Project 7: PSoC: PID control & motors

Abstract

The purpose of this project was to build a circuit that controls a motor using a sensor. We were to use some kind of sensor to either control the speed, direction, or angle of the motor. I used a temperature sensor to control the sense of rotation either clockwise or anticlockwise of a motor. I used a H-Bridge chip, L293D, to control the DC motor. I set the circuit such that if the sensor reads a temperature greater than 70, it turns anticlockwise and if it reads a temperature less than 70 it turns clockwise.

Description

The L293D is a monolithic integrated high voltage, high current four channel driver designed to accept standard DTL or TTL logic levels and drive inductive loads (such as relays solenoides, DC and stepping motors) and switching power transistors. It is assembled in a 16 lead plastic packaage which has 4 center pins connected together. The chip has diodes built in it that protects it from inductive kicks from the motor. Below is a diagram of the bridge

Hbridge.jpg

One can set the motor to move forward by setting a pin high and another low and one can run the motor in reverse by reversing the high and low or either pins.

Schematic and Discussion

David.PNG

Arduino Code

```
float temp;
int tempPin = 0;

void setup()
{
  pinMode(13, OUTPUT);
  pinMode(11, OUTPUT);
  pinMode(10, OUTPUT);
  digitalWrite(13, HIGH);//set this pin to HIGH to enable motor driver

  Serial.begin(9600);
}

void loop()
{
  temp = analogRead(tempPin);
  // read analog volt from sensor and save to variable temp
  temp = temp * 0.48828125;
  // convert the analog volt to its temperature equivalent

  Serial.print("Temperature = ");
  Serial.print(temp); // display temperature value
  Serial.println("C");

  if (temp > 70) {
  digitalWrite(11, HIGH);
  digitalWrite(10, LOW);
  delay(2000);
  }
  if (temp < 70) {
  digitalWrite(11, LOW);
  digitalWrite(10, HIGH);
  delay(2000);
  }
```

}

Results

We observed that when the the sensor reads a temperature greater than 70, it turns anticlockwise and if it reads a temperature less than 70 it turns clockwise.

Conclusion

The purpose of this lab is to control a motor using a sensor. I used a temperature sensor to control the direction of the motor. If the temperature is greater than 70 it turns anticlockwise and if the temperature is less than 70 it turns clockwise. I used a H-Bridge chip, L293D, to communicate with the DC motor

Appendix

https://www.tinkercad.com/things/5XLkhfsqqHC-spectacular-uusam-krunk/editel?sharecode=bu60_iIGcdx46DfLI8A4SYoOeonQuuXfmAcExpj6xm4

If link doesn't work

https://www.tinkercad.com/things/5XLkhfsqqHC-temperature-sensor-and-dc-motor/editel?sharecode=bu60_iIGcdx46DfLI8A4SYoOeonQuuXfmAcExpj6xm4

Statistics Exercise

Exercise 1

σ_{height} = 0.5

σ_{width} = 0.5

Area = height * width

$$A = h^1 * w^1$$
$$\left(\frac{\sigma_d}{d}\right)^2 = \left(1\frac{\sigma_h}{h}\right)^2 + \left(1\frac{\sigma_w}{w}\right)^2$$

In [26]:

```
%matplotlib inline
import pandas as pd
import matplotlib.pyplot as pl
import numpy as np
from scipy.integrate import quad
from scipy.optimize import curve_fit
h = 1
w = 1
sigma_h = 0.005
sigma_w = 0.005
a = 1
sigma_a = a*np.sqrt((sigma_h/h)**2 + (sigma_w/w)**2)
print("Uncertainty in area : %3.3f m" %sigma_a)
```

```
Uncertainty in area : 0.007 m
```

Exercise 2

In [25]:

```
x = np.array([1, 2, 3, 4, 5, 6])
y = np.array([3, 5, 7, 9, 11, 13])
sigma_x = 0.1
sigma_y = 0.2

def linear(x, m, b):
    return (m * x) + b

par, cov = curve_fit(linear, x, y)
m, b = par
N = 1000      # number of samples to take
bList = []
mList = []
for i in range(N):
    xMC = x + sigma_x * np.random.normal(size=len(x))
    yMC = y + sigma_y * np.random.normal(size=len(y))
    par, cov = curve_fit(linear, xMC, yMC)
    mList.append(par[0])
    bList.append(par[1])

marr=np.array(mList)
barr=np.array(bList)

mAvg = marr.mean()
bAvg = barr.mean()
sigm = marr.std(ddof=1)
sigb = barr.std(ddof=1)

print("m= %5.3f +/- %5.3f ( %5.3f, %5.3f )" % (m, sigm, m+sigm, m-sigm ))
print("b= %5.3f +/- %5.3f ( %5.3f %5.3f )" % (b, sigb, b+sigb, b-sigb ))
```

```
m= 2.000 +/- 0.066 ( 2.066, 1.934 )
b= 1.000 +/- 0.258 ( 1.258 0.742 )
```