MTRE4002 Feedback Control Laboratory

Lab #3 Position Control of a DC Motor System

By Dr. Ying Wang

Updated by Dr. Amir Ali Amiri Moghadam

College of Engineering

Kennesaw State University

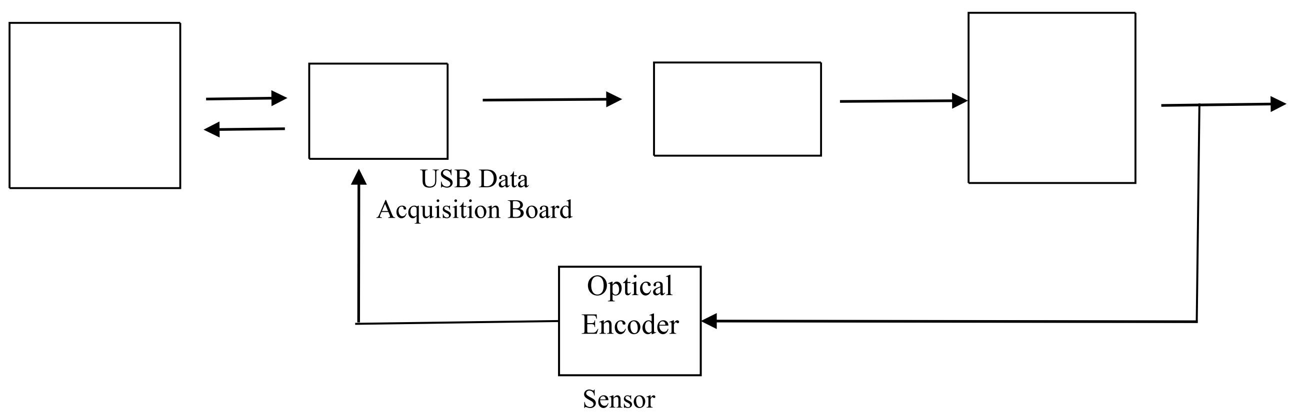
Marietta, GA

Objective

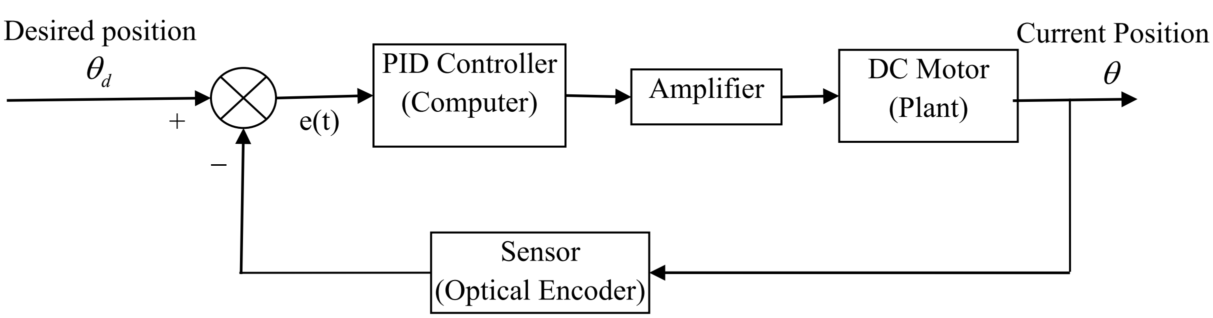
In this lab, students are required to create PID controllers to implement a DC motor position control system. Through tuning the PID gains and measuring the corresponding performance specifications, the effects of the PID control gains will be assessed.

Background knowledge

A DC motor position control system is shown below:



In the above system, the current angular position θ is measured by the optical encoder, and this information is sent to the computer via the data acquisition board. The computer (controller) will compare θ_d (the desired position of the motor) to θ, calculate the error, and determine the control command. The control command (a voltage signal) will be sent to the amplifier through the data acquisition board. Finally, the DC motor's angular position θ is changed. The corresponding block diagram of the above control system is as follows:



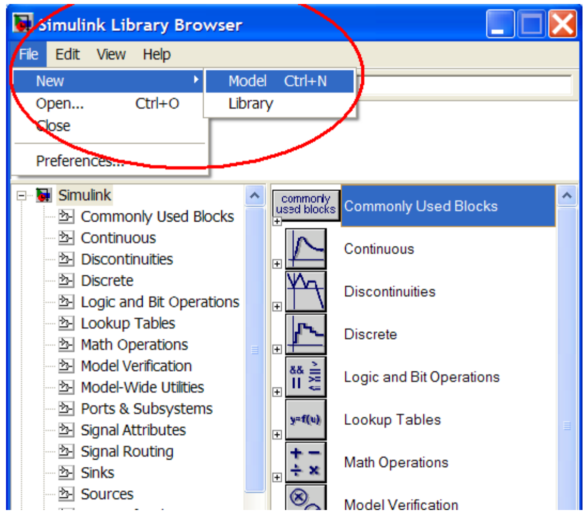
Lab Procedures

1. Proportional Control of the DC Motor

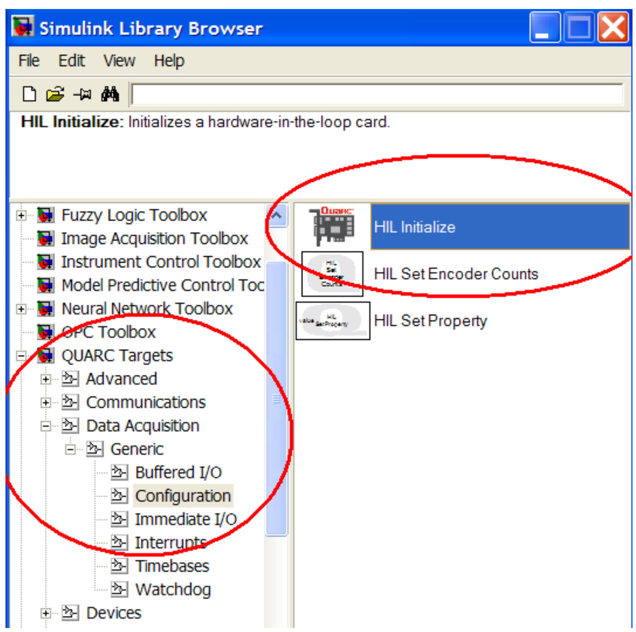
(1) Start MATLAB

(2) Start Simulink

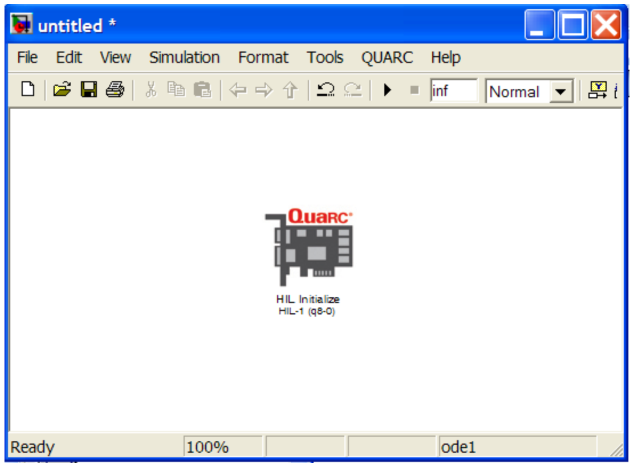
(3) In Simulink, click the menu “File” --- > “New” --- > “Model” to create a blank model.



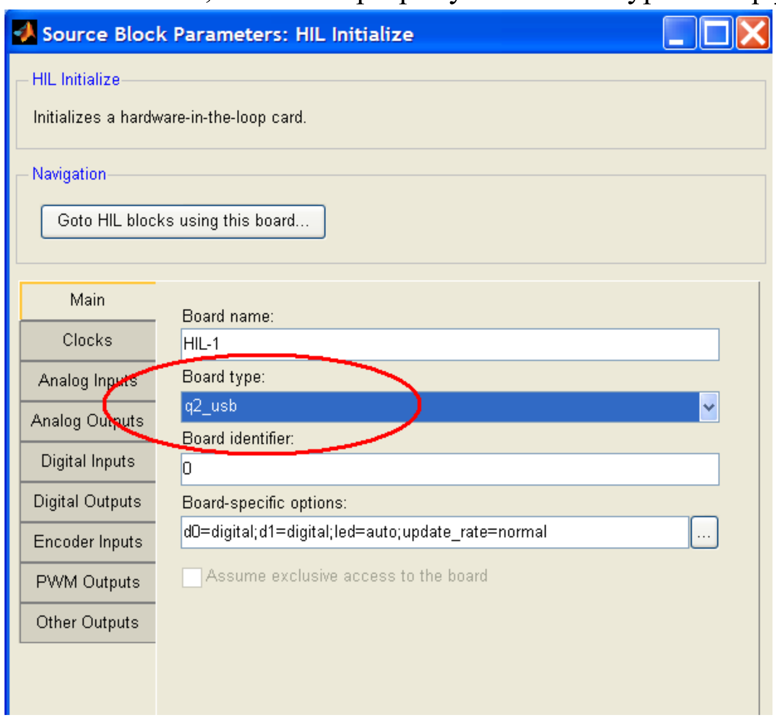
(4) In the “Simulink Library Browser”, click the category “QUARC Targets” --- > “Data Acquisition” -- >“Generic” --- > “Configuration” , and find the block “HIL Initialize”.



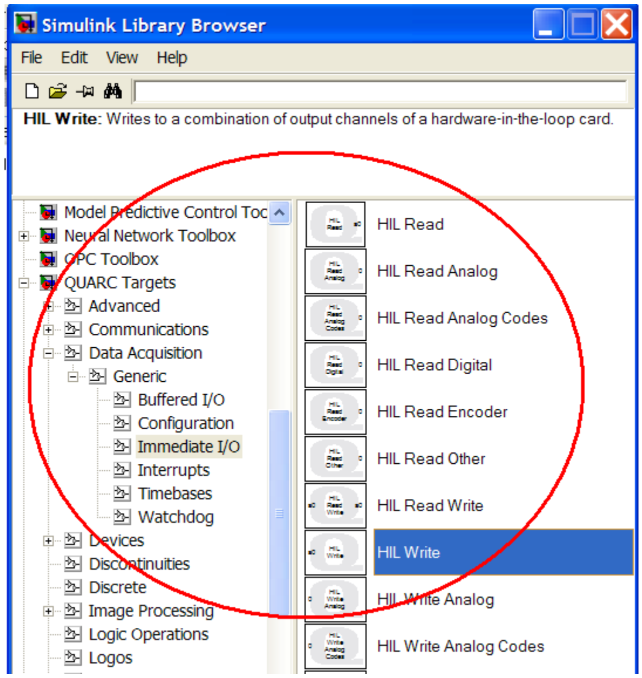
(5) Drag the block “HIL Initialize” to your blank Simulink model as follows.



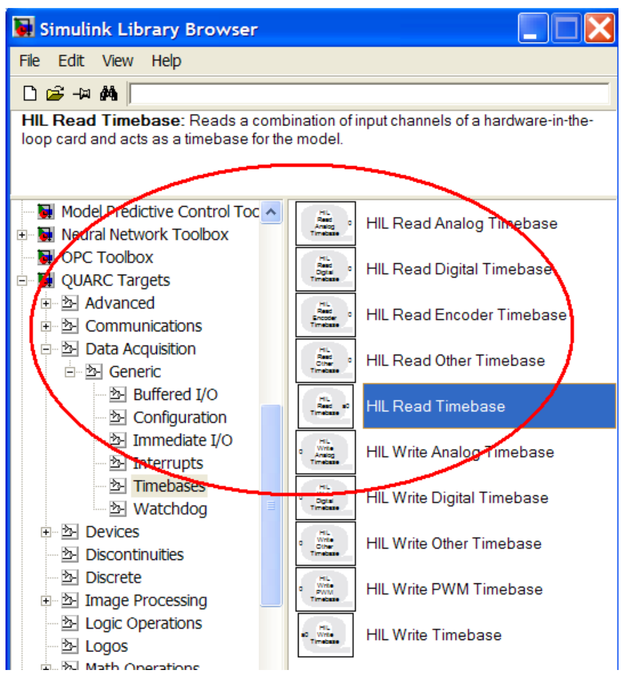
(6) Double click the block "HIL Initialize", and set its property of "Board Type" to "q2_usb".



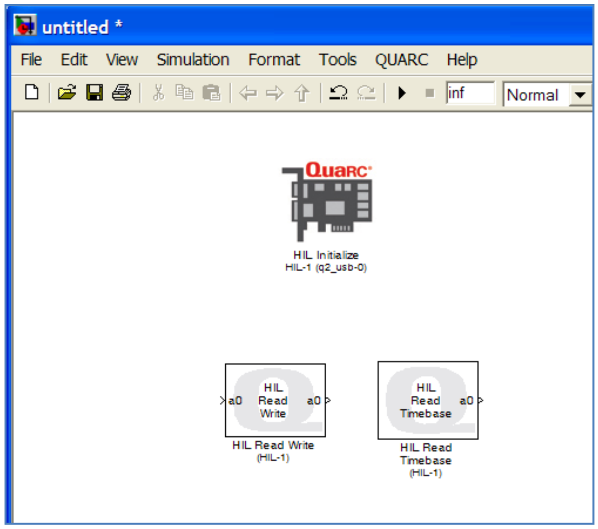
(7) In the "Simulink Library Browser", click the category "QUARC Targets" --- > "Data Acquisition" -- >"Generic" --- > "Immediate I/O" , and find the block "HIL Write". Drag this block to your Simulink model.



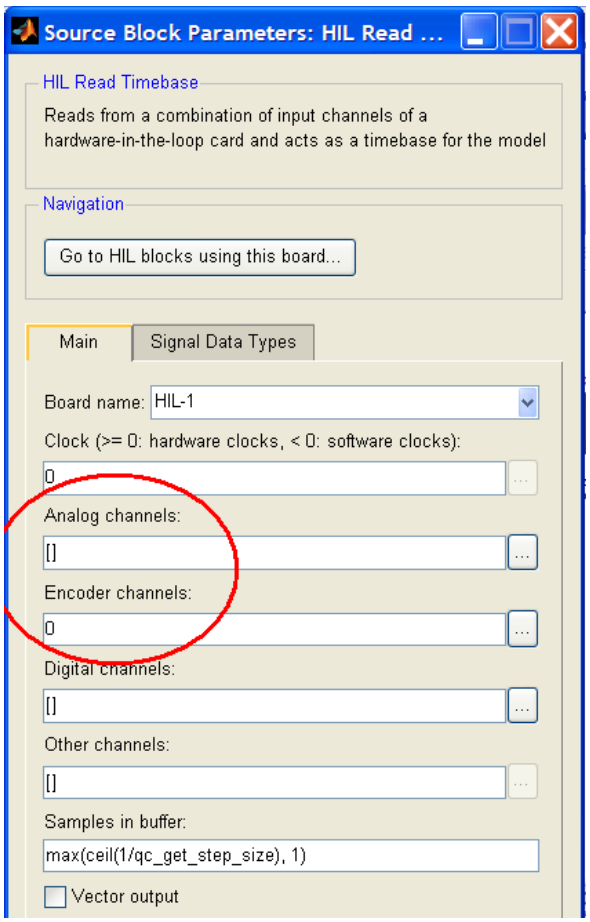
(8) In the “Simulink Library Browser”, click the category “QUARC Targets” --- > “Data Acquisition” -- >“Generic” --- > “Timebases” , and find the block “HIL Read Timebase”. Drag this block to your Simulink model.



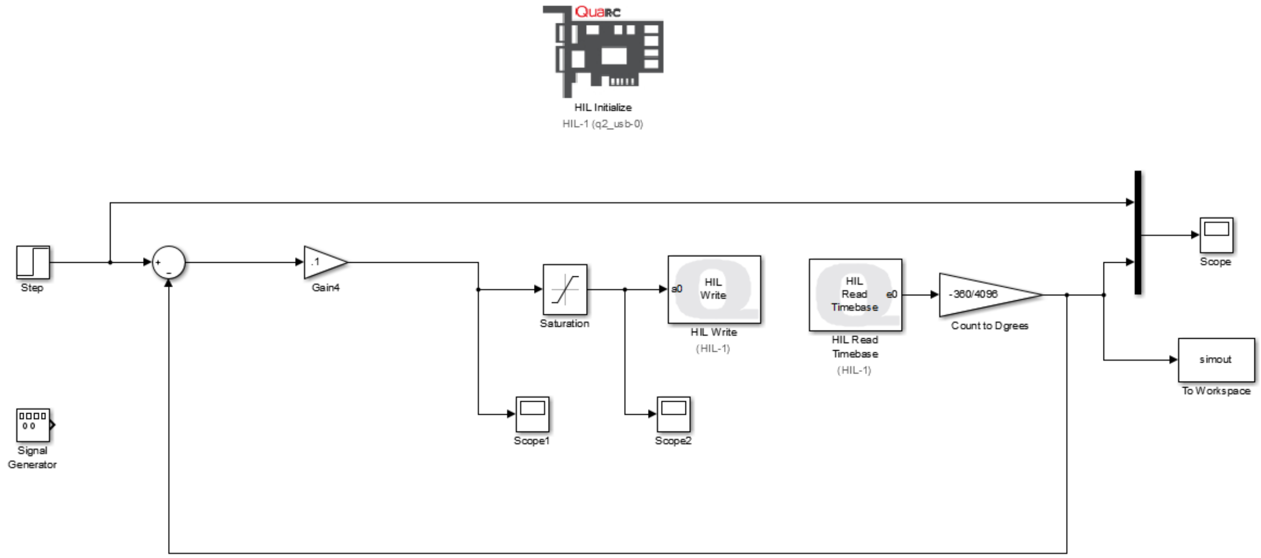
(9) Now your Simulink model looks like the following figure.



(10) Double click the "HIL Read Timebase" block, and set its properties as follows.

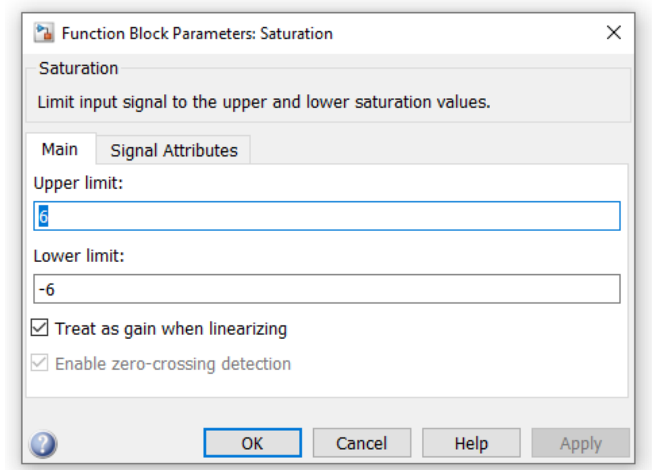


(11) Add the additional blocks to your Simulink model. Finally, it will look like the following figure.

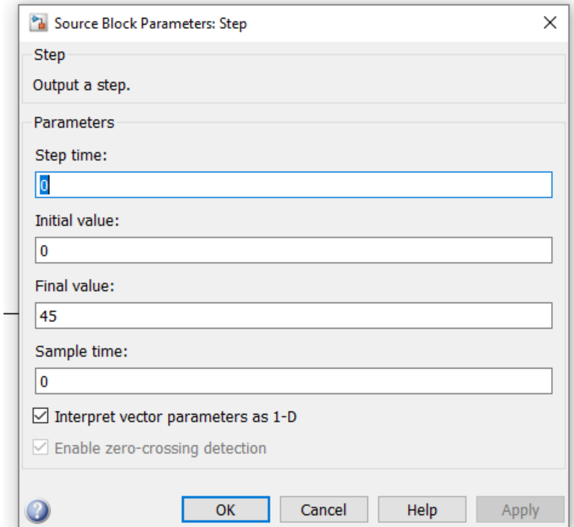


Where, the “Signal Generator” block can be found in the category of “Sources” in the Simulink Library Browser.

(12) Add a saturation block and sets the **limit 6 and – 6**. **This protect the motor from high voltages!**



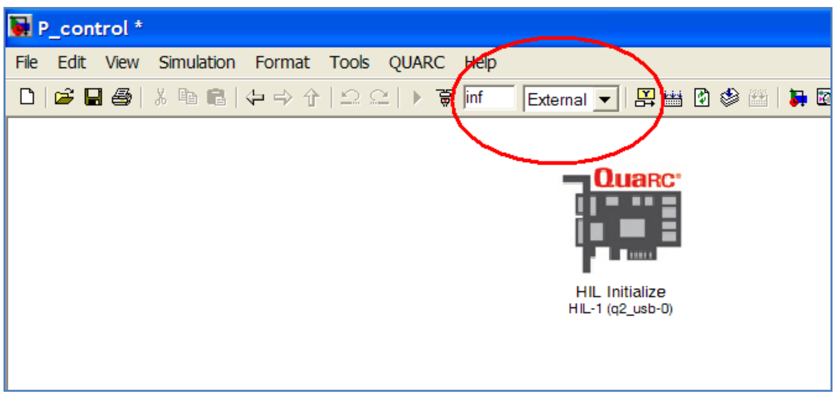
(13) Double click the “Step” block,, and set its properties below:



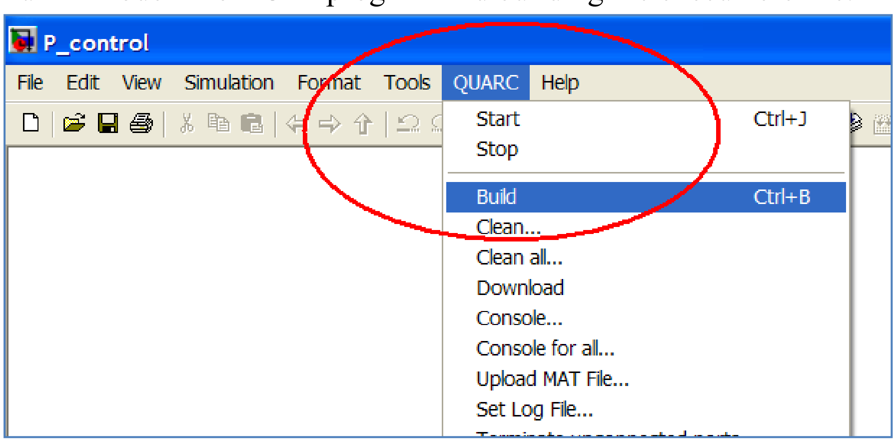
(14) Double click the gain of "Count to Degree" and set its value of "**-360/4096**".

(15) Double click the gain of k_p (the gain of the proportional controller), and set its value of "0.05".

(16) Set "External" in the time type in your Simulink model window as below.

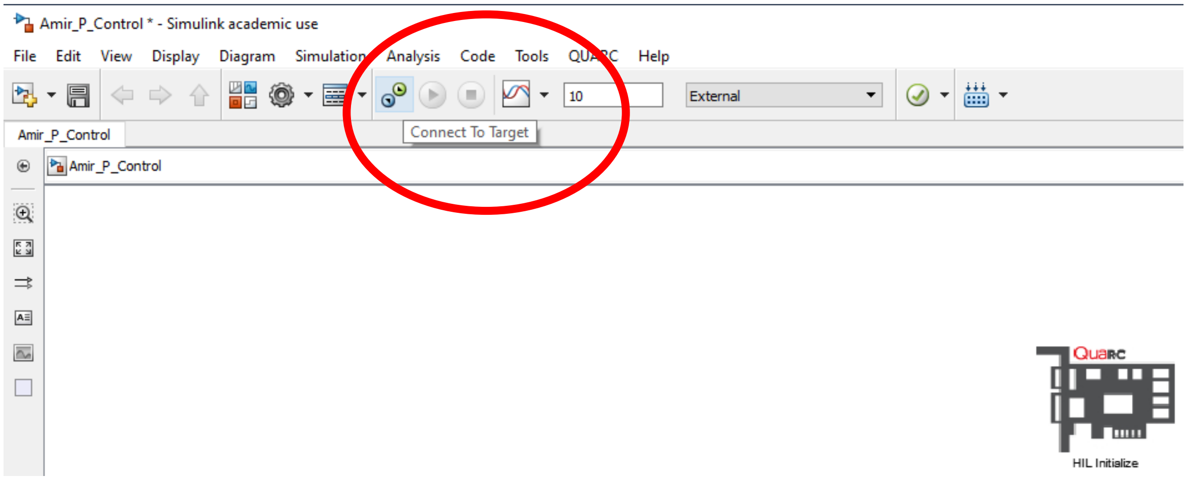


(17) Click the menu "QUARC" --- > "Build Libraries". Wait for some time until MATLAB finishes compiling your Simulink model into a C++ program and building the executable file.

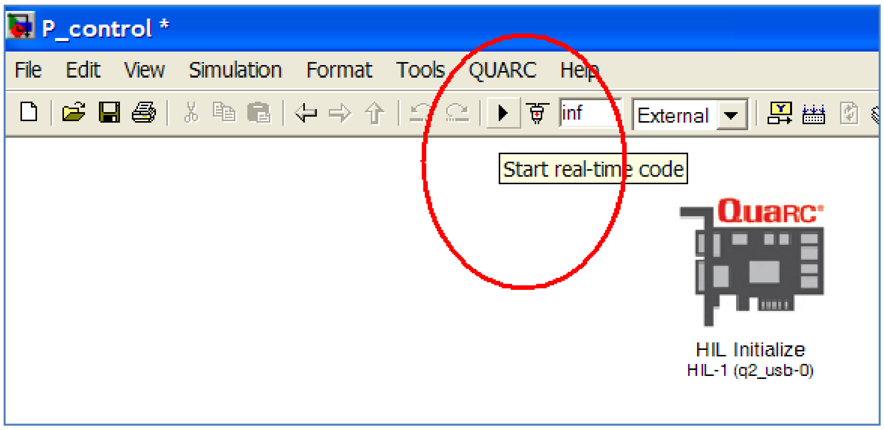


(18) Turn the power of your amplifier, connect the USB cable of your data acquisition board to the computer, check all cable connections, and manually move the arm of the motor to the position of "zero degree".

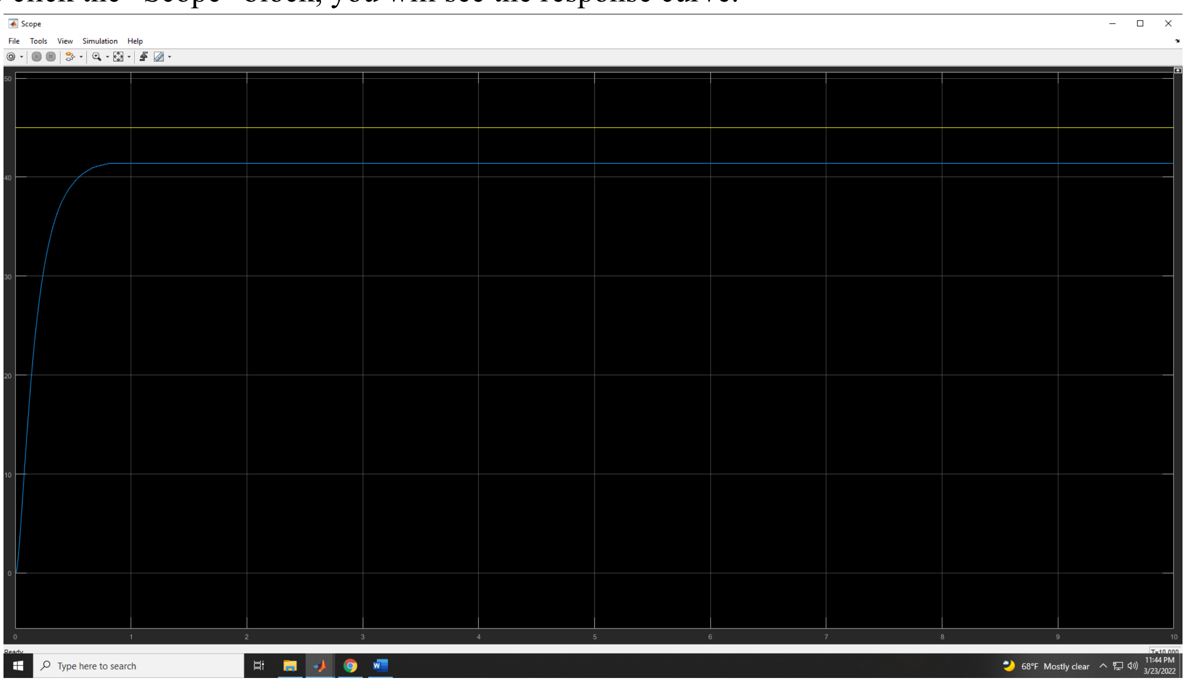
(19) Click the button of "Connect to Target" as shown below.



(20) Click the button of “Start real-time code” to run the program. You will see the DC motor is moving to the desired position (θ_d =45 degrees). **Make sure to set the simulation time to 10 seconds!**



(21) Double click the “Scope” block, you will see the response curve.



Based on the above response curve, you can measure various performance specifications (steady-state error, percent overshoot, rise time, and so on).

(22) The following code can be used to extract the data from simout file.

```
t=simout.time;
y=simout.signals.values;
yfinal=y(end);
plot(t,y)
S = stepinfo( y , t , yfinal )
```

(23) Please set the proportional gain k_p according to Table 1, run the Simulink model, measure the response curve and complete the table:

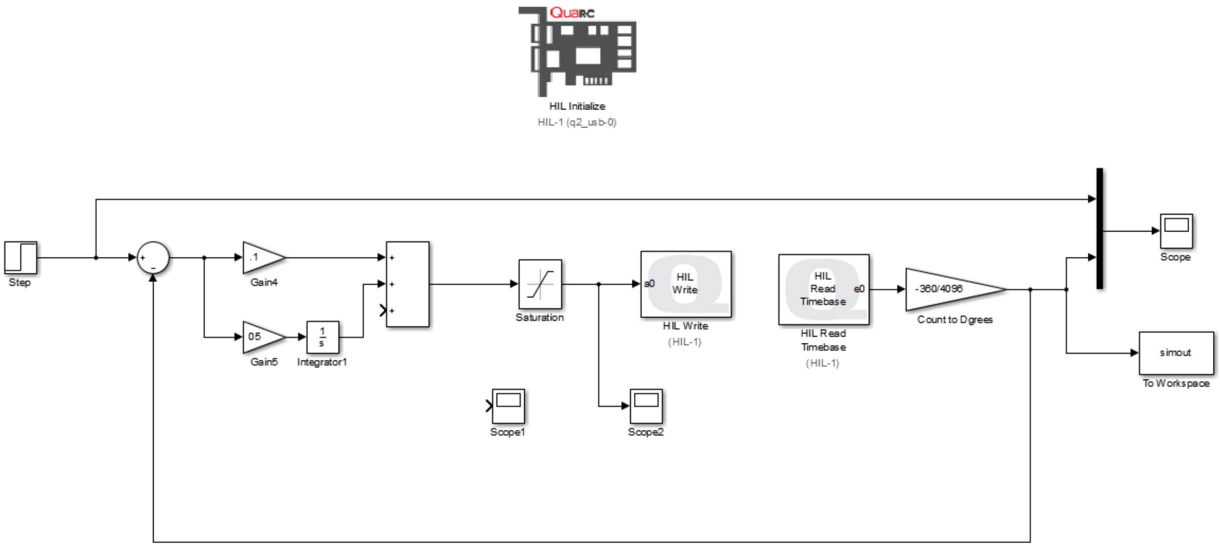
Table 1 Effects of Proportional Gain

Proportional Gain (k_p)	Rise Time (T_r)	Peak Time (T_p)	Setting Time (T_s)	Percent Overshoot (%OS)	Steady-state Error (e_{ss})	Is it a stable system? (Yes/No)	Type of the System (Under/Over/ Critically Damped System?)
0.05							
0.1							
0.3							
0.6							
1							
2							
3							
8							

Based on the information in the above table, please draw a conclusion.

2. PI controller

Manually move the arm of the motor to the position of "zero degree". Please create a PI control system in Simulink, as indicated below.



Please build the above system, set the proportional gain and integral gain according to Table 2, run the Simulink model, measure the response curve and complete Table 2:

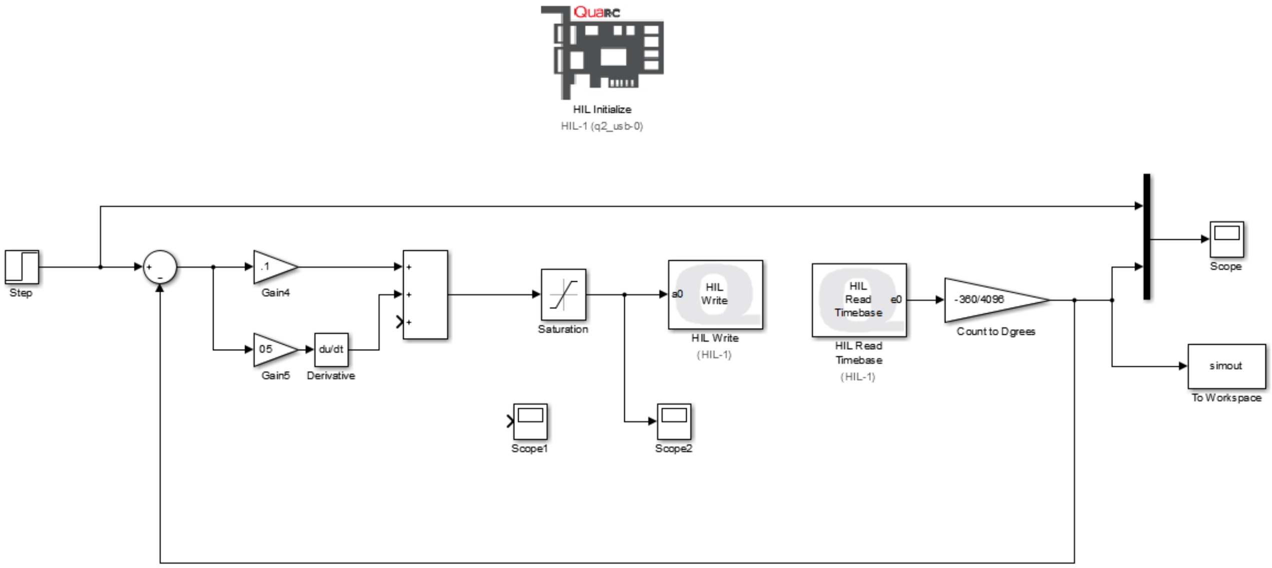
Table 2 Effects of Integral Gain

Proportional Gain (k_p)	Integral Gain (k_i)	Rise Time (T_r)	Peak Time (T_p)	Setting Time (T_s)	Percent Overshoot (%OS)	Steady-state Error (e_{ss})	Is it a stable system? (Yes/No)	Type of the System (Under/Over/ Critically Damped System?)
0.1	0.001							
0.1	0.01							
0.1	0.05							
0.1	0.1							
0.1	0.3							
0.1	0.8							
0.1	1.5							
0.1	2.0							
0.1	3.0							
0.1	10.0							

Based on the information in the above table, please draw a conclusion.

3. PD controller

Manually move the arm of the motor to the position of “zero degree”. Please create a PD control system in Simulink, as indicated below.



Please build the above system, set the proportional gain and derivative gain according to Table 3, run the Simulink model, measure the response curve and complete the table:

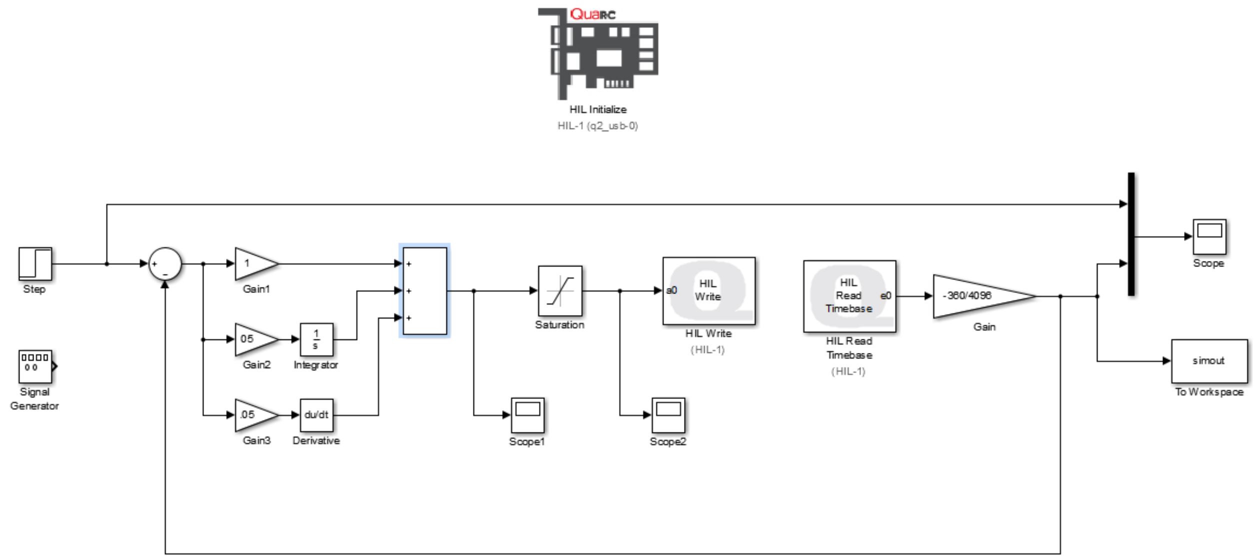
Table 3 Effects of Derivative Gain

Proportional Gain (k_p)	Derivative Gain (k_d)	Rise Time (T_r)	Peak Time (T_p)	Setting Time (T_s)	Percent Overshoot (%OS)	Steady-state Error (e_{ss})	Is it a stable system? (Yes/No)	Type of the System (Under/Over/ Critically Damped System?)
0.1	0.001							
0.1	0.005							
0.1	0.01							
0.1	0.02							
0.1	0.03							
0.1	0.05							
0.1	0.08							
0.1	0.1							
0.1	1							

Based on the information in the above table, please draw a conclusion.

4. PID controller

Manually move the arm of the motor to the position of "zero degree". Please create a PID control system in Simulink, as indicated below.



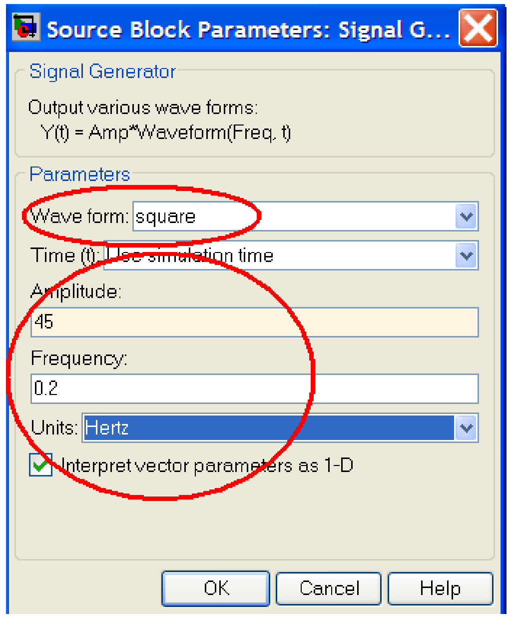
Please build the above system, and use trial and error and try to find the proportional gain, integral gain and derivative gain in a way that the system has the minimum possible overshoot, and settling time.
Show your trial and errors in Table 4.

Table 4 Effects of Derivative Gain

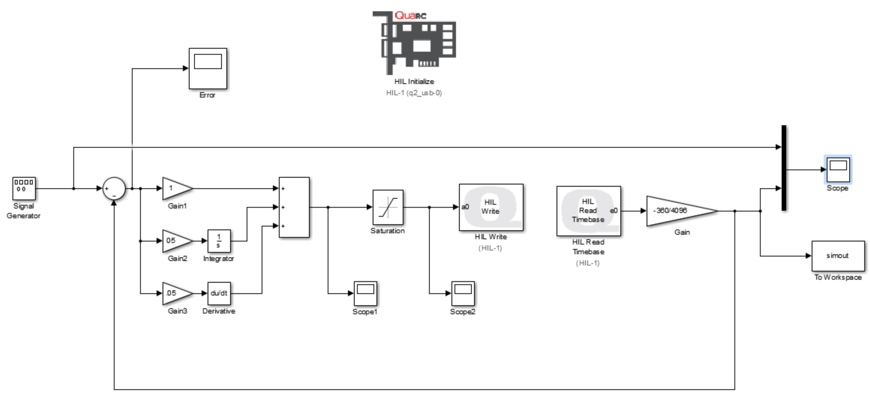
Proportional Gain (k_p)	Integral gain (k_i)	Derivative Gain (k_d)	Rise Time (T_r)	Peak Time (T_p)	Setting Time (T_s)	Percent Over shoot (%OS)	Steady -state Error (e_{ss})	Is it a stable system? (Yes/No)	Type of the System (Under/Over/ Critically Damped System?)

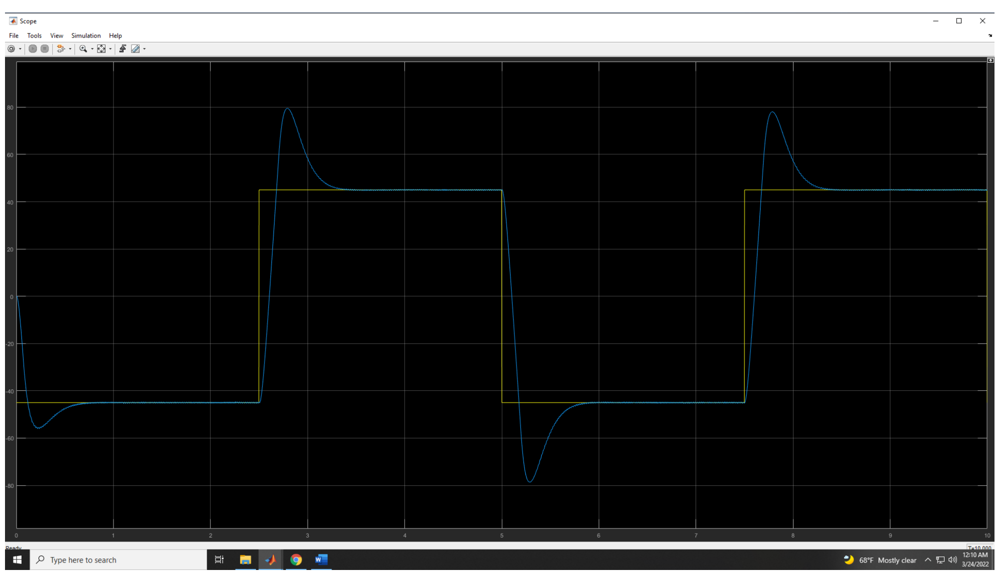
Based on the information in the above table, please draw a conclusion.

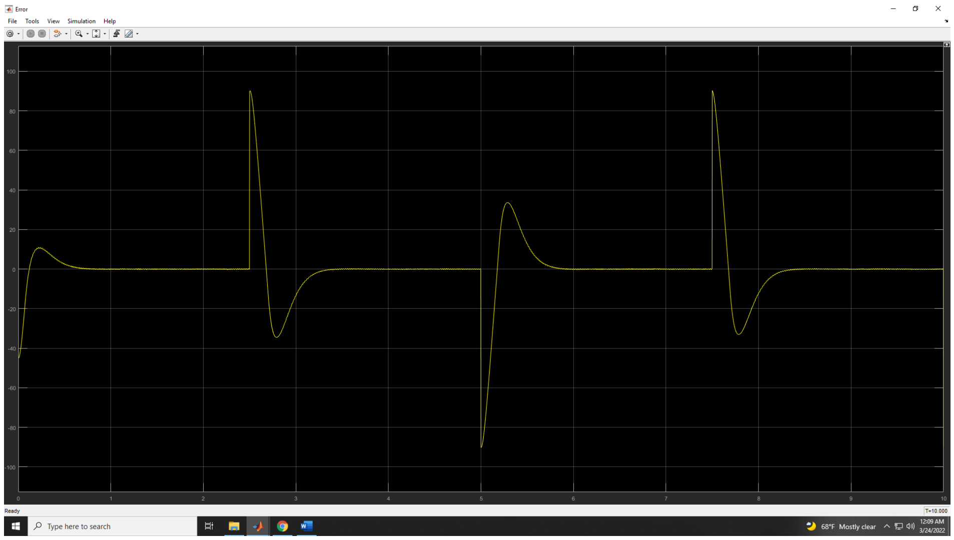
(24) Connect the signal generator block instead of step. Double click the “Signal Generator” block,, and set its properties below:



Simulate the response of the system for the square wave and plot the tracking error.







Lab report requirements

(1) The lab report should be an official technical report, including the information such as experiment objectives, introduction of the experimental system, experimental data and curves, and conclusions.
(2) For each controller (P, PI, PD or PID), you need to present its Simulink model, the response curves, the completed table, and draw a conclusion based on the data in the table.
(3) Each group just submits one lab report.