



**Instituto Politécnico Nacional
Unidad Profesional Interdisciplinaria de Ingeniería
Campus Zacatecas**



Ingeniería Mecatrónica

Tópicos avanzados de sensores

Grupo: 3MM7

Práctica 3. Giroscopio y acelerómetro

Alumnos:

Jesús Aldair García Saucedo

Jesús Alfredo Juárez Madera

Docente: Ramón Jaramillo Martínez

Investigación previa

- **Funcionamiento de acelerómetro y giroscopio analógicos.** El principio de funcionamiento de estos sensores es la variación de la distancia entre placas. Al variar la distancia se produce también una variación en capacitancia, que puede ser medida y convertida a voltaje.
- **MEMS.** Micro-electro-mechanical Systems por sus siglas en inglés son dispositivos que integran estructuras mecánicas y circuitos electrónicos diminutos para la formación de un sistema.
- **Funcionamiento del sensor MPU.**
 - *Funcionamiento.* Por medio de un sistema MEMS que utiliza masas suspendidas, el sensor mide las variaciones mecánicas, las convierte a señales de voltaje y las envía por el protocolo de comunicación I2C.
 - *Aplicaciones (por nombrar algunas):*
 - Detección de movimiento en teléfonos y relojes inteligentes.
 - Control de disparo de bolsa de aire.
 - Controles de videojuegos.
 - Posicionamiento de drones.
 - Control de brazos robóticos.
 - *Modo de operación.* El sensor MPU6050 puede ser utilizado junto a un microcontrolador a través de comunicación I2C desde los pines SDA y SCL.
- **Sensores de vibración.** Estos sensores miden las vibraciones de un sistema ya sea por desplazamiento, velocidad, aceleración, ruido o presión.

Desarrollo

1. Para obtener los datos de giro en los tres ejes, se diseñó un código que separa los valores para tres inclinaciones conocidas y posteriormente estos datos se agrupan en un solo vector.

```
for z = 1 : 3
    muestreoFLAG = 1
    for i = 1: 100
        data = readline(s);
        data1 = char(data);
        %Detectar '<'
        if data1(1) == '<'
            for k = 2 : 5
                pitch(k-1) = data1(k);
            end
            for k = 7 : 10
                roll(k-6) = data1(k);
            end
        end
        switch z
            case 1
                pitchVroll(i,1) = str2double(pitch);
                pitchVroll(i,2) = str2double(roll);
                normPitch(i,1) = inputNorm(pitchVroll(i,1), minValue, maxValue);
                normPitch(i,2) = inputNorm(pitchVroll(i,2), minValue, maxValue);
            case 2
                pitchVroll(i+100,1) = str2double(pitch);
                pitchVroll(i+100,2) = str2double(roll);
                normPitch(i+100,1) = inputNorm(pitchVroll(i+100,1), minValue, maxValue);
                normPitch(i+100,2) = inputNorm(pitchVroll(i+100,2), minValue, maxValue);
            case 3
                pitchVroll(i+200,1) = str2double(pitch);
                pitchVroll(i+200,2) = str2double(roll);
                normPitch(i+200,1) = inputNorm(pitchVroll(i+200,1), minValue, maxValue);
                normPitch(i+200,2) = inputNorm(pitchVroll(i+200,2), minValue, maxValue);
        end
        flush(s);
    end
end
```

Ilustración 1. Recolección de datos

Tras esto, se generó un vector con los valores para las entradas conocidas 1 1, -1 1 y -1 -1, para entrenar la red en base a los datos obtenidos.

```

T3 = [-1*ones(100,1), -1*ones(100,1)];
T2 = [1*ones(100,1), -1*ones(100,1)];
T1 = [ones(100,1), ones(100,1)];
R = [T1;T2;T3];

```

Ilustración 2. Vectores de entrenamiento

```

nodeHidden = 2;
outputs = 2;

fhidden = 'tansig';
foutput = 'tansig';

%% Training
[W1, b1, W2, b2, emedio] = neuralTrain(P, T, nodeHidden, fhidden, foutput);
figure
plot(emedio)

```

Ilustración 3. Entrenamiento

Estos vectores se introducen al código generado por el profesor para el entrenamiento (Ilustración 3) y con esto se obtiene los valores W y B necesarios para el programa de Arduino y así identificar de una manera más fácil la inclinación del sensor.

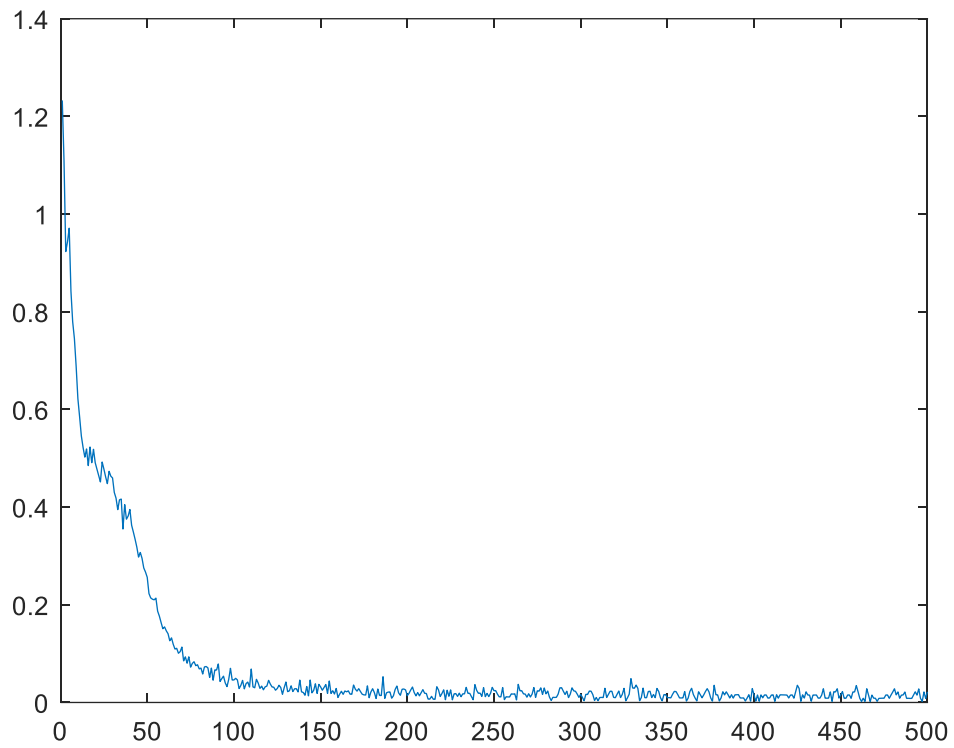


Ilustración 4. Error en las iteraciones.

```
float P[input][1];

float W1[node][input]={2.9984,2.2870},
                    {4.4617,0.2777}};

float b1[node][1]={-0.2880},
                {1.0383}};

float a1[node][1];

float W2[outputs][node]={1.1292,2.8251},
                        {-2.8956,2.9012}};
```

Ilustración 5. Valores W y B en Arduino.

```

if (a2[0][0]<=0 && a2[1][0]<=0){
  Serial.println("Atras");
}
else if(a2[0][0]<=0 && a2[1][0]>0){
  Serial.println("Izquierda");
}
else if(a2[0][0]>0 && a2[1][0]<=0){
  Serial.println("Derecha");
}
else
{
  Serial.println("Adelante");
}

delay(50);

```

Ilustración 6. Clasificación de giro

2. Ahora, para la generación de un código similar, pero sin el uso de la red neuronal, simplemente se diseña un rango entre los valores obtenidos por el sensor de manera empírica, es decir, dependiendo de la inclinación y si se está dentro de un margen.

```

void trama(int a, int b){
  if(a<500){
    Serial.println("izquierda");
  }else if(a>600){
    Serial.println("derecha");
  }else if(b<500){
    Serial.println("atrás");
  }else if(b>600){
    Serial.println("adelante");
  }else{
    Serial.println("");
  }
}

```

3. Para obtener la fuerza aplicada al sensor, se obtienen los datos de aceleración específicamente en un eje y con esto y la masa de la protoboard y el Arduino, se puede saber la fuerza aproximada aplicada para mover el sistema.

$$F = m * a = 0.118 * a$$

Donde 0.118 es la masa de la placa Arduino en la protoboard.

4. Para determinar la vibración de un objeto con el sensor MPU es necesario obtener las variaciones mínimas y detectar si estas tienden a repetirse, es decir, si se marca una oscilación entre los datos y así se podría saber si un objeto está en vibración.

Conclusiones

El uso de un sensor de este tipo es de bastante utilidad para comprender los fenómenos físicos de la aceleración, además de ayudar a reforzar conocimientos sobre el manejo de la comunicación I2C. Aunque fue complejo el poder separar los datos de utilidad, las operaciones con estos permiten resolver aplicaciones como la dirección de un robot móvil o un dron.

Referencias

<https://lastminuteengineers.com/mpu6050-accel-gyro-arduino-tutorial/>

<https://www.fluke.com/en/learn/blog/vibration/top-5-industrial-applications-for-vibration-sensors#:~:text=in%20your%20browser,-.What%20is%20a%20vibration%20sensor%3F,and%20frequency%20of%20vibration%20signals.>

<https://www.ni.com/es-mx/shop/data-acquisition/sensor-fundamentals/measuring-vibration-with-accelerometers.html>

<https://www.te.com/usa-en/products/sensors/vibration-sensors.html>

<https://www.elprocus.com/mpu6050-pin-diagram-circuit-and-applications/#:~:text=MPU6050%20is%20highly%20preferred%20for,panoramic%20photo%20capture%2C%20and%20viewing.>

<https://www.escatec.com/blog/what-are-mems-sensors>

<https://predictiva21.com/sensores-de-vibracion/>

https://naylorlampmechatronics.com/blog/45_tutorial-mpu6050-acelerometro-y-giroscopio.html