EECS 106B : Lab #4

Spring 2018

Yichi Zhang, Jingjun Liu, Jiarong Li
27005775, 3033483513, 3033483511

May 27, 2018

Section 1

Video

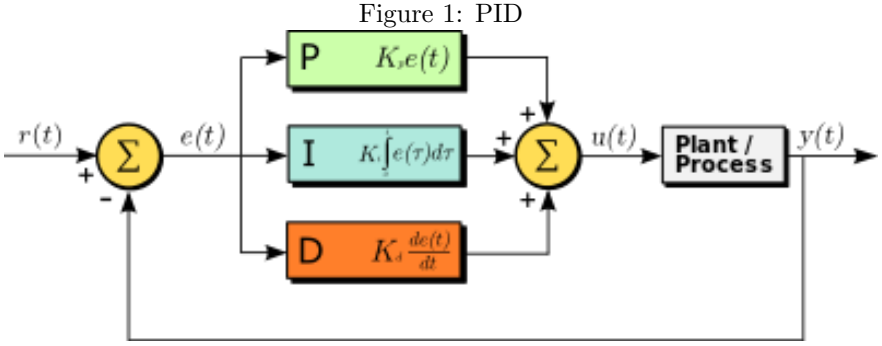
https://youtu.be/bTTMlNWNM5g

Section 2
Methods

Controller

We use PID controller package from official website of Arduino and tuning parameters $Kp, Ki\&Kd$ in steps.

Figure 1: PID



Section 3
Difficulties

1. typo error in starter code

In the starter code, the sensor port is A0, but in circuit it is port A1.

2. Difficulties in adjusting parameters

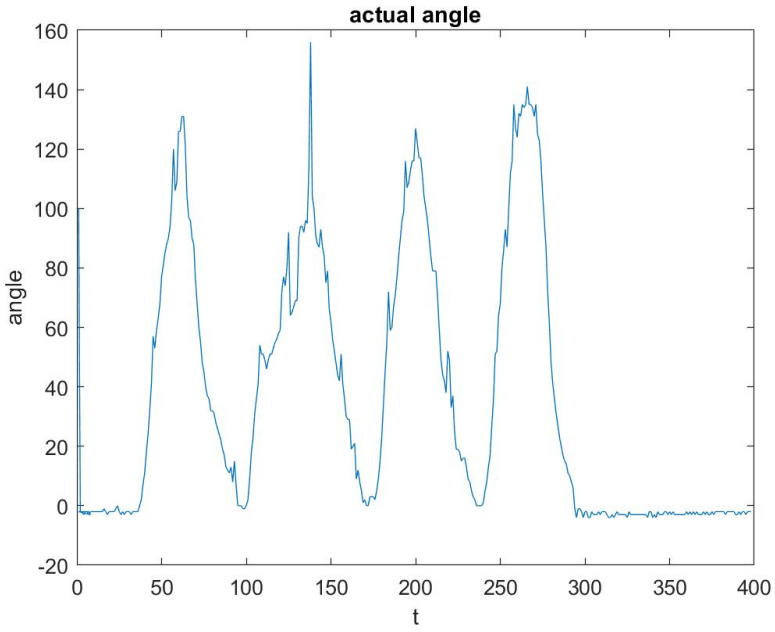
We spent a lot of time adjusting three parameters Kp, Ki and Kd for both controllers. We found that Kp should be around 50 so that it has enough 'force' to lift the finger. Also, Kp affects the smoothness of this motion. So we can set it very large, over 100. Ki has little affects on the control.

Section 4

Flex Sensor

In testing the behavior of Flex Sensor, we bend the finger periodically by hand and record the angle it reads. As shown in the plot, the sensor is a little noisy, and the plot is not smooth enough.

Figure 2: PID

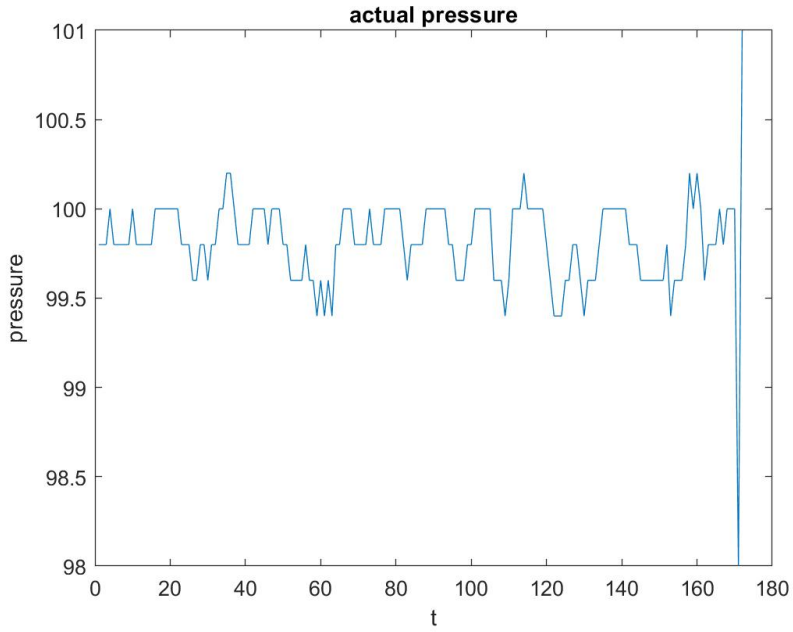


Section 5

Pressure sensor

In testing the behavior of pressure Sensor, we used the same method as testing flex sensor. Basically the pressure range has a smaller scale compared with angle. So the noise is still huge even though we compute the average value every five figures.

Figure 3: PID



Section 6
Results

The step response are as follows.

Figure 4: Step response of angle controller

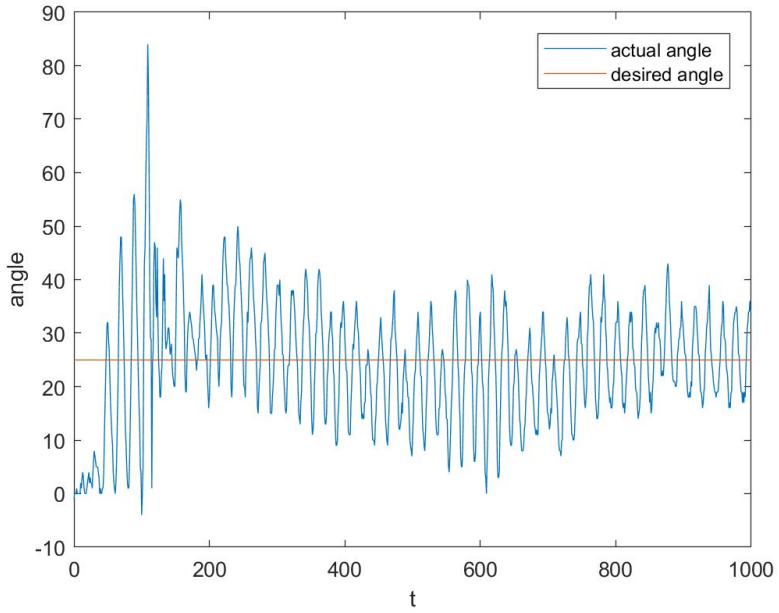
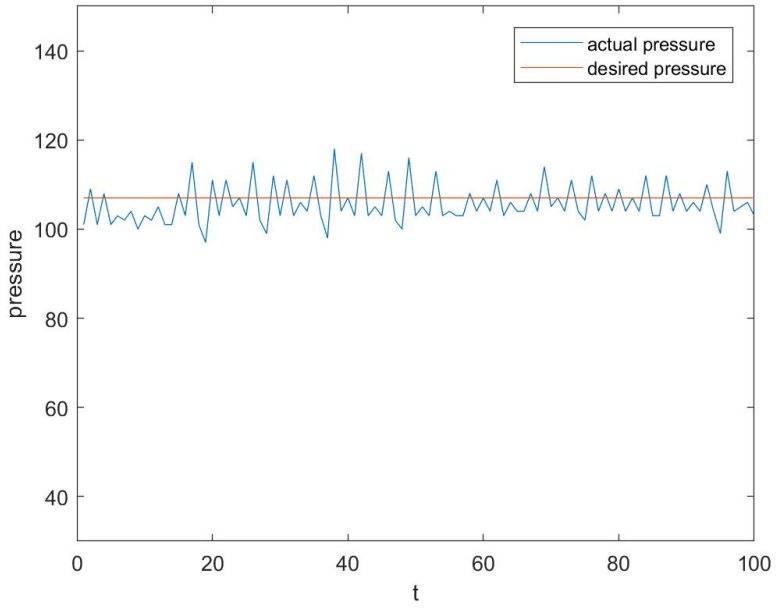


Figure 5: Step response of pressure controller



The noise is not small in these sensors.

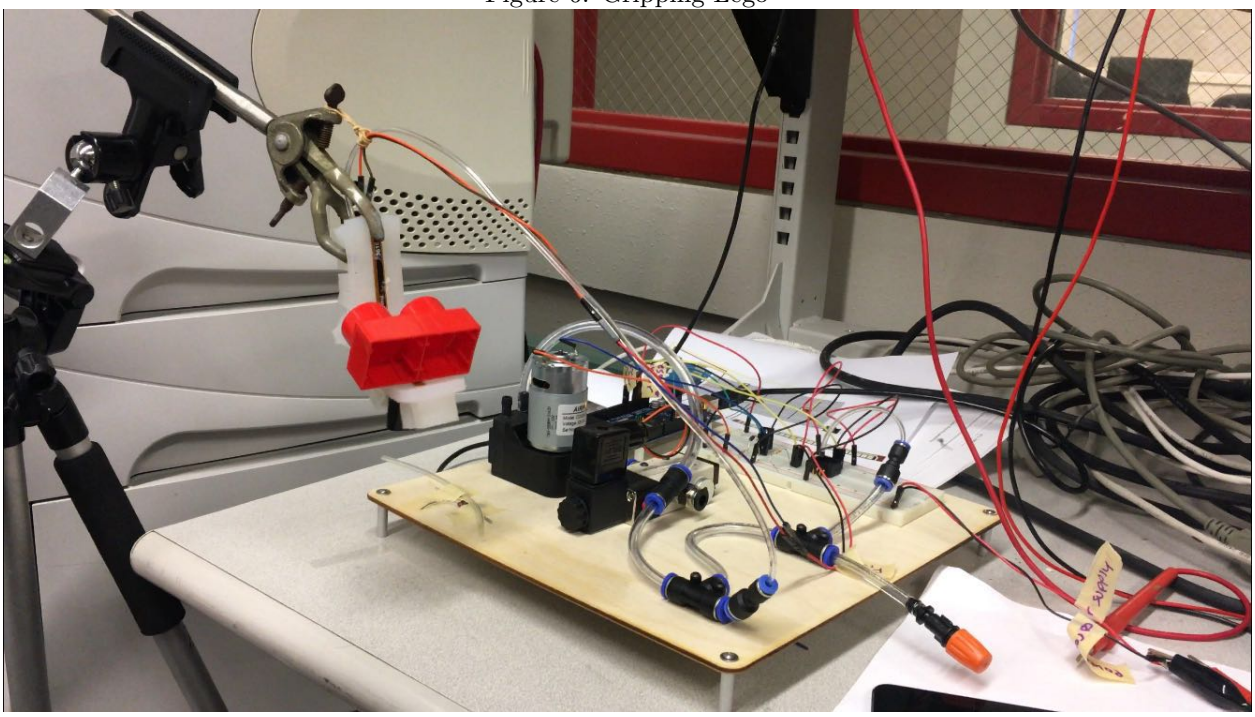
For flex sensor, the error comes from temperature in environment, soft material elastic deformation uncertainty. For pressure sensor, the error comes from air leakage on the tube and system and temperature. It influence our PID controller to get a better response. We try to deal with it by not using each sensor data but average several data in a range of time.

Section 7

Soft fingers Examples

The rigid finger can not pick up materials that with strange shape or quite low surface friction, such as ocean creatures and smooth ball. Here we use it to pick up an Lego. It is also in the video.

Figure 6: Gripping Lego



Section 8

Lab documentation

The lab document is written in details! Every thing is awesome except the resistance detach from the soft materials. Thank you for all your help during the whole semester.

Section 9

Code

1. pressure control

```
%% pressure_control.ino %%
#include <PID_v1.h>
#define sensor A0
const int FLEX_PIN = A1;
const float VCC = 4.98;
const float R_DIV = 3300.0;
const float STRAIGHT_RESISTANCE = 1170.0;
const float BEND_RESISTANCE = 2950.0;
int incomingByte = 0;
double thdes, pressuredes = 0.0;
double th = 0.0;
double pressure;
double e1,e2;
double Kp1 = 35, Ki1 = 1.0, Kd1 = 300.0;
double Kp2 = 35, Ki2 = 1.0, Kd2 = 300.0;
int flexADC;
float flexV;
float flexR;
double PWM = 0.0;
int a = 1;
PID myPID2(&pressure, &PWM, &pressuredes, Kp2, Ki2, Kd2, DIRECT);

void setup(){
pinMode(FLEX_PIN, INPUT);
pinMode(3, OUTPUT);
// set mosfet gate pin to an output
Serial.begin(9600);
pressuredes = 110;
// opens serial port, sets data rate to 9600 bps
Serial.println("[");
}

void loop(){
flexADC = analogRead(FLEX_PIN);
flexV = flexADC * VCC / 1023.0;
flexR = R_DIV * (VCC / flexV - 1.0);
pressure = analogRead(sensor);

myPID2.SetMode(AUTOMATIC);

myPID2.SetOutputLimits(0,100);

myPID2.Compute();
th = map(flexR, STRAIGHT_RESISTANCE, BEND_RESISTANCE,0, 90.0);
```

Section 9 continued on next page...

```
//Serial.println("th: "+ String(th));
Serial.println();
Serial.println(String(a) + ",");
Serial.println(String(pressure) + ";");
//Serial.println(+ String(PWM));
a += 1;
analogWrite(3,PWM);
delay(100); //You may change this to be whatever frequency you want
}
```

2. angle control

```
%% angle_control.ino %%
#include <PID_v1.h>
#define sensor A0
const int FLEX_PIN = A1;
const float VCC = 4.98;
const float R_DIV = 3300.0;
const float STRAIGHT_RESISTANCE = 1170.0;
const float BEND_RESISTANCE = 2950.0;
int incomingByte = 0;
double thdes, pressuredes = 0.0;
double th = 0.0;
double pressure;
double e1,e2;
double Kp1 = 55.0, Ki1 = 10.0, Kd1 = 100.0;
int flexADC;
float flexV;
float flexR;
double PWM = 0.0;
int cnt = 0;
PID myPID1(&th, &PWM, &thdes, Kp1, Ki1, Kd1, DIRECT);

void setup(){
pinMode(FLEX_PIN, INPUT);
pinMode(3, OUTPUT);
// set mosfet gate pin to an output
Serial.begin(9600);
thdes = 25;
// opens serial port, sets data rate to 9600 bps
Serial.println("[");
}
  void stopSketch(void)
    {
     noInterrupts();
     while(1) {}
    }
void loop(){
```

```
flexADC = analogRead(FLEX_PIN);
flexV = flexADC * VCC / 1023.0;
flexR = R_DIV * (VCC / flexV - 1.0);

myPID1.SetMode(AUTOMATIC);

myPID1.SetOutputLimits(0,100);

myPID1.Compute();

th = map(flexR, STRAIGHT_RESISTANCE, BEND_RESISTANCE,0, 90.0);
//Serial.println("th: "+ String(th));
//Serial.println("PWM: " + String(PWM));
Serial.println(String(cnt) + ",");
Serial.print(String(th) + ";");
analogWrite(3,PWM);
cnt += 1;
delay(20); //You may change this to be whatever frequency you want
}
```

```
#include <PID_v1.h>
#define sensor A0
const int FLEX_PIN = A1;
const float VCC = 4.98;
const float R_DIV = 3300.0;
const float STRAIGHT_RESISTANCE = 1170.0;
const float BEND_RESISTANCE = 2950.0;
int incomingByte = 0;
double thdes, pressuredes = 0.0;
double th = 0.0;
double pressure;
double e1,e2;
double Kp1 = 35, Ki1 = 1.0, Kd1 = 300.0;
double Kp2 = 35, Ki2 = 1.0, Kd2 = 300.0;
int flexADC;
float flexV;
float flexR;
double PWM = 0.0;

PID myPID1(&th, &PWM, &thdes, Kp1, Ki1, Kd1, DIRECT);
PID myPID2(&pressure, &PWM, &pressuredes, Kp2, Ki2, Kd2, DIRECT);

void setup(){
pinMode(FLEX_PIN, INPUT);
pinMode(3, OUTPUT);
// set mosfet gate pin to an output
Serial.begin(9600);
thdes = 30;
pressuredes = 110;
// opens serial port, sets data rate to 9600 bps

}

void loop(){
```

Section 9 continued on next page...

```
flexADC = analogRead(FLEX_PIN);
flexV = flexADC * VCC / 1023.0;
flexR = R_DIV * (VCC / flexV - 1.0);
pressure = analogRead(sensor);
//e1 = thdes - th;
//e2 = pressuredes - pressure;
myPID1.SetMode(AUTOMATIC);
myPID2.SetMode(AUTOMATIC);
myPID1.SetOutputLimits(0,100);
myPID2.SetOutputLimits(0,100);
//myPID1.Compute();
myPID2.Compute();
th = map(flexR, STRAIGHT_RESISTANCE, BEND_RESISTANCE,0, 90.0);
Serial.println("th: "+ String(th));
Serial.println("pressure: "+ String(pressure));
Serial.println("PWM: " + String(PWM));
//Serial.println("e1: " + String(e1));
//Serial.println("e2: " + String(e2));
Serial.println();
analogWrite(3,PWM);
delay(10); //You may change this to be whatever frequency you want
}
```

Pump.cpp

```
#define sensor A0
int incomingByte = 0;
const int FLEX_PIN = A1;
const float VCC = 4.98;
const float R_DIV = 3300.0;
const float STRAIGHT_RESISTANCE = 1170.0;
const float BEND_RESISTANCE = 2950.0;

// Pressure sensor


void setup(){

Serial.begin(9600);
pinMode(FLEX_PIN, INPUT);
// opens serial port, sets data rate to 9600 bps
pinMode(3, OUTPUT);
// set mosfet gate pin to an output
}

void loop(){
   // put your main code here, to run repeatedly:
int flexADC = analogRead(FLEX_PIN);
float flexV = flexADC * VCC / 1023.0;
float flexR = R_DIV * (VCC / flexV - 1.0);
Serial.println("Resistance: " + String(flexR) + " ohms");
// Use the calculated resistance to estimate the sensor s
// bend angle:
float angle = map(flexR, STRAIGHT_RESISTANCE, BEND_RESISTANCE,0, 90.0);
Serial.println("Bend: " + String(angle) + " degrees");
Serial.println(analogRead(sensor)); // prints pressure sensor value
Serial.println();
Serial.print(angle);
analogWrite(100, incomingByte); //write the user entered value(0-254) to pwm outdput
delay(1000);
}
```

Section 9 continued on next page...

pressure_sensor.cpp