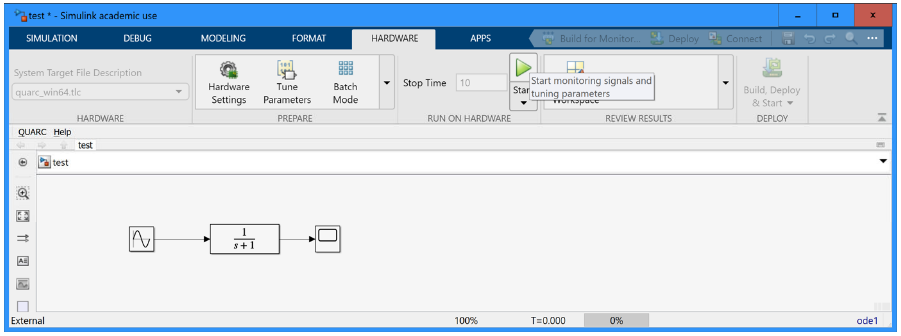


c. Now you can run the model in real time by clicking on the "Monitor & Tune" icon on the taskbar. This will *build* (compile) a real-time app, *deploy* the app to the hardware (your PC in this case), *connect* the app to your Simulink model (so you can tune parameters and monitor the scope signals in real time), and start the app. The model then runs in "External Mode" (as opposed to "Normal" mode).

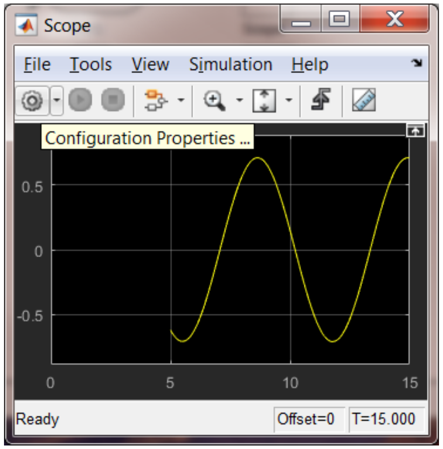


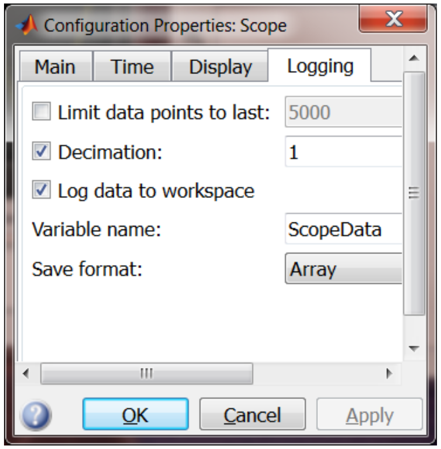
Monitor & Tune - Create and run instrumented application on hardware.
Monitor signals and tune parameters (external mode).

d. Alternatively, you can initiate the *Build*, *Deploy*, *Connect* and *Start* actions step by step. The advantage of using the step by step commands is that you can restart your model quicker without building. The only time you need to rebuild your model is when you make a structural change to your model. If you tune a parameter (i.e. change a numerical value) in one of your blocks, you can usually skip the *build* step. While the Build command has a keyboard shortcut (Ctrl+B), the other step by step commands unfortunately do not. Therefore it may be convenient to right-click on each command and add them to your Quick Access Toolbar. Once you connect, then the "Monitor & Tune" icon changes to a "Start" icon.



5. By default, the number of data points that Simulink displays/logs is limited. If the duration of your simulation is too long, you may notice that only the last few seconds of your data will be displayed/logged. In order to prevent this, a few settings must typically be changed:
    a. On the scope window, click the "Configuration Properties" icon, select the "Logging" tab, and turn off "Limit data points to last:" as shown below:





□ Limit data points to last: 5000

☑ Decimation: 1

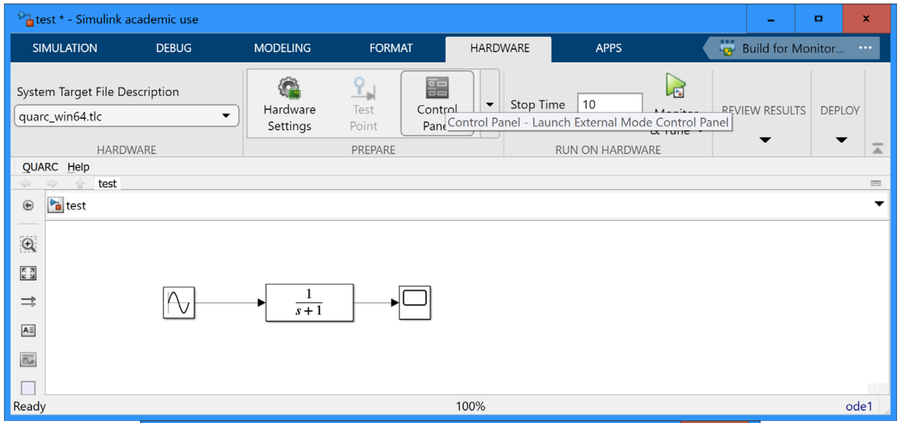
☑ Log data to workspace

Variable name: ScopeData

Save format: Array

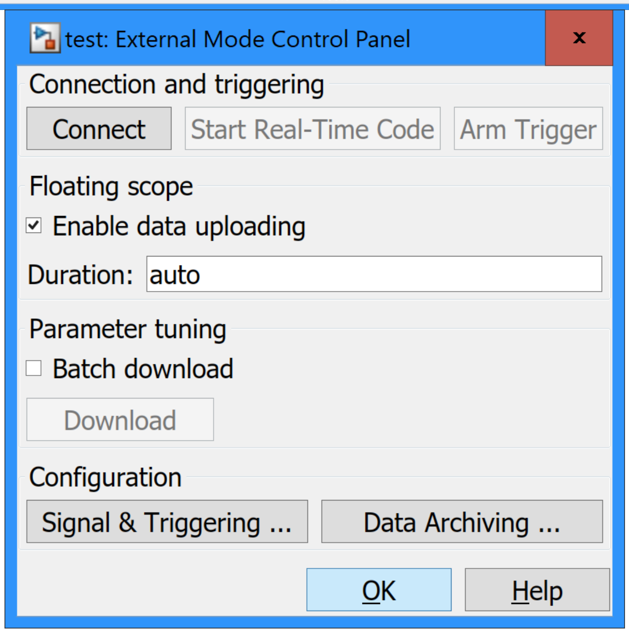
b. While you are in the logging tab of your scope, you may wish to check the "Log data to workspace" box so that you can analyze your data in the MATLAB workspace.

c. While in the Hardware tab of your Simulink menu, click on the "Control Panel" to bring up the "External Mode Control Panel" as shown in the figure below. In the External Mode Control Panel, select the "Signal & Triggering" button. In the External Signal & Triggering Panel, modify the Duration setting according to your needs. For example, if you are sampling every 0.002 sec (i.e. your step size) and have a Simulation Stop Time of 20 seconds, then you would need to set the Duration to 10000 data points in order to log the entire simulation.


test * - Simulink academic use
SIMULATION DEBUG MODELING FORMAT HARDWARE APPS
Build for Monitor...
Hardware Settings
Test Point
Control Panel
Control Panel - Launch External Mode Control Panel
Monitor & Tune
REVIEW RESULTS
DEPLOY
HARDWARE PREPARE RUN ON HARDWARE
QUARC Help
test
1
s+1
Ready
100%
ode1

System Target File Description: quarc_win64.tlc

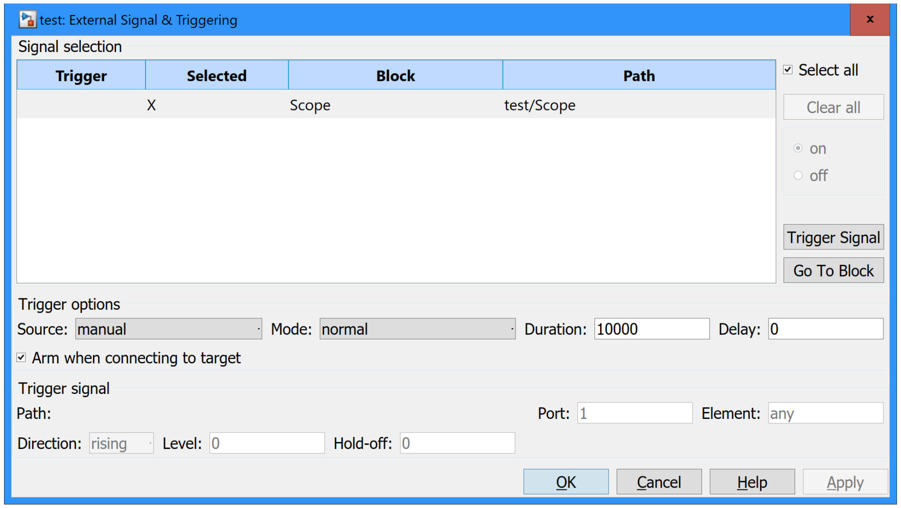
Stop Time: 10



☑ Enable data uploading

Duration: auto

□ Batch download



Signal selection

Trigger	Selected	Block	Path
	X	Scope	test/Scope

☑ Select all

○ on
○ off

Trigger options

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

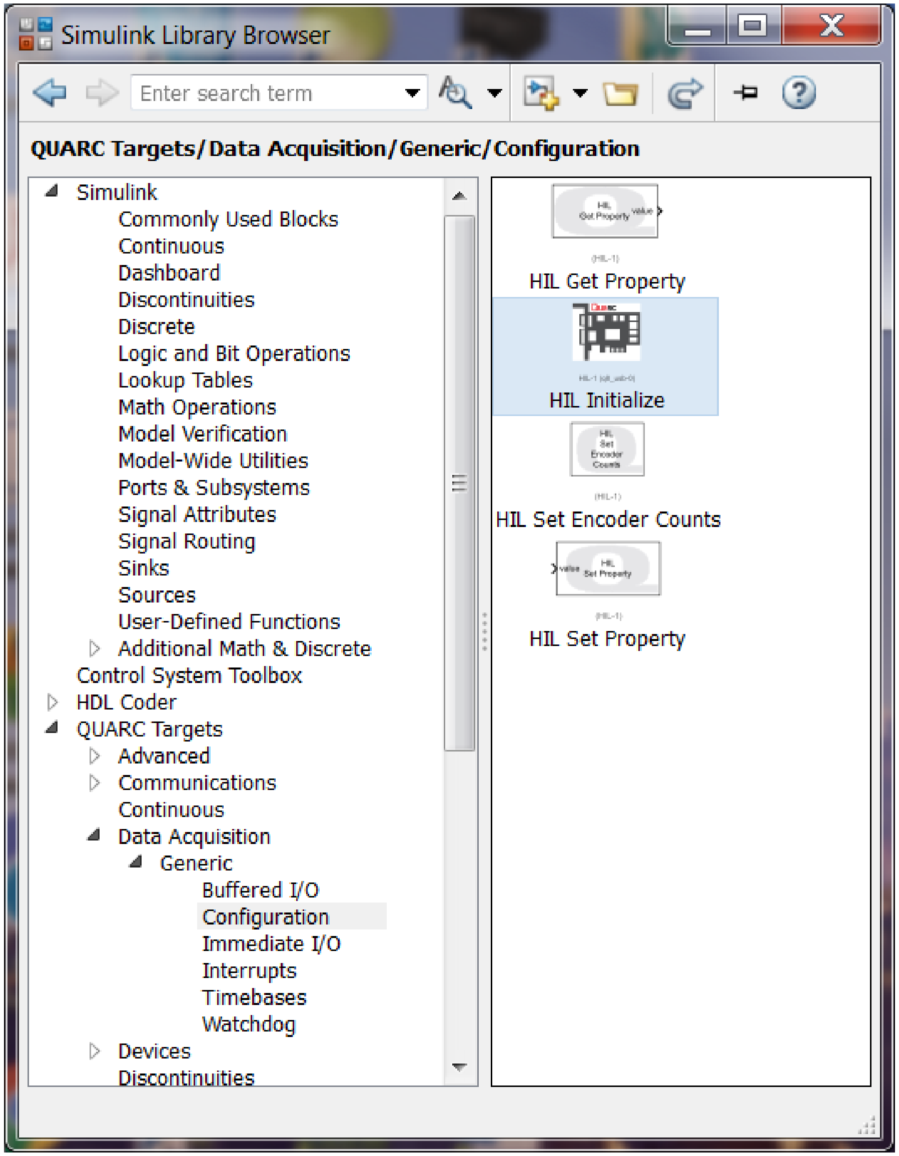
☑ Arm when connecting to target

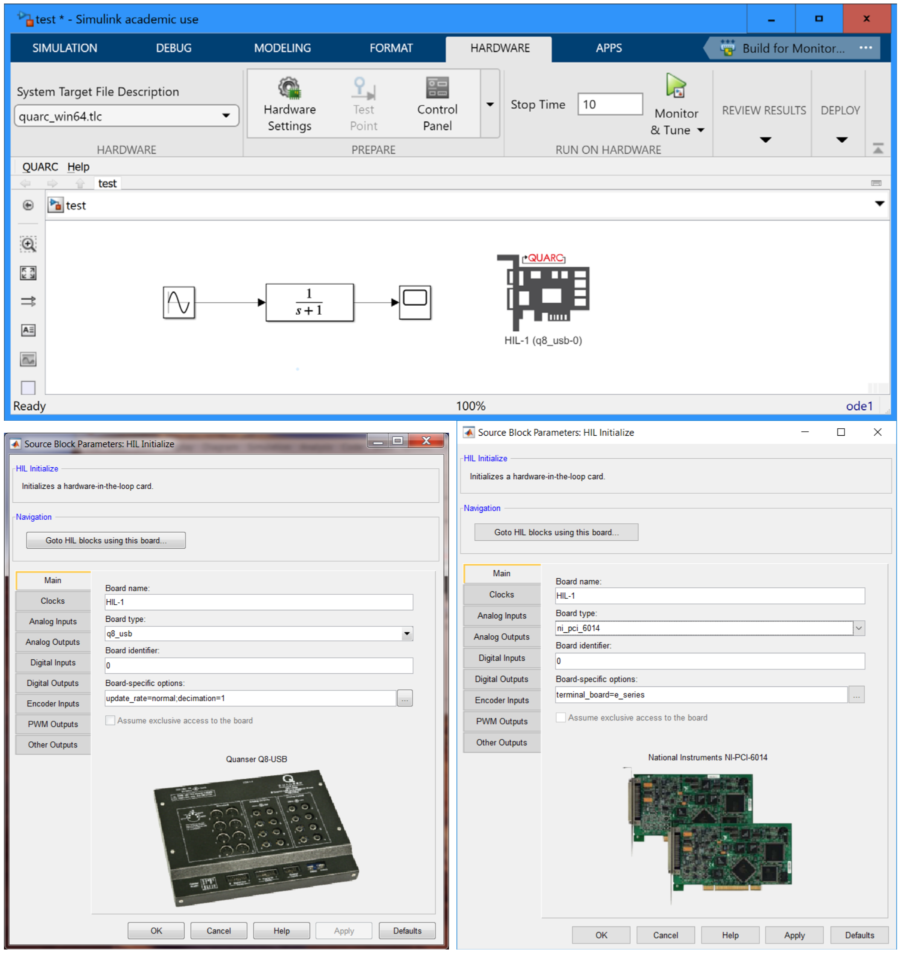
Trigger signal

Path: Port: 1 Element: any

Direction: rising Level: 0 Hold-off: 0

6. To interface your model with data acquisition hardware connected to the computer, you must do the following.
    a. You must first add an "HIL Initialize" block to initialize the "Hardware-in-the-loop." This is found in the "Simulink Library Browser" window under "QUARC Targets → Data Acquisition → Generic → Configuration" as shown below:





System Target File Description: quarc_win64.tlc

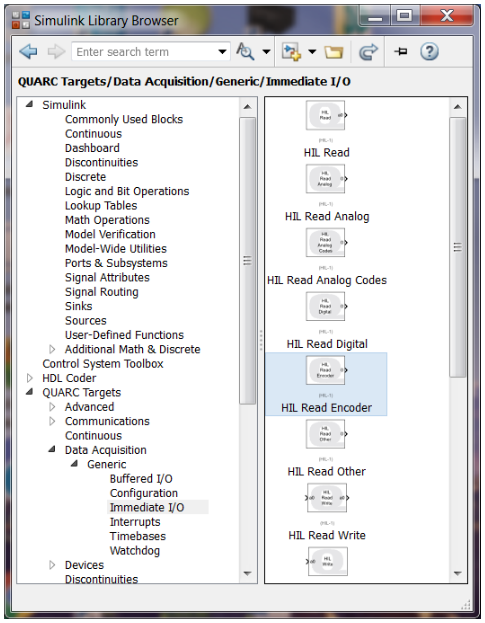
Stop Time: 10

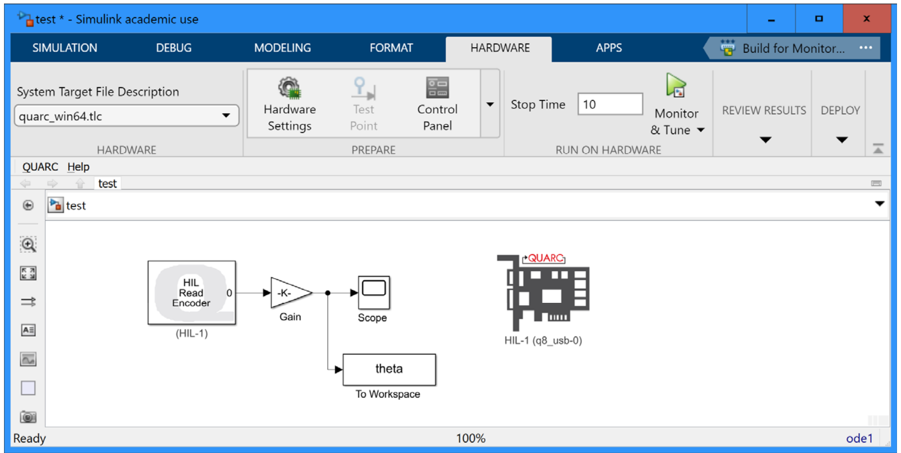
Board name: HIL-1
Board type: q8_usb
Board identifier: 0
Board-specific options: update_rate=normal;decimation=1

Board name: HIL-1
Board type: ni_pci_6014
Board identifier: 0
Board-specific options: terminal_board=e_series

b. After adding the HIL Initialize block, you must double-click on it to configure the data acquisition board. For Robot Control, we will generally be using the Quanser Q8-USB board, which is connected to your computer via one of the USB ports. However some stations may have a Q2-USB board. Under the board type, make sure the appropriate board is connected. The name and picture of the board that are displayed should match the board you are using.

c. To count encoder pulses, add a "HIL Read Encoder" block to the model. This is found in the "Simulink Library Browser" window under "QUARC Targets → Data Acquisition → Generic → Immediate I/O" as shown below:

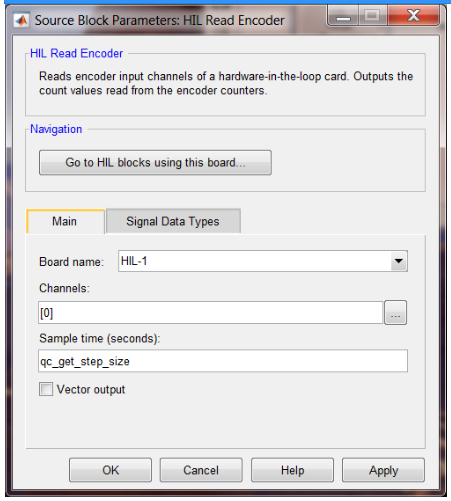




System Target File Description: quarc_win64.tlc

Stop Time: 10

theta


Source Block Parameters: HIL Read Encoder
Navigation
Go to HIL blocks using this board...
Main
Signal Data Types
OK
Cancel
Help
Apply

HIL Read Encoder

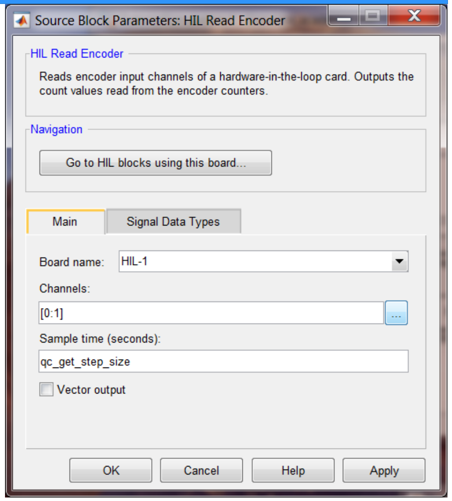
Reads encoder input channels of a hardware-in-the-loop card. Outputs the count values read from the encoder counters.

Board name: HIL-1

Channels: [0]

Sample time (seconds): qc_get_step_size

□ Vector output


Source Block Parameters: HIL Read Encoder
Navigation
Go to HIL blocks using this board...
Main
Signal Data Types
OK
Cancel
Help
Apply

HIL Read Encoder

Reads encoder input channels of a hardware-in-the-loop card. Outputs the count values read from the encoder counters.

Board name: HIL-1

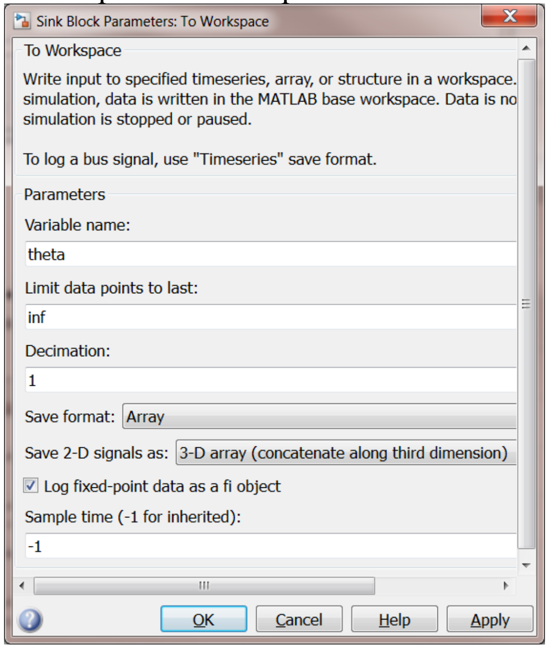
Channels: [0:1]

Sample time (seconds): qc_get_step_size

□ Vector output

d. After adding the Read Encoder block, you can double-click on it to configure it. Select the channel(s) you want to use. For 1-DOF experiments, this would typically be Channel 0. For 2-DOF experiments, you would choose both channels [0;1].[2]

e. You will also need to add a gain block to convert to units of choice (revolutions or degrees or radians). You can then send the output to a "Scope" and/or "To Workspace" block. If you use a "To Workspace" block, then you will need to double-click on it and set the Save Format to "Array" or "Structure with Time" depending on your needs. The "TimeSeries" option is incompatible with external mode.



To Workspace

Write input to specified timeseries, array, or structure in a workspace. simulation, data is written in the MATLAB base workspace. Data is no simulation is stopped or paused.

To log a bus signal, use "Timeseries" save format.

Parameters

Variable name: theta

Limit data points to last: inf

Decimation: 1

Save format: Array

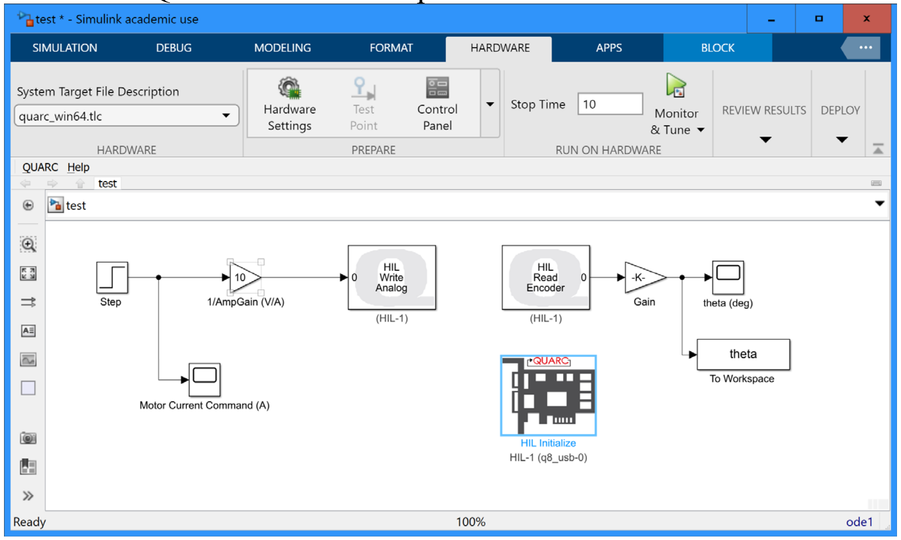
Save 2-D signals as: 3-D array (concatenate along third dimension)

☑ Log fixed-point data as a fi object

Sample time (-1 for inherited): -1

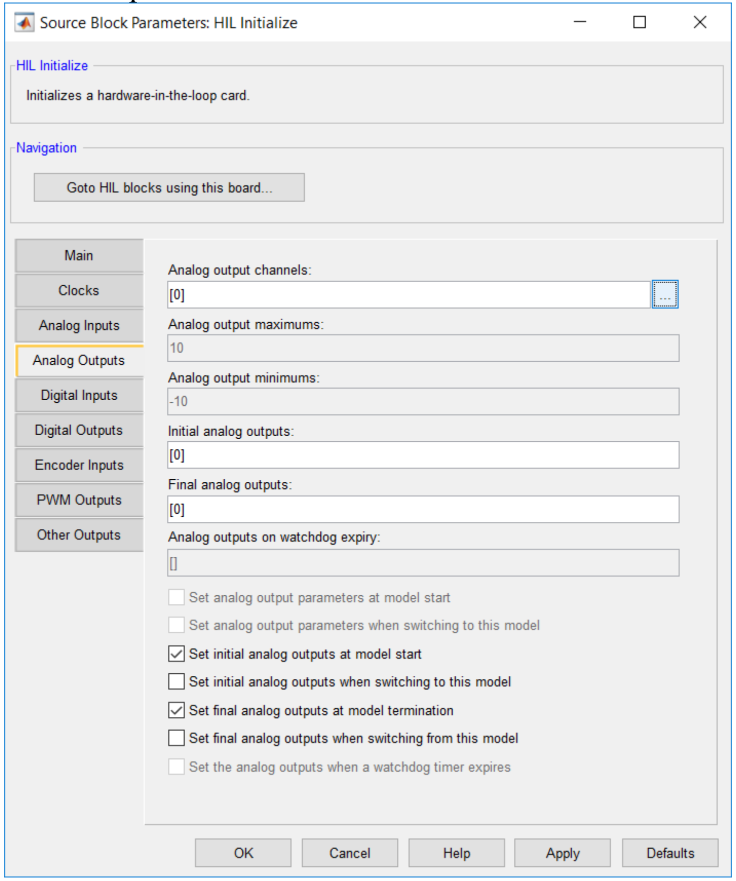
[2] If you are using multiple channels, you may also want to check the "Vector Output" box, so that the Read Encoder block outputs an array of encoder counts on a single port, rather than using two separate output ports.

f. Next, to send a voltage out to the servo amplifier, add an "HIL Write Analog" block to the model from the QUARC Immediate I/O library. You can also add a gain block before the analog write block to account for the gain of the power amplifier so that your motor command is in units of Amps. **Begin with a small motor command!** It is also a good practice to put a scope on your motor command to ensure you are not saturating the amplifier during your experiment. Saturation introduces an unwanted nonlinearity in the dynamics. The AMPAQ amplifiers from Quanser have an AmpGain of 0.1 A/V and saturate at 1A.



g. Double click on the "Write Analog" block to configure the channel(s) in the same manner you did for the Read Encoder block. Note that in general, you should not alter the Sample Time settings within the blocks. By default, the sample time for each of these blocks is inherited from the value you chose it to be in the Configuration Parameters.

h. Next, reopen the "HIL Initialize Block" and click on the "Analog Outputs" tab. Add channel 0 to the "Analog output channels" and make sure that the "Final analog outputs" value is set to zero. This will ensure that the voltage command to the amplifier resets to zero when your model stops running. If you don't do this properly, the motor may continue running at high power even after your model stops!



Initializes a hardware-in-the-loop card.

Analog output channels: [0]

Analog output maximums: 10

Analog output minimums: -10

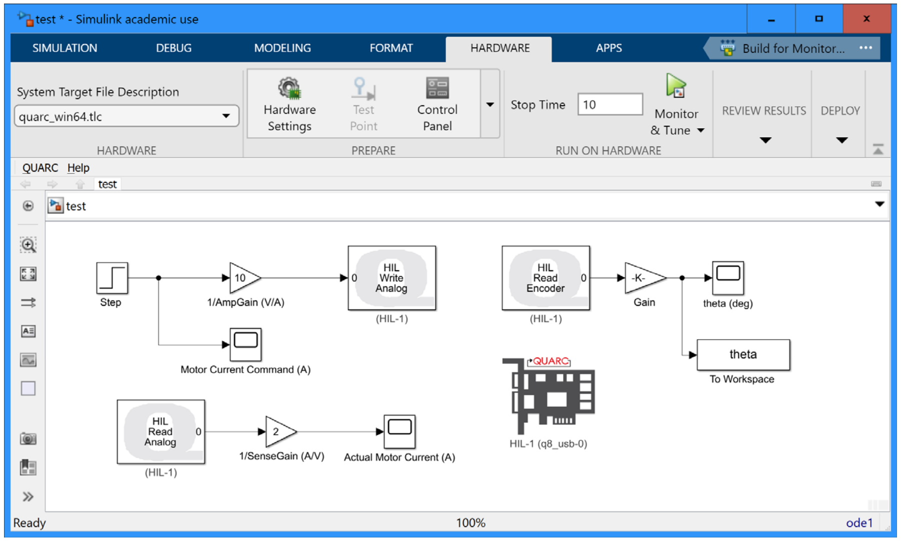
Initial analog outputs: [0]

Final analog outputs: [0]

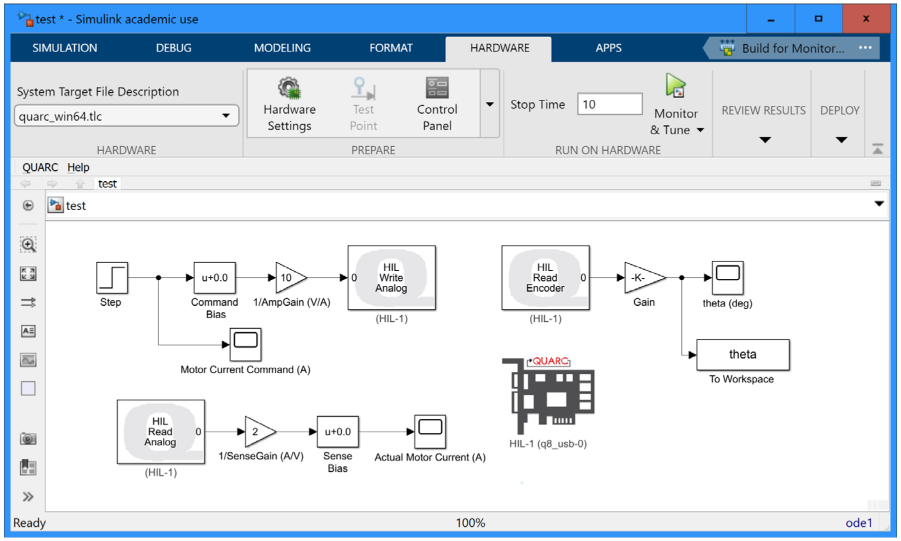
Analog outputs on watchdog expiry: []

□ Set analog output parameters at model start
□ Set analog output parameters when switching to this model
☑ Set initial analog outputs at model start
□ Set initial analog outputs when switching to this model
☑ Set final analog outputs at model termination
□ Set final analog outputs when switching from this model
□ Set the analog outputs when a watchdog timer expires

i. Finally, it is also a good practice to monitor the actual current in the motors to ensure it is the same as the commanded current. To do so, add an “HIL Read Analog” block from the QUARC Immediate I/O library and connect to a scope. The current sense output on the AMPAQ amplifiers has a SenseGain of 0.5 V/A so an additional gain block is needed to account for this.


System Target File Description
quarc_win64.tlc
Stop Time
10

j. To compensate for bias in the amplifiers, you can add “Bias” blocks from the “Simulink → Math Operations” library. The biases generally differ on each amp/channel. First determine if there is a bias on the current sense by disabling the amp with the E-stop (to ensure actual current is zero) and viewing the current sense output. If the current sense is not reading zero, place a bias block before your current sense scope and adjust the bias value sensed current reads zero. Second, determine if there is any bias on the current command by comparing the actual motor current to the commanded current when the amp is enabled. Release the E-stop (twist it and it will pop up). Place another bias block on your current command, and adjust the bias value until the actual motor current matches the commanded motor current.[3]


System Target File Description
quarc_win64.tlc
Stop Time
10

k. You are now ready to experiment with your hardware and add feedback control. Assuming that your hardware is wired correctly, and the Emergency Stop Button (E-stop) is released (twist it and it will pop up), you should see your hardware move. **If the E-stop is depressed or unplugged, the amplifier will not output any power to your motor.**

[3] If your application requires there to be no bias current even when your simulation is stopped, you can also modify the parameters of the Analog Outputs in the “HIL Initialize” block. Specifically, you can set the initial and final analog output voltages to offset the bias (you need to account for the amp gain when doing so).

l. **VERY IMPORTANT: Begin with a small motor command! Before implementing any feedback loops, a good practice is to see whether a positive command signal to your motors drives the motors in the direction you want to define as positive angle, and whether a positive rotation results in a positive encoder reading. If not, you need to put a negative sign on your 1/AmpGain and/or Encoder Gain blocks.**

Please Note:

- Any changes to the settings or structure of the model require that the model is rebuilt. Changing values of certain parameters (e.g. changing the gain value of a gain block) does not require a rebuild; however if in doubt, it never hurts to rebuild your model.
- If MATLAB or Windows crashes, the most likely explanation is that the sampling time is too small. If you experience recurrent crashing, try choosing a larger sampling time or using a lower-order solver.
- These instructions have been tested on **Windows 10 with MATLAB R2021b, Microsoft Visual Studio 2019, and QUARC 2021 SP1**[4]. Other combinations of software versions have not been tested by us. Models created in MATLAB R2021b may or may not be compatible with other versions of MATLAB and vice versa.

[4] Use the QUARC help menus for more information on QUARC features. There are many interesting blocks in the QUARC libraries that may be useful for special projects.