Universidad Nacional Autónoma de México
Facultad de Ingeniería

Sánchez Monterde Luis Antonio
Vizcayno García José Agustín
Grupo:4
Laboratorio: Bio-Robótica
Temas selectos de control y robótica

Práctica 2.- Conexión de Sensores a la Tarjeta de Arduino, Parte 1

Miercoles 12 de septiembre de 2018

Sensor	Pin	Descripción
contact	2	Push button
photord	3	Foto resistencia digital
temp	A0	Sensor de temperatura
photora	A1	Foto resistencia analógica
infrared	A2	Sensor infrarrojo

Tabla 1: Tabla de asignación

1. Objetivo

- Familiarizar al alumno con la conexión de diferentes sensores a la tarjeta Arduino.

2. Desarrollo

Se requiere crear un programa que lea cinco sensores, dos digitales y tres analógicos, se hacen las conexiones como indica el formato de la práctica y se les asigna a los pines resumidos en la tabla 1.

Para leer dichos sensores se realizo un programa en C, el programa se encuentra en el listado 1

```
#include<string.h> //para strcmp() compara cadenas

//Inicialización y declaración de variables
short int contacto = 0,
          foto_Resistencia_Digital = 0;
double  temperatura = 0,
        foto_Resistencia_Analogica = 0,
        infrarojo = 0;
char buff[20];  //Buffer serial;

void setup(){
  //Inicialización de puertos y pines
  pinMode(2,INPUT);
  pinMode(3,INPUT);
  pinMode(A0,INPUT);
  pinMode(A1,INPUT);
  pinMode(A2,INPUT);

  Serial.begin(9600);   //Habilita comunicación serial
  Serial.setTimeout(100); //100ms máximo para caracter nulo
}

void loop(){

}

void serialEvent(){ //Espera a que existan cambios en el puerto
  if(Serial.available()){ //Sí, hay datos en el buffer
    //Carga los datos en un buffer auxiliar
```

```
    Serial.readBytesUntil('\n', buff, 20);
    //Serial.println(buff);
    if(!strcmp(buff, "shs contact")){//Compara el buffer
      contacto = digitalRead(2); //guarda el valor del pushbotton
      Serial.print("contacto ");
      Serial.println(contacto);
    }else if(!strcmp(buff, "shs temp")){
      temperatura = analogRead(A0);
      Serial.print("Temperatura ");
      Serial.println(temperatura);
    }else if(!strcmp(buff, "shs photord")){
      foto_Resistencia_Digital = digitalRead(3);
      Serial.print("Fotoresistencia Digital ");
      Serial.println(foto_Resistencia_Digital);
    }else if(!strcmp(buff, "shs photora")){
      foto_Resistencia_Analogica = analogRead(A1);
      Serial.print("Foto Resistencia Analógica ");
      Serial.println(foto_Resistencia_Analogica);
    }else if(!strcmp(buff, "shs infrared")){
      infrarojo = analogRead(A2);
      Serial.print("Infrarojo ");
      Serial.println(infrarojo);
    }else{
        Serial.println("Ningun sensor con esa descripción");
    }
    memset(buff,0,sizeof(buff));//Limpia el buffer
  }
}
```

Listing 1: Programa para sensar

Los detalles importantes de este programa son pocos, el primero es que se opto por un manejo por eventos en lugar de uno en modo poleo, el segundo es que se utiliza un buffer auxiliar para leer la cadena de entrada. El programa funciona a la perfección por lo que solo queda agregar una rutina que pueda calcular la distancia más próxima a la verdadera de acuerdo a nuestro sensor.

Para encontrar el valor más cercano a la distancia verdadera, se utiliza el método de regresión lineal por mínimos cuadrados, en nuestro caso con R, para ello se toman muestras espaciadas, en nuestro caso muestras cada $5cm$ como en el listado 2

```
30      34.58
30      34.58
30      34.58
30      34.58
30      35.03
30      34.58
30      35.03
30      34.13
30      34.58
30      34.58
30      34.13
30      34.58
30      34.58
30      34.58
30      34.58
```

Distancia [cm]	Valor real del sensor
5	483
10	842
15	939
20	975
25	991
30	998

Tabla 2: Valores reales

```
30      35.03
30      35.03
30      34.58
.   .
.   .
.   .
```

Listing 2: Muestra de las muestras

las mediciones reales correspondientes a cada $5cm$ en función del ADC se muestran en la tabla 2 Con esta información se realizo la regresión lineal correspondiente para polinomios de primero a cuarto orden, con la ayuda del script en R proporcionado en clase,

```
#Se crea un función que actúa como el polinomio y se le nombra m3
m3=lm(formula = x ~ I(y)+I(y^2)+I(y^3))

#se calcula los coeficientes, con funciones propias de R
a0=m3$coef[1]
a1=m3$coef[2]
a2=m3$coef[3]
a3=m3$coef[4]

#Asignación simple de una función, con los coeficientes
    correspondientes
yc3 = a0 + a1*y + a2*y^2 + a3*y^3

#Imprime los coeficientes
cat("\ntercero a0= ",a0, " a1=", a1, " a2=", a2, "a3= ", a3)
```

Se tienen los coeficientes requeridos, para ver su comportamiento se grafican en las figuras 1 a 4,

Los coeficientes para cada polinomio son:

- Primer orden

$$\begin{aligned} a_0 &= 0.3117388 \\ a_1 &= 0.8640718 \end{aligned}$$

- Segundo orden

$$a_0 = -0.7533509$$

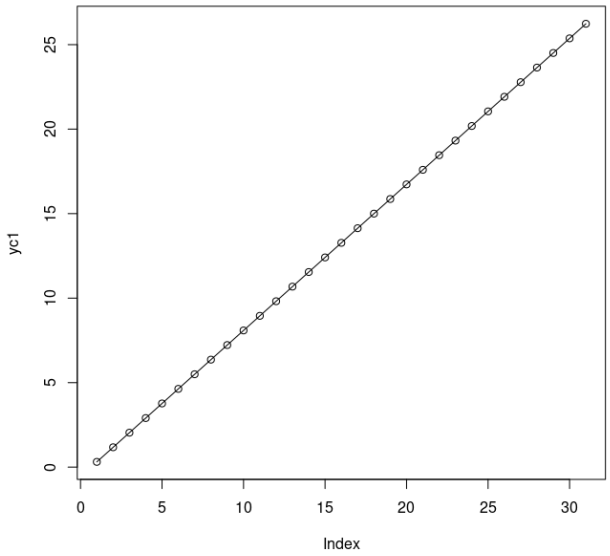

Figura 1: Polinomio de primer orden

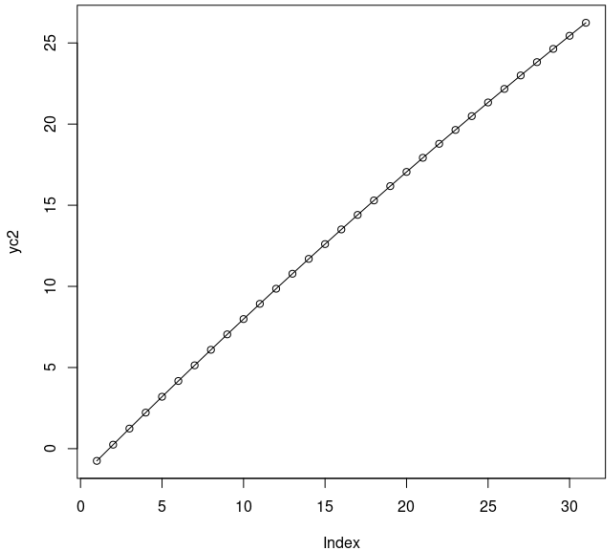

Figura 2: Polinomio de segundo orden

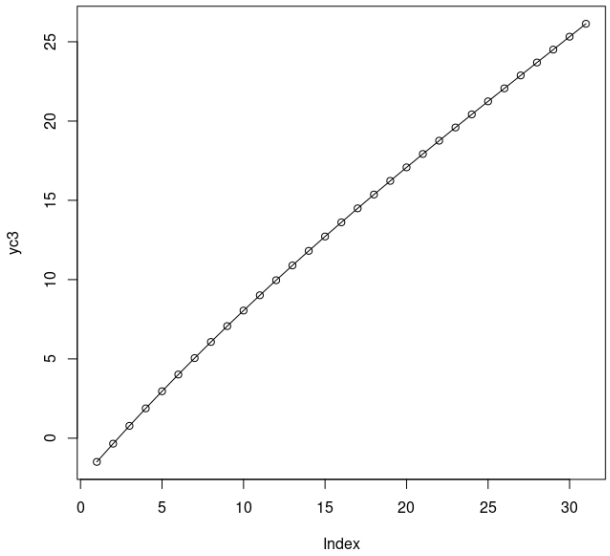

Figura 3: Polinomio de tercer orden

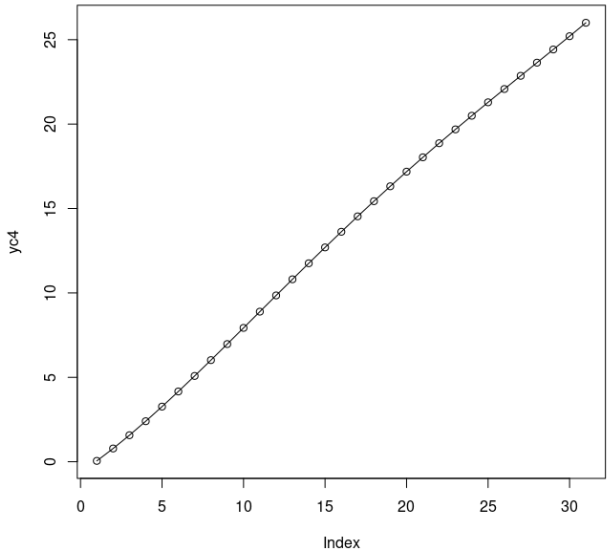

Figura 4: Polinomio de cuarto orden

$$
\begin{aligned}
a_1 &= 1.001657 \\
a_2 &= -0.003390901
\end{aligned}
$$

- Tercer orden

$$
\begin{aligned}
a_0 &= -1.494873 \\
a_1 &= 1.160523 \\
a_2 &= -0.01246657 \\
a_3 &= 0.0001494483
\end{aligned}
$$

- Cuarto orden

$$
\begin{aligned}
a_0 &= 0.0505364 \\
a_1 &= 0.7056092 \\
a_2 &= 0.029781 \\
a_3 &= -0.00138495 \\
a_4 &= 1.898157e-05
\end{aligned}
$$

por simple inspección de las figuras se ve que las dos primeras son prácticamente iguales, es obvio debido a que los coeficientes son muy parecidos, por otro lado los polinomios de tercer y cuarto orden se ven más "curvos", haciendo pruebas con cada uno de los polinomios, determinamos que el de cuarto orden es el más adecuado.

Se agrega al programa, reemplazando el código

```
infrarojo = analogRead(A2);
Serial.print(``Infrarojo '');
Serial.println(infrarojo);
```

por

```
//a0, a1, a2, a3, a4, son los coeficientes corregidos
x = 13*pow((analogRead(A2)*0.0048828115),-1);
y = a0+a1*x+a2*pow(x,2)+a3*pow(x,3)+a4*pow(x,4);
Serial.print("Infrarojo ");
Serial.println(y);
```