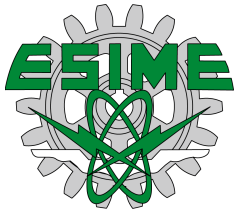




Control inteligente basado en navegación inercial para brazo robótico



Índice general

Introduccion

Los brazos robóticos articulados son sistemas mecánicos con articulaciones rotativas, diseñados para replicar funciones del brazo humano, incluyendo movimientos de rotación y alcance. Suelen tener una pieza en el extremo del robot llamado efector final (Fig. 1), que realiza la función del robot en el entorno (soldar, manipular objetos, etc.). Cada unión en las articulaciones representa un grado de libertad (DoF) [1].

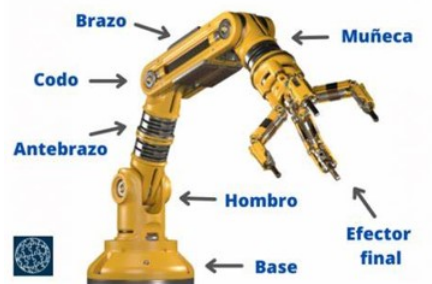


Figura 1: Partes de un brazo robótico articulado de 3 DoF

Sistemas Eléctricos de Potencia Computarizada (SEPDC) es una empresa mexicana que se dedica a fabricar la serie Kaab (Fig. 2) de Controladores Lógicos Programables (PLC), computadoras especializadas para la automatización industrial (tienen inmunidad al ruido eléctrico y resistencia a la vibración y al impacto). Cada uno de ellos puede ser operado de forma remota a través de un software llamado SettDev. Dichos PLCs pueden ser empleados para controlar sistemas críticos.



Figura 2: PLC-Kaab fabricado por SEDPC

Planteamiento del problema

Los brazos robóticos se han introducido rápidamente en la industria, sin embargo, aún existen desafíos para posicionar de manera precisa el brazo robótico que dificultan su introducción en industrias como la textil, sobre todo cuando dicho posicionamiento tiene que realizarse de manera autónoma [2].

Ubicar las articulaciones de un brazo para llegar a una posición deseada es un problema clásico de la robótica llamado cinemática inversa. Existen métodos algebraicos, geométricos e iterativos para resolverlo, sin embargo, dependen de la cantidad de grados de libertad del robot, además de que consumen una gran cantidad de recursos computacionales, lo que dificulta su uso en un sistema crítico [3].

Objetivo general

Desarrollar e implementar un sistema crítico de control, por medio de una Raspberry Pi, sensores inerciales y una red neuronal entrenada con aprendizaje no supervisado, para controlar la posición del efector final de un brazo robótico.

Objetivos específicos

- Desarrollar e implementar un programa en C++, utilizando los ángulos de inclinación en los tres ejes obtenidos por los sensores MPU6050, para determinar la posición del sensor en el extremo de la ortesis de brazo.
- Implementar la comunicación entre la Raspberry Pi y el software SettDev por medio de sockets TCP en C++ y C# para enviar los ángulos de inclinación calculados al software, para poder guardar los datos y permitir reproducirlos en la simulación en 3D de un brazo robótico.
- Desarrollar e implementar una interfaz gráfica de usuario utilizando una pantalla táctil por medio del framework Qt para visualizar la cinemática inversa del brazo robótico a controlar.
- Desarrollar e implementar en SettDev la comunicación entre el módulo del brazo robótico en 3D y el PLC por medio de sockets UDP en C# para enviar los ángulos de inclinación al controlador y reproducir los ángulos de inclinación en los servomotores posicionales.
- Implementar un sistema de inferencia neuro-difuso adaptativo (red neuronal ANFIS) utilizando C++ para resolver el problema de la cinemática inversa del brazo.
- Desarrollar e implementar un algoritmo de aprendizaje no supervisado utilizando C++ para entrenar la red neuronal.

Justificacion

El sistema permitirá implementar un nuevo método para resolver la cinemática directa de forma trigonométrica a través de la navegación inercial, además de implementar un método que no dependa de la cantidad de grados de libertad del robot. Asimismo, será una aplicación práctica de las investigaciones previas a este trabajo sobre redes neuronales para el control de un brazo robótico.

Estado del arte

Marco teorico

Raspberry Pi 3 B+

- Funciona con un sistema operativo basado en la arquitectura ARM.
- Módulo Wi-Fi de banda dual de 2,4 y 5 GHz.
- 40 pines Entrada/Salida de Propósito General (GPIO)
- Salidas de 3.3 y 5 V, buses I²C, SPI



Figura 3: Raspberry Pi 3 B+

Qt

- Optimizada para aplicaciones con interfaces gráficas en sistemas embebidos.
- Soporte para C++
- Utiliza el lenguaje declarativo QML para programar la interfaz



Figura 4: Logotipo de Qt Framework

Bus I2C

- Bus serial de comunicación de tipo maestro-esclavo.
- Línea serial de datos bidireccional (SDA) y línea serial de reloj (SCL).
- Espacio de direcciones, cada dirección identifica a un dispositivo conectado al bus.

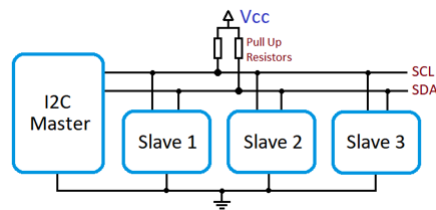


Figura 5: Diagrama del bus I2C

Bus SPI

- Bus serial de comunicación de tipo maestro-esclavo.
- Líneas de entrada al esclavo (MOSI) y salida al esclavo (MISO), línea de reloj (SCLK).
- Línea de selección del chip (SS).

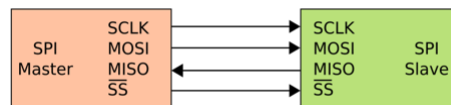


Figura 6: Diagrama del bus SPI

MPU-6050

- Giroscopio de vibración de Coriolis y acelerómetro de 3 ejes.
- Procesador de movimiento digital (DMP) que mide la orientación del sensor en tres ejes.
- Buffer FIFO interno.
- Filtro paso bajo.
- Comunicación por medio del bus I²C de hasta 400 KHz.
- Solo puede tomar las direcciones 0x68 y 0x69.



Figura 7: Sensor inercial MPU6050

MPU-9250

- Giroscopio de vibración de Coriolis y acelerómetro de 3 ejes.
- Procesador de movimiento digital (DMP) que mide la orientación del sensor en tres ejes.
- Buffer FIFO interno.
- Filtro paso bajo.
- Comunicación por medio del bus SPI de hasta 10 MHz.



Figura 8: Sensor inercial MPU9250

Sensor flexible capacitivo

- Mide el ángulo de torsión al que se somete el sensor en una dirección.
- El valor medido es una resistencia variable



Figura 9: Sensor flexible capacitivo

Convertidor analógico-digital MCP3004

- Voltaje de referencia de 2,7 V a 5,5 V



Figura 10: Convertidor A/D MCP3004

Microservomotor posicional

- Ángulo de entrada de 0° a 180°

- Utiliza señales de modulación de ancho de pulso (PWM)
- Movimiento bidireccional



Figura 11: Microservomotor posicional

Propuesta de solución

Desarrollo

Sensado

El algoritmo para obtener los ángulos de inclinación es el que se muestra a continuación.