

Materia:

DISEÑO ELECTRÓNICO BASADO EN
SISTEMAS EMBEBIDOS

Alumno:

Posadas Pérez Isaac Sayeg

Paniagua Rico Juan Julian

García Azzúa Jorge Roberto

Grado y grupo:

8°G

Profesor:

Garcia Ruiz Alejandro Humberto

Tarea 11:

Interpolación Lineal

Interpolación Lineal en Proyectos de Arduino

Introducción

La interpolación lineal es una técnica matemática que usualmente es utilizada para estimar valores dentro de un rango conocido de datos. En el contexto de Arduino y sistemas embebidos, esta técnica es de gran utilidad para el procesamiento de señales, la calibración de sensores y la estimación de datos faltantes. Se basa en la suposición de que la relación entre los puntos de datos es aproximadamente lineal en un intervalo determinado.

El uso de interpolación lineal en proyectos de Arduino permite mejorar la precisión de mediciones en sensores, convertir valores de señales analógicas a escalas específicas y realizar ajustes de datos en procesos de control automatizados. Esta investigación explora los fundamentos de la interpolación lineal, su implementación en Arduino mediante código y su impacto en el procesamiento de datos.

Concepto de Interpolación Lineal

La interpolación lineal es un método matemático para calcular valores desconocidos basándose en dos puntos de datos conocidos. Si se tienen dos puntos $(x_1, y_1), (x_2, y_2)$ se puede estimar un valor intermedio y para un x dado con la siguiente fórmula:

$$y = y_1 + \frac{(y_2 - y_1)}{(x_2 - x_1)} \times (x - x_1)$$

En esta ecuación, y es el valor interpolado correspondiente a un x dentro del intervalo definido por los puntos de referencia. La interpolación lineal se basa en la premisa de que el cambio entre los valores es uniforme y sigue una progresión lineal.

Importancia de la Interpolación Lineal en Arduino

Dentro de los sistemas embebidos y proyectos con Arduino, la interpolación lineal juega un papel clave en diversas aplicaciones. Se emplea principalmente en la

calibración de sensores, la conversión de unidades y la mejora de la precisión de mediciones.

En la calibración de sensores, la interpolación lineal permite transformar las lecturas de sensores en valores comprensibles y precisos. Por ejemplo, los sensores de temperatura suelen proporcionar una señal de voltaje que debe convertirse a grados Celsius o Fahrenheit. Con la interpolación lineal, es posible realizar esta conversión con facilidad.

Otra aplicación importante es la estimación de datos perdidos. En situaciones donde un sensor no proporciona datos en ciertos momentos, se pueden interpolar valores intermedios basándose en mediciones previas y posteriores. Esto es útil en sistemas de monitoreo continuo, donde la falta de datos podría afectar la fiabilidad del análisis.

Además, la interpolación se usa en la corrección de mediciones cuando un sensor presenta una respuesta no completamente lineal. Mediante el ajuste de datos, se puede mejorar la precisión de las lecturas, haciendo que los dispositivos sean más confiables en aplicaciones prácticas.

Implementación de Interpolación Lineal en Arduino

En la plataforma Arduino, la interpolación lineal se puede implementar mediante una función personalizada que realice el cálculo basado en la ecuación mencionada.

Código para Interpolación Lineal

```
float interpolacionLineal(float x1, float y1, float x2, float y2, float x) {  
    return y1 + ((y2 - y1) / (x2 - x1)) * (x - x1);  
}  
  
void setup() {  
    Serial.begin(9600);  
}  
  
void loop() {  
    float x1 = 10, y1 = 20;  
    float x2 = 20, y2 = 40;  
    float x = 15;  
    float resultado = interpolacionLineal(x1, y1, x2, y2, x);  
    Serial.print("Valor interpolado: ");  
    Serial.println(resultado);  
    delay(2000);  
}
```

}

Este código define una función `interpolacionLineal` que calcula un valor estimado para un punto dado dentro de un rango de datos.

Aplicación en Sensores

Un ejemplo concreto de interpolación lineal en Arduino es su uso en sensores de temperatura, como los termistores. Estos sensores no presentan una relación lineal perfecta entre su resistencia y la temperatura, pero en rangos específicos se puede aproximar mediante interpolación lineal.

En el siguiente ejemplo, se convierte una lectura de voltaje proveniente de un termistor en grados Celsius:

```
float convertirVoltajeATemperatura(float voltaje) {  
    float v1 = 1.0, t1 = 25.0;  
    float v2 = 3.0, t2 = 75.0;  
    return t1 + ((t2 - t1) / (v2 - v1)) * (voltaje - v1);  
}  
  
void setup() {  
    Serial.begin(9600);  
}  
  
void loop() {  
    float voltaje = analogRead(A0) * (5.0 / 1023.0);  
    float temperatura = convertirVoltajeATemperatura(voltaje);  
    Serial.print("Temperatura estimada: ");  
    Serial.println(temperatura);  
    delay(1000);  
}
```

Ventajas y Limitaciones de la Interpolación Lineal

Ventajas

Una de las principales ventajas de la interpolación lineal es su facilidad de implementación en sistemas de bajo costo computacional, como Arduino. Al requerir solo cálculos básicos de suma, resta, multiplicación y división, se adapta bien a microcontroladores con recursos limitados.

Otra ventaja es su utilidad en sensores que presentan una relación aproximadamente lineal con la variable medida. Al aplicar interpolación lineal, se pueden mejorar las lecturas de sensores y aumentar la precisión de los sistemas embebidos.

Limitaciones

Sin embargo, la interpolación lineal tiene sus limitaciones. No es adecuada cuando los datos presentan una relación altamente no lineal, ya que la aproximación puede generar errores significativos. Además, si los puntos de referencia no son precisos, la interpolación resultante tampoco lo será. Por último, la interpolación lineal no permite extrapolar valores fuera del rango de los datos conocidos de manera fiable.

Conclusión

La interpolación lineal es una herramienta que es fundamental en la calibración y procesamiento de datos en sistemas embebidos como Arduino. Su implementación es sencilla y eficiente, permitiendo mejorar la precisión de sensores, estimar valores faltantes y convertir unidades en diferentes aplicaciones. Aunque tiene limitaciones en casos de datos no lineales, sigue siendo una técnica útil para mejorar la calidad de los datos en proyectos de electrónica y control.

Bibliografía

1. Banzi, M., & Shiloh, M. (2014). *Getting Started with Arduino*. Maker Media, Inc.
2. Monk, S. (2015). *Programming Arduino: Getting Started with Sketches*. McGraw-Hill Education.
3. Arduino. (n.d.). *Arduino Reference*. Recuperado de <https://www.arduino.cc/reference/en/>
4. Adafruit. (n.d.). *Learning System*. Recuperado de <https://learn.adafruit.com/>
5. Simon Monk. (2014). *Arduino Cookbook*. O'Reilly Media.