

Module-04: Speed control of DC Motor

In Module 03, we designed a water level control system. Motor M1 was used to pump out water from the container and motor M2 was used for sluice gate. In this module, we want to improve the system by providing control over the speed at which the sluice gate will operate. Note that the speed of rotation of motors is directly related to the input voltage. The higher the input voltage, the higher will be the rotational speed of the motor. Of course, the input voltage should not be more than the recommended operating voltage of the motor. Thus, in order to control the speed of a DC motor, we will need to provide a variable voltage to the DC motor. Usually, a pulse width modulation technique is used to generate a variable DC voltage from constant DC voltage source. For more on this check: <https://microcontrollerslab.com/dc-motor-l293d-motor-driver-ic-arduino-tutorial/>

Objectives:

1. In order to measure the speed of M2, interface a speed measuring sensor and encoder to M2. Use an LCD module to display the rpm measured using speed measuring sensor and encoder.
2. Assuming that the maximum rpm of M2 is r at 9V, we provide the desired speed for M2 using a DIP switch. As indicated in the table below, 4 different reference speeds can be given using the DIP switch:

State	Reference speed
0	90% of r
1	80% of r
2	70% of r
3	60% of r

Display the desired rpm in the LCD as well. (You might have already used two of the DIP switches in Module 03, use the other two for providing the desired speed.)

3. Design a PID controller to attain the desired rpm. While tuning the controller gains first start with only proportional controller, i.e, keep $K_I = 0$ and $K_d = 0$. Increase K_p and check the output using serial port plotter of the IDE. Based on your requirement, whether you want steady state or transient performance improvement change K_I and K_d value.
4. Design the entire system of Module 3, replacing the ON-OFF controller for M2 with the newly designed PID controller.

Points to note:

- Note that the reading from the speed measuring grove might not be steady, hence you might have to use a filter to clean the data. A 1nF capacitor should help you do that.
- Implement the speed control algorithm first and after that integrate the algorithm with Module 03. One way of doing that is writing a new code for Module 04 and executing it. This way you won't have to disturb any of the circuit used in Module 03.
- Given that you will be adding LCD modules, speed measuring sensor, etc to Module 03, you will require a lots of pins of Arduino. You can also use the analog pins (A0-A5) as digital output/input.
- Use the serial monitor of Arduino to graphically plot the desired speed and measured speed versus time.

Pre-lab work:

1. Read about controlling speed of a DC motor using PWM and L293D module.
(see <https://microcontrollerslab.com/dc-motor-l293d-motor-driver-ic-arduino-tutorial/>).
2. Read about interfacing LCD to Arduino Uno.
(see <https://www.arduino.cc/en/Tutorial/LibraryExamples/HelloWorld>)
3. Read about serial port monitoring and plotting.
(see <https://docs.arduino.cc/software/ide-v2/tutorials/ide-v2-serial-plotter>)

Report Guidelines: The report to be submitted must have the following:

1. Objectives of the module.
2. Circuit diagram with proper labelling.
3. Block diagram of the system designed.
4. Flowchart of the algorithm.
5. Observations, if any.
 - Snap shots of the serial monitor needs to be submitted with your observations. You have to show how is the performance of the system effected for different values of K_p , K_I , and K_d .
 - Plot of the output from the encoder with and without the filter.
6. Answers to the questions below.

Answer the following:

1. Draw the block diagram of the closed loop system you have implemented in this module. What is the set point of your system?

2. Why did you use a closed loop system to implement speed control? Cant you use an open loop control, i.e., provide the required PWM at the input of the motor and get the desired rpm of the motor?
3. How did you tune the controller gains?
4. What is your observation when the value of K_p is increased keeping K_I and K_d zero? Can you relate it to the theory you have studied in Control Theory course?
5. What kind of filter have you used to clean the data recorded (High pass, Low pass, etc)? Based on the output plot from the encoder, why do you think the filter helped in cleaning the data?