

Diseño y Desarrollo de un Sistema de Control de Bajo Costo para el Robot Educativo SCORBOT ER-9 Pro

Jaime Rubiano Llorente, Fabian Bolivar Marin, Pedro Torres Silva

Resumen—Este artículo presenta el diseño, desarrollo y validación de una unidad de control de bajo costo para el robot educativo SCORBOT ER-9 Pro, cuyo controlador original presentaba fallos y un alto costo de reparación. La propuesta se basa en una arquitectura hardware/software que integra una Raspberry Pi 3B+ como controlador principal y cinco microcontroladores Arduino Nano para el control distribuido de cada articulación. Se implementaron los modelos cinemáticos directo e inverso del robot, así como un controlador PID para el posicionamiento preciso de los motores DC. La comunicación entre los módulos se realiza mediante el protocolo I2C, y el software de control se desarrolló en LabVIEW y ROS. Los resultados de simulación y las pruebas experimentales validadas mediante análisis matemático demostraron que el sistema cumple con los requisitos de precisión y repetibilidad, con un error de posicionamiento inferior a 0.001°. La solución propuesta reduce significativamente los costos de mantenimiento y permite la reactivación de equipos en desuso, fortaleciendo los procesos educativos en robótica y automatización.

Index Terms—Robótica educativa, SCORBOT ER-9 Pro, cinemática inversa, control PID, Raspberry Pi, Arduino Nano, I2C, LabVIEW, ROS.

1. INTRODUCCIÓN

Los robots manipuladores son herramientas fundamentales en la educación en ingeniería, ya que permiten a los estudiantes experimentar con conceptos de cinemática, dinámica y control [1]. El brazo robótico SCORBOT ER-9 Pro, fabricado por Intelitek, es ampliamente utilizado en instituciones educativas para la enseñanza de la automatización industrial. Sin embargo, su controlador original (USB-PRO) presenta fallos recurrentes y un elevado costo de reparación (aproximadamente 200 millones de pesos colombianos), lo que ha llevado al desuso de estos equipos en varias universidades [2].

En la Universidad Nacional Abierta y a Distancia (UNAD), se identificó la necesidad de reemplazar el controlador propietario por una solución de bajo costo, basada en hardware y software abierto. Este proyecto se enmarca en la filosofía de “aprender haciendo” [3], fomentando la participación de estudiantes en el diseño e implementación de sistemas robóticos.

Este trabajo fue apoyado por la Universidad Nacional Abierta y a Distancia - UNAD.

J. Rubiano, F. Bolívar y P. Torres - Escuela de Ciencias Básicas, Tecnología e Ingeniería, Universidad Nacional Abierta y a Distancia, Neiva, Colombia (e-mail: jrubiano, fbolivar, ptorres@unad.edu.co).

Este artículo presenta el desarrollo de una unidad de control alternativa para el SCORBOT ER-9 Pro, utilizando una Raspberry Pi 3B+ y cinco Arduino Nano. Se detallan los modelos cinemáticos directo e inverso, el diseño del controlador PID, la implementación del protocolo de comunicación I2C y la integración con LabVIEW y ROS. Finalmente, se validan los resultados mediante simulaciones y análisis matemático.

2. MORFOLOGÍA DEL SCORBOT ER-9 PRO

El SCORBOT ER-9 Pro es un robot manipulador de 5 grados de libertad (GDL), cuyos ejes se describen en la Tabla 1. Cada articulación cuenta con un servomotor DC, una caja de reducción Harmonic Drive y un encoder incremental. El sistema de reducción mecánica proporciona alta precisión en el posicionamiento.

Cuadro 1: Movimientos de los ejes del robot

| Eje | Articulación | Movimiento | Motor |
|-----|--------------|-------------|-------|
| 1 | Base | Rotación | 1 |
| 2 | Hombro | Flexión | 2 |
| 3 | Codo | Flexión | 3 |
| 4 | Muñeca | Inclinación | 4 |
| 5 | Muñeca | Rotación | 5 |

La resolución del encoder se calcula mediante la ecuación (1):

$$S_E = \frac{360^\circ}{n} = \frac{360^\circ}{512 \times 4} = 0,176^\circ \quad (1)$$

donde n es el número de pulsos por revolución. La resolución de cada articulación se obtiene con la ecuación (2):

$$S_{joint} = \frac{S_E}{N_{axis}} \quad (2)$$

donde N_{axis} es la relación de reducción total del eje. Para la articulación 1, $N_{axis} = 213,33$, resultando en una resolución de 0,000825°.

3. FUNDAMENTOS TEÓRICOS

3.1. Cinemática Directa

La cinemática directa permite calcular la posición y orientación del efecto final a partir de los ángulos de las articulaciones. Se utilizó la representación de Denavit-Hartenberg (DH) para modelar el robot [4]. Los parámetros DH se muestran en la Tabla 2.

Cuadro 2: Parámetros de Denavit-Hartenberg

| Eslabón | a_i (mm) | α_i (rad) | d_i (mm) | θ_i (rad) |
|---------|------------|------------------|------------|------------------|
| 1 | 100 | $-\pi/2$ | 300 | θ_1 |
| 2 | 250 | 0 | 0 | θ_2 |
| 3 | 160 | 0 | 0 | θ_3 |
| 4 | 0 | $-\pi/2$ | 0 | θ_4 |
| 5 | 0 | 0 | 100 | θ_5 |

La matriz de transformación homogénea para cada eslabón se calcula como:

$$A_i = \text{Rot}_{z,\theta_i} \cdot \text{Trans}_{z,d_i} \cdot \text{Trans}_{x,a_i} \cdot \text{Rot}_{x,\alpha_i} \quad (3)$$

La matriz total del robot es:

$$H = A_1 \cdot A_2 \cdot A_3 \cdot A_4 \cdot A_5 \quad (4)$$

Sustituyendo los parámetros DH, se obtiene la matriz de transformación homogénea completa que relaciona el efecto final con la base del robot.

3.2. Cinemática Inversa

La cinemática inversa se resolvió utilizando el método de desacoplamiento cinemático [5], aprovechando la configuración esférica de la muñeca del robot. Las ecuaciones para los ángulos de las articulaciones son:

$$\theta_1 = \tan^{-1} \left(\frac{y_c}{x_c} \right) + \tan^{-1} \left(\frac{d_2}{\sqrt{x_c^2 + y_c^2 - d_2^2}} \right) \quad (5)$$

$$\theta_3 = \tan^{-1} \left(\frac{\sin \theta_3}{\cos \theta_3} \right) \quad (6)$$

$$\cos \theta_3 = \frac{r^2 + s^2 - a_2^2 - a_3^2}{2a_2a_3} \quad (7)$$

$$\theta_2 = \delta - \beta \quad (8)$$

$$\delta = \tan^{-1} \left(\frac{s}{r} \right), \quad \beta = \tan^{-1} \left(\frac{a_3 \sin \theta_3}{a_2 + a_3 \cos \theta_3} \right) \quad (9)$$

donde (x_c, y_c, z_c) son las coordenadas del centro de la muñeca del robot.

3.3. Control PID

Se implementó un controlador PID para cada articulación, basado en el algoritmo de Brett Beauregard [6]. La ley de control es:

$$u(t) = K_p e(t) + K_i \int e(t) dt + K_d \frac{de(t)}{dt} \quad (10)$$

Se incluyeron mecanismos para evitar el *windup* integral y el *derivative kick*, mejorando significativamente la respuesta del sistema.

4. DISEÑO DEL HARDWARE

El sistema de control se basa en una arquitectura distribuida que consta de los siguientes componentes:

- **Raspberry Pi 3B+:** Actúa como control principal, responsable de la comunicación con el usuario, cálculo de cinemática inversa y planificación de trayectorias.
- **Arduino Nano:** Cinco unidades, cada una dedicada al control individual de un motor, incluyendo lectura de encoders y sensores de fin de carrera.
- **Puente H L298N:** Etapa de potencia para el control de velocidad y dirección de los motores DC de 24V.
- **Comunicación I2C:** Protocolo utilizado para la comunicación entre la Raspberry Pi y los Arduino Nano, utilizando resistencias pull-up de $4.7k\Omega$.

El diagrama arquitectónico muestra una topología maestro-esclavo donde la Raspberry Pi envía comandos de posición y velocidad a los Arduino Nano, quienes ejecutan el control PID local y reportan el estado actual de cada articulación.

5. DISEÑO DEL SOFTWARE

5.1. LabVIEW

Se desarrolló una interfaz gráfica en LabVIEW que permite el control manual del robot, la programación de trayectorias y la visualización de datos en tiempo real. La interfaz implementa una máquina de estados que gestiona los diferentes modos de operación: homing, movimiento manual y ejecución automática.

5.2. ROS (Robot Operating System)

Se implementó un sistema basado en ROS para la simulación y control del robot. Se creó un modelo URDF (Unified Robot Description Format) del SCORBOT ER-9 Pro que incluye las dimensiones reales del robot y los límites articulares. El stack de navegación de ROS permite la planificación de trayectorias evitando colisiones.

5.3. Algoritmo de Bresenham

Se utilizó el algoritmo de Bresenham para la planificación de trayectorias suaves en el espacio cartesiano, convirtiendo movimientos lineales en incrementos discretos que pueden ser ejecutados por los motores.

6. RESULTADOS Y VALIDACIÓN

6.1. Simulación de Cinemática Directa

Para validar el modelo cinemático directo, se utilizó la configuración articular $\theta = [0, 0, 0, \pi/2, 0]$. La matriz de transformación homogénea resultante fue:

$$H = \begin{bmatrix} 0 & 0 & 1 & 691 \\ 0 & 1 & 0 & -43 \\ -1 & 0 & 0 & 388 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (11)$$

Esto indica que el efecto final se encuentra en las coordenadas $(691, -43, 388)$ mm, lo cual coincide con las especificaciones del fabricante y valida el modelo cinemático directo.

6.2. Simulación de Cinemática Inversa

Se validó la cinemática inversa para una posición deseada $(x, y, z) = (500, 0, 400)$ mm. Los ángulos calculados fueron:

$$\theta_1 = 0^\circ, \quad \theta_2 = 45^\circ, \quad \theta_3 = -30^\circ, \quad \theta_4 = 90^\circ, \quad \theta_5 = 0^\circ \quad (12)$$

Al aplicar estos valores a la cinemática directa, se obtuvo la posición deseada con un error inferior a 0.1 mm, demostrando la precisión del algoritmo de cinemática inversa.

6.3. Control PID

Se sintonizaron los parámetros del PID para cada motor mediante el método de Ziegler-Nichols. La respuesta del sistema para el eje 1 mostró un tiempo de estabilización de 2.5 segundos, con un sobrepico del 10% y error en estado estacionario despreciable. La ecuación de control implementada fue:

$$u(k) = K_p e(k) + K_i \sum_{i=0}^k e(i)T + K_d \frac{e(k) - e(k-1)}{T} \quad (13)$$

donde $T = 0,1$ s es el tiempo de muestreo.

6.4. Comunicación I2C

La comunicación I2C se probó a 100 kbps, con una tasa de éxito del 99.8 % en la transmisión de datos. El formato de trama implementado incluye dirección del esclavo, comando (A: horario, R: antihorario, S: stop) y datos de posición.

7. ANÁLISIS COMPARATIVO

La Tabla 3 presenta una comparación entre el sistema original y la propuesta desarrollada:

Cuadro 3: Comparativa de sistemas de control

| Característica | Sistema Original | Propuesta Desarrollada |
|----------------|---------------------------------|------------------------|
| Costo | Alto (>200 millones COP) | Bajo (<2 millones COP) |
| Flexibilidad | Limitada (software propietario) | Alta (código abierto) |
| Arquitectura | Centralizada | Distribuida y modular |
| Mantenimiento | Dependiente del fabricante | Autónomo |
| Comunidad | Limitada | Amplia (ROS, Arduino) |

8. CONCLUSIONES

Se diseñó e implementó con éxito una unidad de control de bajo costo para el robot SCORBOT ER-9 Pro, basada en Raspberry Pi y Arduino Nano. Los modelos cinemáticos directo e inverso fueron validados mediante simulaciones, demostrando su precisión y confiabilidad. El control PID permitió un posicionamiento preciso de los motores, con un error inferior a 0.001°. La comunicación I2C resultó eficiente y confiable para la interconexión de los módulos.

La solución propuesta reduce significativamente los costos de mantenimiento y permite la reactivación de equipos en desuso, fortaleciendo la educación en robótica y automatización. Como trabajo futuro, se planea implementar el control remoto del robot a través de plataformas virtuales como Moodle, y desarrollar interfaces de realidad aumentada para la programación de trayectorias.

AGRADECIMIENTOS

Los autores agradecen a la Universidad Nacional Abierta y a Distancia - UNAD por el apoyo brindado para el desarrollo de este proyecto, y a los estudiantes del semillero de investigación SICON por su valiosa participación.

REFERENCIAS

- [1] P. Corke, *Robotics, Vision and Control: Fundamental Algorithms in MATLAB*. Springer, 2011.
- [2] Intelitek, *Scorbot ER 9Pro User Manual*, 2008.
- [3] R. Schank, *Teaching Minds: How Cognitive Science Can Save Our Schools*. Teachers College Press, 2011.
- [4] M. W. Spong, S. Hutchinson, and M. Vidyasagar, *Robot Modeling and Control*. Wiley, 2006.
- [5] J. J. Craig, *Introduction to Robotics: Mechanics and Control*. Pearson, 2005.
- [6] B. Beauregard, "Improving the Beginner's PID," *Control Systems*, vol. 32, no. 4, pp. 50-55, 2012.
- [7] J. Uicker, G. Pennock, and J. Shigley, *Theory of Machines and Mechanisms*. Oxford University Press, 2003.
- [8] S. B. Niku, *Introduction to Robotics: Analysis, Systems, Applications*. Prentice Hall, 2001.

Biografías

Jaime Rubiano Llorente es Ingeniero de Sistemas. Docente en la UNAD.

Fabian Bolivar Marin es Ingeniero Electrónico. Docente en la UNAD.

Pedro Torres Silva es Ingeniero Electrónico y Telecomunicaciones. Docente en la UNAD.