EENG-340 Intro to Simple Motors: DRAFT

February 27, 2024

Become familiar with DC Motor control and open loop servo motors.

Pre-Lab Plan:
1 Requirements
2 Servo Motors, PWM,
3 Stepper Motor, IC driver
4 Small DC Motor, IC driver
5 Large DC Motor Control, gate driver
4 Implementation

Requirements

This project has multiple goals all about working with a few different types of motors and the basics of simple motor control (See Table 1). We'll be using a lightweight servo motor (9g A009 or similar), to get a sense of how they function, a small stepper motor, a small low-torque DC motor, and finally a much larger DC motor that will require a separate, discrete, MOSFET power transistors driven by an IC designed for that purpose. You have freedom to invent any input method you like to drive the motor (e.g., potentiometer, digital encoder, etc.).

Update pos/speed Frequency	3Hz or more, if possible
Angle Control	(0-180°$\pm$10°)
Speed Control	DC (FS-CCW $\rightarrow$ FS-CW)

Table 1: Requirements for Simple Motor Control: Based on the distance measured control either the angle (Servo, or Stepper) or the speed (DC Motors)

You can choose any input you like as a method of controlling the motor, but the actual control of the motor must be handled in either hardware, or an interrupt routine or some combination of the two. The main program loop must not be used, other than for initialization. Based on the input the motor should respond with an angular displacement (servo motor, stepper motor: 0-180°$\pm$ 10°), or a speed (full speed CCW to full speed CW for the DC motors). You'll need to estimate the servo and stepper motor angles, and you should come up with some way to estimate the continuous motor speeds, but we won't try to control the speed in this lab.

Servo Motors, PWM control

A servo motor employs a control signal consisting of a 50 Hz PWM pulse train with a variable pulse width between 1.0 ms and 2.0 ms corresponding to an angular position between 0-180°.

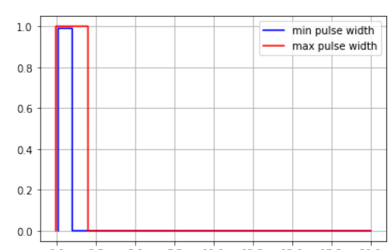

Figure 1: Range of pulse-widths used for servo control.

Depending on the sensor you select for the distance measurement, you may see a lot of noise in the distance data. It's a good idea to digitally filter those measurements so that the servo control signal is relatively smooth. You can use a digital filter like the one we developed for the last lab in the Electronics course, or you can do something quite a lot simpler (simple average of the last N readings using a circular buffer). Before connecting the distance sensor, control the motor with known PWM pulses to evaluate the reproducibility and accuracy of the servo. You'll find these are not expensive servos, so they're accuracy is limited. How limited?

Stepper Motor, IC driver

Next, use the same basic setup but controlling a stepper motor. We'll use an IC controller for this small (12V) stepper. The concept is exactly the same as the previous case except now with a stepper you'll be keeping track of the current position by sending a specific number of steps in either the CCW or CW direction. You'll notice that there are advantages and disadvantages to stepper motors relative to servos. Again, you'll want to estimate the angular accuracy of the stepper before connecting to the distance measurement.

Small DC Motor, IC driver

Next, we'll control a small DC motor using an IC H-bridge circuit. The idea here is similar to the previous two sections, but now we'll be controlling motor speed rather than position. Note that we'll need to come up with some way to *measure* the motor speed. The PWM signal for a DC motor is much less constrained than a servo. The idea is to vary the PWM duty cycle to adjust the speed of the motor by energizing the coils only a fraction of the time.

DC Motor Control, gate driver

Lastly, we'll control a much larger motor using discrete MOSFETs and an IC gate driver (several of these, e.g., irs2453ds). The main issues here are MOSFET and motor protection, power budgeting, heat-sinking, and generally dealing with much larger currents and voltages safely. Having said that, the requirements and constraints are very similar. Also, we won't be loading the motor much in this lab, so the actual power used will be fairly small even though, under load, it could be quite large.

Implementation

The details of implementation are up to you. You need to document your design planning, simulation, and testing carefully and completely.