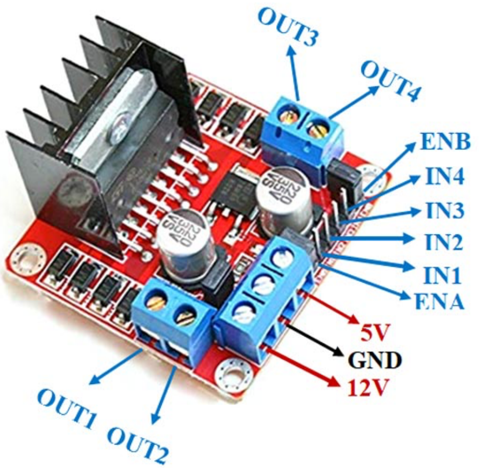
Louisiana Tech University

ELEN 479: Automatic Control Systems

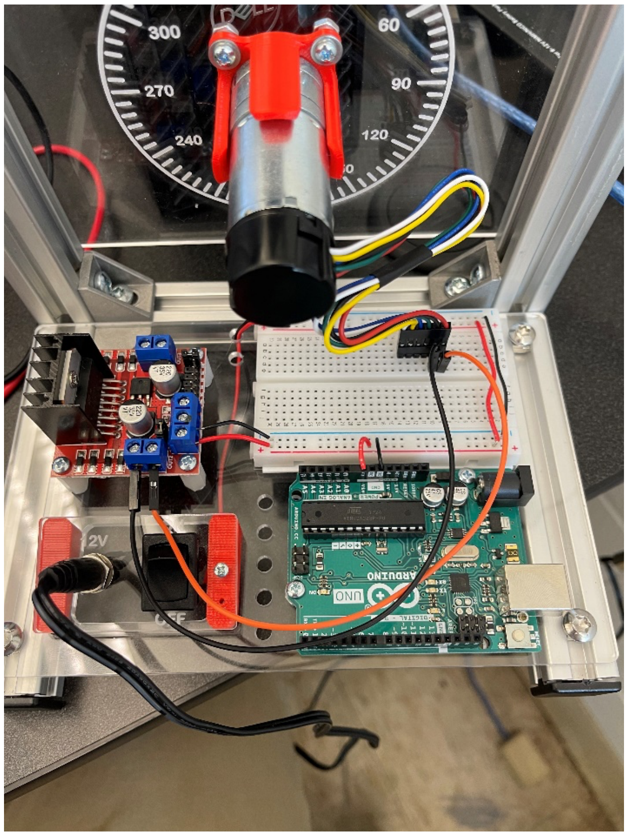
Lab 6: Motor Control with PID Controller

Task 1: Control motor speed and direction using Arduino.

1. L298N Motor Driver



2. Connect batteries power line:
    a. OUT1 – Black line
    b. OUT2 – Red line



3. Connect the motor driver with the Arduino
   In1 – Pin 4
   In2 – Pin 5
   ENA – Pin 9 (Remove the jumper in ENA pin, otherwise, the speed control won't work).

4. Connect the Encoder with Arduino
   Encoder is the device to measure how fast the motor spins, i.e., measure motor's speed. There are four pins (white, yellow, blue, and green) with the encoder. From left to right, they are connected to Arduino as follows:
   White – Pin 3
   Yellow – Pin 2
   Blue – 5V
   Green – Ground

5. Write the code to control the motor.
   a. Open "SpeedDirectionTest.ino"
   b. Connect Arduino with Computer
   c. Upload the program
   d. Try different values in "pwr" (50, 100, 200). What do you observe?
6. Write the code to read motor speed.
   a. How does the encoder measure speed?
      i. Method 1
      ii. Method 2
   b. Open "EncoderTest.ino"
   c. Connect Arduino with Computer
   d. Upload the program
   e. Open Serial Plotter in the right upper corner in the Arduino.
   f. In the plot, Value 1 (blue line) is the speed measurement from the first method, and Value 2 (red line) is the speed measurement from the second method. You can see that the first method only produces discrete integers, while the second method's measurement is smoother.
7. Add Low-pass filter
   a. We can add a Low-pass filter to filter out high frequency components in speed measurements (method 1).
   b. Open "AddLowPassFilter.ino"
   c. Connect Arduino with Computer
   d. Upload the program
   e. Open Serial Plotter in the right upper corner in the Arduino.
   f. In the plot, Value 1 (blue line) is the raw speed measurements from Method 1, and Value 2 (red line) is the result after the low-pass filter. We can see that the plot is smoother after the low-pass filter.
8. Add PID Controller to control motor speed
   a. PID Control 1
   b. PID Control 2

c. PID Control 3
d. Open “PController.ino”
e. Locate to line 65, which sets a target speed (target value is 100)
f. In line 69, the gain of the proportional controller is defined. (You can change this value from 0 – 5) and run the code to see what will happen.
g. Upload the program and open Serial Plotter.
h. In the plot, we can see that the actual speed is not equal to the target one.
i. Change kp in line 69 to 40, see what will happen.
    i. When kp is 40, the voltage generated to control the motor is too large, causing the motor to enter a cycle of overshooting and undershooting.
j. Now, let’s try to add the integral controller.
k. Open “PIController.ino”
l. Locate to line 70, we define the integral control gain as ki, and in line 73, the control output is the sum of proportional and integral controller.
m. Try different values of kp and ki to make the motor speed close to the target speed.