

Lab 6: Deadband

Overview

This lab has three main learning objectives:

1. Students will learn what deadband is.
2. Students will learn how to investigate and quantify deadband.
3. Students will learn to compensate for deadband in feedback control of a DC motor.

In order to complete the lab, teams have four main tasks:

1. Program an Arduino to run tests to determine steady-state rotational speed vs. PWM command while gradually increasing or decreasing the PWM duty cycle
2. Generate an experimental graph of steady-state rotational speed vs. PWM command
3. Create a deadband compensation algorithm
4. Investigate how inverse deadband works with proportional control

Pre-Lab Assignment

- add something next year; this lab is coming out too late for much pre-lab work

Background

Friction in motors will prevent them from turning when small voltages are applied. This effect is normally known as ‘stiction’, a combination of the words static and friction. Friction is present in all motors. However, this effect is greater in lower cost motors. When a motor is connected to a mechanical system the friction may increase greatly. In most cases, the cost of a new design can be reduced by using less expensive motors and compensating for larger friction effects in software.

*Note that this lab was adapted from material by Nael Barakat and Amy Lenz

Motor internal friction consists of two types: static and kinetic. Static friction, ‘stiction’, will prevent initial motion. If the motor supplies enough torque to break free and start turning, the kinetic friction will provide a roughly constant friction torque. Figure 1 shows an example of how this friction affects the relationship between the voltage applied to a motor and the resulting velocity. At very low voltages the torque is too low to overcome the stiction. The region where the applied voltage has no effect is called the deadband. Once the applied voltage exceeds the deadband limits the motor will start to turn. While turning, the motor experiences kinetic friction, damping and other effects. If the voltage supplied to the motor is slowly decreased it will eventually ‘stick’. However, the sticking force will be lower than the static friction force. This alternate path is shown in the figure as a heavy dashed line.

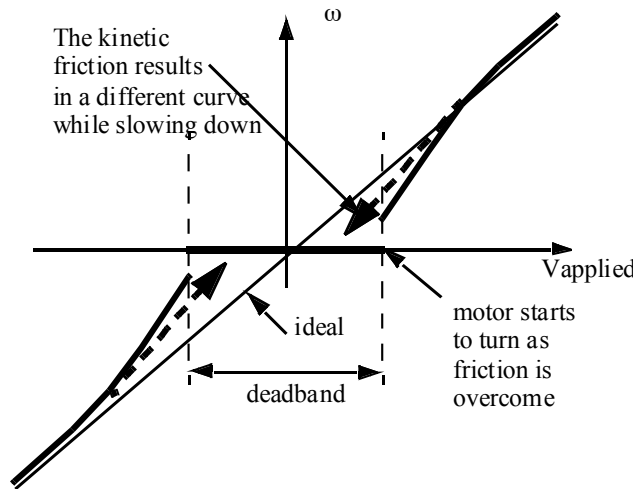


Figure 1: Graphical representation of motor deadband for a bidirectional motion.

There are a number of methods for compensating for a non-linear deadband in a system. A simple method is shown in Figure 2. This method involves finding the voltages where the motor breaks free in the positive and negative directions. These are used in a program to change the wanted output voltage, to an adjusted output voltage.

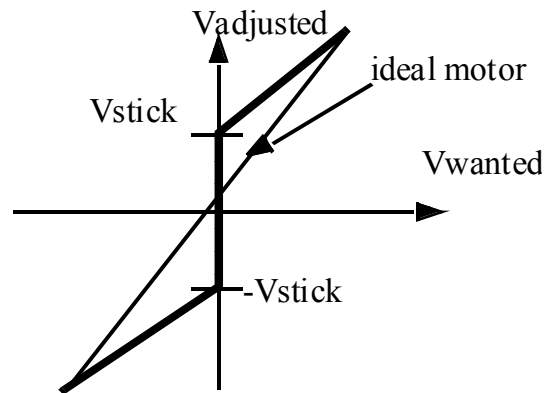


Figure 2: Possible deadband compensation scheme for a bidirectional motor

In this lab, you will modify code from the previous lab used to control a DC motor. By the end of this lab, your code should include deadband compensation as well as saturation. Note that the deadband may not be the same in the positive and negative directions.

Hardware Description

This lab will use the same motor/H-bridge systems used in the previous lab. The H-bridge circuit shown in Figure 3 can be used to drive a motor in two directions. As before, the transistors are used to switch current flow. In this case applying a voltage to the A1 and A2 terminals would allow voltage to flow left-to-right through the motor. Turning on inputs B1 and B2 would cause current to flow right-to-left through the motor. When the inputs are controlled using a PWM signal the effective motor voltage can be varied.

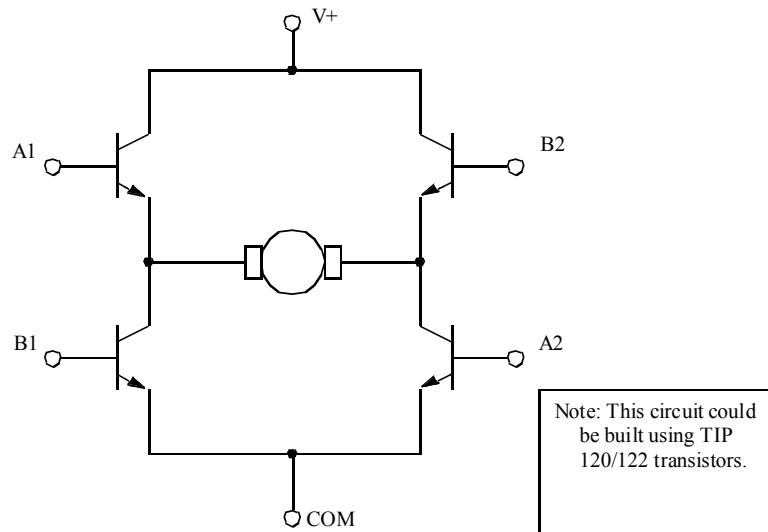


Figure 3: Schematic of the H-bridge circuit for motor control

When controlling an H-bridge some extra logic circuitry is commonly used; the main purpose for this circuitry is to protect the H-bridge. In this lab an integrated circuit that already contains protection circuitry will be used. This is the same motor controller (H-Bridge) board used in the previous lab. The PWM signal is sent to the enable pin (EnA or EnB) while direction is determined by pins IN1 and IN2 or IN3 and IN4.

Deadband Investigation

The main goal of this portion of the lab is to experimentally generate a graph that like that shown in Figure 1. Note that the x axis is commanded PWM output and the y axis is steady-state rotational velocity. Note that there should be different curves for increasing the decreasing the input. For each open-loop input value, you will need to find the steady-state rotational velocity.

Modify your open-loop motor code from Lab 5, Part 1 to generate the graph for $\dot{\theta}_{ss}$ vs. PWM

command. Your code should make it easy to run multiple tests and overlay the data. You should do this by stepping the PWM input up and down in a for loop. The resulting data should be sent to Python somehow to generate the graph.

Deadband Compensation

Once you have data and a reasonable model for the deadband of the motor, develop code that compensates for the deadband as illustrated in Figure 2. Make sure that your deadband compensation can easily be turned on and off so that you can compare the response with and without it. Your compensation scheme will likely need to consider where the system is currently operating on Figure 1.

When you believe you have a good deadband compensation algorithm, re-run your tests for $\dot{\theta}_{ss}$ vs adjusted PWM in a for loop and see if the curve shows that the adjustment is working properly.

Proportional Control with Deadband Compensation

Setup your Arduino to run proportional control tests like in the previous lab, but make it possible to turn deadband compensation on and off for each test. Choose values for K_p that lead to step responses that are lightly damped, moderately damped, and overdamped. Then run tests for several different desiring stopping points θ_d with and without deadband compensation.

- How does your deadband compensation affect the system response for small and large values of θ_d ?
- How does the deadband compensation affect the system response for large and small values of K_p ?

Refinement

Refine your deadband compensation scheme as necessary to lead to the best possible proportional control step response over a range of $\dot{\theta}_{ss}$ values.

Report Specifications

- Discuss the code that allowed you to generate Figure 1
- Discuss your experimental Figure 1
 - What does it mean?
 - How do you model/quantify it?
 - Any additional observations?

- Discuss your deadband compensation scheme
 - What did you intend for it to do?
 - Are there any novel features?
 - How did you code it up?
- Present a graph of open-loop, adjusted PWM vs $\dot{\theta}_{ss}$ using your deadband compensation algorithm
 - Is the deadband compensation working correctly?
- Discuss the effectiveness of your deadband compensation scheme when combined with proportional control
 - Did it improve the system performance?
 - * In what ways?
 - Did you refine it? If so, how?
 - Are there any drawbacks to using your deadband compensation scheme?