

Diseño e implementación de un Brazo Robótico de 5 GDL para pulido

Marco Flores, Kaleb Granados Acuña, Gregory Jiménez Jiménez

Escuela de Ingeniería Mecatrónica

Tecnológico de Costa Rica

(marcoflores, kalebgranac13, gregoryjime28)@estudiantec.cr

Resumen—Este artículo describe el diseño e implementación de un brazo robótico de 5 GDL para la tarea de pulido de superficies planas. Utilizando la plataforma DIYMORE ROT3U como base, se presenta un estudio de requerimientos, modelado con cinemática directa e inversa, diagrama eléctrico, generación de trayectorias con polinomios cúbicos y simulación en Matlab. Se muestra además el espacio de trabajo objetivo, el cálculo de la matriz Jacobiana y los parámetros DH del sistema. Se obtuvieron como resultados, porcentajes de error aceptables con los cuales se cumple el objetivo de pulido propuesto para este proyecto.

Palabras clave—Cinemática inversa, MATLAB, Pulido, Robótica, Trayectoria por polinomios cúbicos.

I. INTRODUCCIÓN

En el campo de la robótica, la innovación es constante y la búsqueda de soluciones más eficientes para la automatización sigue siendo un objetivo clave. Frente a tareas tediosas y de difícil maniobrabilidad, los brazos robóticos representan una alternativa eficaz, ya que la automatización convencional basada en PLCs y sistemas similares no siempre ofrece el nivel de precisión necesario que un brazo robótico puede proporcionar.

Los brazos robóticos son capaces de ejecutar tareas que los humanos no pueden realizar fácilmente debido a factores como la fuerza requerida, el riesgo involucrado, la eficiencia operativa, la geometría de trabajo o simplemente la naturaleza repetitiva del proceso [2]. Tal es el caso del pulido, una actividad delicada y laboriosa que puede ser ejecutada con gran precisión por un brazo robótico, gracias a su capacidad de adaptarse a diferentes perfiles y superficies.

En este proyecto se plantea un enfoque académico que busca, a partir de una tarea industrial, desarrollar una solución confiable y replicable. Se reconoce que para realizar un pulido completo sobre geometrías arbitrarias, se requeriría un robot con seis grados de libertad, lo cual implica una solución más costosa y compleja. Por esta razón, se aborda una problemática más concreta: la tarea de “pulir” un cuadrado previamente marcado sobre una pizarra acrílica, lo cual permite explorar los principios fundamentales del control robótico en un entorno simplificado pero representativo.

II. DESCRIPCIÓN DEL PROBLEMA

La superficie a trabajar está representada por un cuadrado de 10x10 centímetros, dibujado sobre una pizarra acrílica. El sistema robótico deberá ser capaz de borrar esta área como parte de su función de pulido. Además, se define que la base

del robot estará situada a una distancia de 15 cm tanto en el eje vertical como en el eje horizontal con respecto al punto inicial del área a pulir. Esta disposición establece un marco fijo de referencia espacial para el diseño y alcance del brazo robótico. En cuanto a la herramienta de pulido, se impone una restricción dimensional: su diámetro no debe exceder los 3 centímetros. Esta limitación garantiza que la herramienta sea lo suficientemente compacta para operar con precisión sobre superficies pequeñas sin comprometer la maniobrabilidad del sistema. Por último, el robot debe contar con al menos tres grados de libertad (GDL), lo cual es un requisito mínimo para permitir movimientos adecuados en el espacio tridimensional y cumplir con la tarea de pulido con la orientación deseada. Con la problemática clara, se puede pasar al diseño e implementación para lograr lo planteado.

III. ENFOQUE E IMPLEMENTACIÓN DE LA SOLUCIÓN

III-A. Diseño Mecánico y Plataforma

En este caso tenemos como base una plataforma de brazo robótico DIYMORE ROT3U 1 que funciona como un comienzo muy fuerte en la parte de diseño mecánico, ya que al ser un brazo comercial de introducción a la robótica, resulta ser bastante robusto. Sus motores MG996R son bastante fuertes para un servomotor (11 Kg de torque [3]) y su estructura de aleación de aluminio nos brinda una base confiable para los servomotores a utilizar.

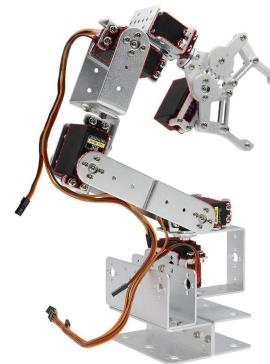


Figura 1: Robot DIYMORE ROT3U 5 GDL

La herramienta por defecto del robot es una pinza, lo que no resulta muy útil para el pulido, por lo que se cambió por un motor DC de 5V con un acople impreso en 3D, el cual nos va

a permitir utilizarlo como la herramienta pulidora (ver figura 2). Hay que tener en cuenta que dicho motor no tiene mucho torque, por lo que hay que tener un acercamiento preciso a la zona de ataque ya que un acercamiento abrupto pararía por completo el movimiento de la herramienta.

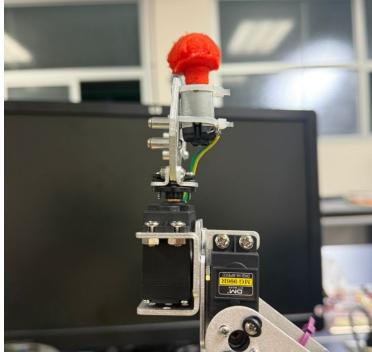


Figura 2: Herramienta impresa en 3D con felpa para pulir

Cabe mencionar que robot posee 6 actuadores, donde 2 de ellos están alineados, perdiendo un grado de libertad. En el caso de este proyecto, se plantea solo el uso de 4 de los 6 motores del robot, debido a que los otros 2 motores no nos ayudan con el proceso de orientación de la herramienta. A lo largo del artículo se verá cómo se incluye el ángulo 5, pero no influye en ningún cálculo.

Además de la estructura del brazo y los servomotores, el conjunto adquirido incluía un controlador PCA9685, el cual permite manejar hasta 16 canales PWM. Este driver se comunica mediante el protocolo I2C, lo que facilita su programación. Además, la alimentación de los motores se realiza de manera independiente a la lógica del sistema, permitiendo un consumo continuo de hasta 5 A [4], lo cual es adecuado considerando que cada motor puede llegar a consumir hasta 2.5 A bajo carga máxima [3].

Para el suministro de potencia se optó por una fuente regulable con dos salidas: una de 5V a 4A y otra de 12V a 1A. Naturalmente, se utilizó la salida de 5V para alimentar el sistema. La fuente se conecta directamente a una toma de corriente de 120VAC mediante un cable de alimentación, lo que hace que la implementación del sistema de potencia en el robot sea bastante sencilla.

Finalmente, el “cerebro” del proyecto es el microcontrolador XIAO ESP32-C3, un dispositivo de dimensiones reducidas pero con capacidades suficientes de procesamiento y excelente fiabilidad. Su tamaño compacto lo hace ideal para esta aplicación. Gracias al uso del driver PCA9685, no es necesario que el microcontrolador genere múltiples señales PWM, lo cual permite utilizar un modelo más simple como este. También podrían haberse empleado otros microcontroladores como un Arduino Nano o un ESP32-WROOM. El sistema completo se observa en la figura 3

III-B. Sistema de Control y Arquitectura Electrónica

Para el control del sistema robótico se utilizó el conjunto de softwares de *Matlab* y *Arduino IDE*. El código de Arduino

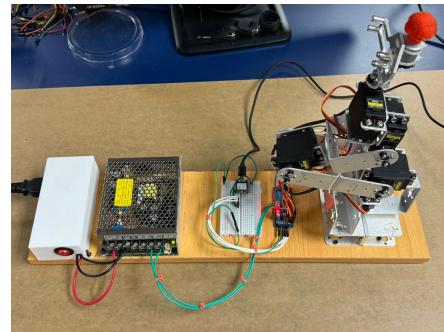


Figura 3: Proyecto robótico completo

considera una lectura continua del puerto serial para modificar los ángulos enviados por Matlab. Por otra parte, para enviar dichos ángulos, Matlab calcula toda la cinemática directa, inversa y trayectorias para el pulido.

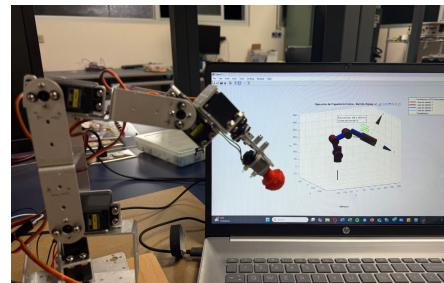


Figura 4: Programación del robot de 5 GDL con Matlab

Este enfoque permite evitar la sobrecarga del microcontrolador, ya que el procesamiento intensivo se realiza en la computadora, la cual cuenta con mayor capacidad de cálculo. Además, facilita el proceso de depuración (debugging) al permitir una visualización más rápida y detallada del comportamiento del sistema. Al delegar el análisis algebraico a la computadora, se logra una solución más cercana a la realidad de los sistemas utilizados en la robótica industrial (observar figura 4), además de una respuesta más ágil. Nuestra implementación también permite observar los cambios en tiempo real, lo que posibilita comparar la posición actual del brazo robótico con la calculada por Robotic Toolbox, otorgando así un alto grado de fiabilidad al sistema.

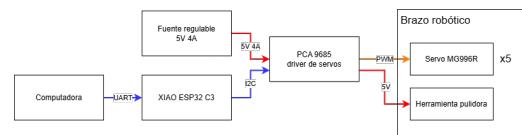


Figura 5: Arquitectura del sistema robótico

Para reflejar todo el sistema robótico se puede utilizar una arquitectura de datos y de potencia que represente todo el proyecto, esto es congruente para diagramas de alto y bajo nivel, ya que no hay ningún bloque en el que se pueda detallar demás. Esta arquitectura se puede observar en la figura 5.

III-C. Estudio Cinemático

III-C1. Parámetros DH y cinemática directa: Para determinar los parámetros de Denavit Hartenberg (DH) primero se definieron las posiciones de los ejes para cada articulación y se definieron las distancias que se utilizarán en todo el desarrollo del proyecto. Cabe mencionar que ciertos giros no son antihorarios, por lo que hay ejes Z's que entran y otros que salen de las articulaciones. Toda esta información se puede observar en el diagrama del robot en la figura 6.

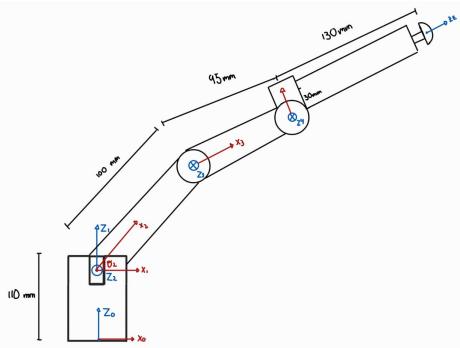


Figura 6: Diagrama del robot 5 GDL para DH.

Para verificar los parámetros DH modificados, se graficó el robot según la Figura 7 utilizando el código en el Anexo A.

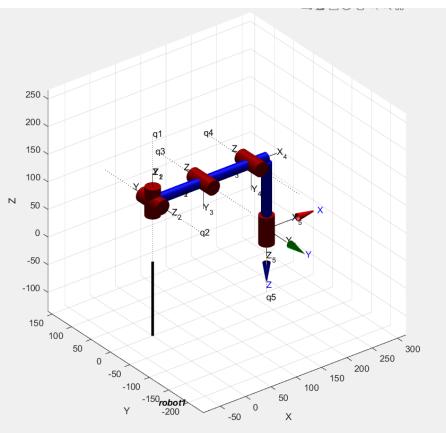


Figura 7: Modelo del robot generado en MATLAB con DH.

Cuadro I: Parámetros DH modificados del robot

i	θ_i	d_i	a_i	α_i
1	θ_1	110	0	0
2	θ_2	0	0	$\frac{\pi}{2}$
3	θ_3	0	100	π
4	θ_4	0	95	0
5	θ_5	130	30	$-\frac{\pi}{2}$

III-C2. Matrices de transformación: Las matrices de transformación se derivan de la matriz de DH, por lo que resulta sencillo ahora determinarlas. Además, la validez de estas matrices fueron verificadas simbólicamente utilizando el código mostrado en el Anexo B.

$$A_1^0 = \begin{bmatrix} \cos(\theta_1) & -\sin(\theta_1) & 0 & 0 \\ \sin(\theta_1) & \cos(\theta_1) & 0 & 0 \\ 0 & 0 & 1 & 110 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_2^1 = \begin{bmatrix} \cos(\theta_2) & -\sin(\theta_2) & 0 & 0 \\ 0 & 0 & -1 & 0 \\ \sin(\theta_2) & \cos(\theta_2) & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_3^2 = \begin{bmatrix} \cos(\theta_3) & -\sin(\theta_3) & 0 & 100 \\ -\sin(\theta_3) & -\cos(\theta_3) & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_4^3 = \begin{bmatrix} \cos(\theta_4) & -\sin(\theta_4) & 0 & 95 \\ \sin(\theta_4) & \cos(\theta_4) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_5^4 = \begin{bmatrix} 1 & 0 & 0 & 30 \\ 0 & 0 & 1 & 130 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_0^5 = \begin{bmatrix} \cos(\theta_3 - \theta_4) \cos(\theta_1 + \theta_2) & -\cos(\theta_3 - \theta_4) \sin(\theta_1 + \theta_2) & \sin(\theta_3 - \theta_4) & 110 \sin(\theta_3 - \theta_4) + 100 \cos(\theta_4) + 125 \\ -\sin(\theta_3 - \theta_4) \cos(\theta_1 + \theta_2) & -\cos(\theta_3 - \theta_4) \sin(\theta_1 + \theta_2) & 0 & 130 \\ \sin(\theta_3 - \theta_4) \cos(\theta_1 + \theta_2) & -\sin(\theta_3 - \theta_4) \sin(\theta_1 + \theta_2) & -\cos(\theta_3 - \theta_4) & -110 \cos(\theta_3 - \theta_4) - 100 \sin(\theta_4) \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

III-D. Cinematica inversa

La cinemática inversa puede resolverse analíticamente solo para ciertos manipuladores cuyas estructuras geométricas lo permiten, como es el caso del PUMA 560. Cuando se trata de manipuladores con un número distinto a seis grados de libertad, como aquellos con cinco GDL, el problema se complica notablemente. En estos casos, no siempre es posible alcanzar una orientación arbitraria del efecto final para una posición dada, ya que el sistema no posee los grados de libertad suficientes para desacoplar completamente la orientación de la posición. Esta limitación estructural implica que la cinemática inversa puede no tener solución para ciertas poses o requerir métodos numéricos y aproximaciones específicas al diseño del robot [1].

El acercamiento que se tiene es tomar la matriz de transformación de la base al efecto final y deducir la cinemática inversa por medio de álgebra para, seguidamente, obtener ciertos ángulos con un acercamiento geométrico. Esto con el fin de lograr determinar todos los ángulos del robot de alguna manera. En el caso práctico, es mucho mejor utilizar herramientas que hagan las aproximaciones de la cinemática de forma iterativa, como lo hace *Robotic ToolBox* de Matlab o como se podría programar en lenguajes como Python.

Se desea determinar los valores de los ángulos articulares $\theta_1, \theta_2, \theta_3, \theta_4$. Por lo que se van a seccionar en pasos para entender el procedimiento necesario:

Para facilitar la notación, se definen:

$$\varphi = \theta_1 + \theta_2$$

$$\gamma = \theta_3 - \theta_4$$

Paso 1: Determinación de φ :

$$\varphi = \text{atan}2(-r_{21}, r_{11}) \quad (1)$$

Paso 2: Determinación de γ :

$$\gamma = \text{atan}2(r_{13}, -r_{33}) \quad (2)$$

Paso 3: Determinación de θ_4 :

$$\theta_4 = \cos^{-1} \left(\frac{p_x - 110 \sin(\gamma) - 125}{100} \right) \quad (3)$$

Paso 4: Determinación de θ_3 :

$$\theta_3 = \gamma + \theta_4 \quad (4)$$

Paso 5: Determinación de θ_2 :

$$\rho = \sqrt{(p_x - 125)^2 + p_z^2} \quad (5)$$

$$\theta_2 = \cos^{-1} \left(\frac{\rho^2 - a_2^2 - a_3^2}{2a_2a_3} \right) \quad (6)$$

Paso 6: Determinación de θ_1 :

$$\theta_1 = \varphi - \theta_2 \quad (7)$$

Para condensar todas las ecuaciones encontradas se procede a resumir todas las expresiones en la tabla II.

Cuadro II: Resumen de la cinemática inversa para el robot de 5 GDL

Ángulo	Expresión
φ	$\text{atan}2(-r_{21}, r_{11})$
γ	$\text{atan}2(r_{13}, -r_{33})$
θ_4	$\cos^{-1} \left(\frac{p_x - 110 \sin(\gamma) - 125}{100} \right)$
θ_3	$\gamma + \theta_4$
ρ	$\sqrt{(p_x - 125)^2 + p_z^2}$
θ_2	$\cos^{-1} \left(\frac{\rho^2 - a_2^2 - a_3^2}{2a_2a_3} \right)$
θ_1	$\varphi - \theta_2$
θ_5	$\text{atan}2(r_{32}, r_{31})$

III-D1. Matriz Jacobiana: Se utilizó el código en el Anexo C para calcular y verificar esta matriz simbólicamente.

$$J = \begin{bmatrix} 110 \cos(\theta_3 - \theta_4) & -110 \cos(\theta_3 - \theta_4) - 100 \sin(\theta_4) \\ 0 & 0 \\ 110 \sin(\theta_3 - \theta_4) & -110 \sin(\theta_3 - \theta_4) - 100 \cos(\theta_4) \end{bmatrix}$$

III-D2. Cálculo teórico de torques para borrar una pizarra blanca: Para estimar los torques que el robot debe generar al aplicar fuerza sobre la pizarra blanca, se consideró una fuerza de contacto normal teórica basada en referencias de la literatura y recomendaciones técnicas.

Diversos estudios indican que una fuerza en el rango de 2 N a 5 N es suficiente para borrar efectivamente sin dañar la superficie [5]. Estudios ergonómicos para tareas manuales semejantes respaldan fuerzas similares [6].

$$\mathbf{F} = \begin{bmatrix} 0 \\ 0 \\ -3 \end{bmatrix} \text{ N}$$

y se evaluó la configuración articular

$$\mathbf{q} = [60^\circ, 0^\circ, 90^\circ, 100^\circ, 0^\circ].$$

Esta fuerza se introdujo en el modelo simbólico del robot para calcular la expresión de los torques articulares mediante la relación:

$$\boldsymbol{\tau} = J_v^\top \cdot \mathbf{F}$$

donde J_v es la Jacobiana traslacional del efecto final.

Los resultados obtenidos para los torques son:

$$\begin{aligned} \tau_1 &= 0,000 \text{ Nm}, & \tau_2 &= 0,000 \text{ Nm}, \\ \tau_3 &= 0,0573 \text{ Nm}, & \tau_4 &= -0,102 \text{ Nm}, \\ \tau_5 &= 0,000 \text{ Nm}. \end{aligned}$$

Para más detalles sobre la implementación del cálculo de torques, se puede consultar el código en el Anexo D.

III-E. Generación de Trayectorias mediante Polinomios Cúbicos

El sistema implementado utiliza polinomios cúbicos por segmentos para generar trayectorias suaves entre waypoints articulares del robot manipulador. Cada segmento se modela mediante la ecuación $q(t) = a_0 + a_1t + a_2t^2 + a_3t^3$, donde los coeficientes se determinan a partir de cuatro condiciones de frontera: posiciones inicial y final específicas, y velocidades nulas en ambos extremos. Esta aproximación garantiza continuidad en posición y velocidad, eliminando discontinuidades que provocarían sacudidas mecánicas indeseables. El cálculo de coeficientes se realiza analíticamente mediante la resolución del sistema de ecuaciones lineales resultante, permitiendo una evaluación computacionalmente eficiente de la trayectoria en intervalos de tiempo regulares definidos por el paso dt .

El sistema de barrido robótico opera sobre un área de 100x100 mm mediante un eficiente patrón en zigzag. Para mejorar la cobertura, la trayectoria incorpora un desfase progresivo, creando un plano de barrido inclinado. La operación optimiza el tiempo usando velocidades diferenciadas: movimientos de trabajo lentos y precisos, transiciones verticales rápidas entre líneas, y desplazamientos seguros para el inicio y fin de la tarea. Este enfoque garantiza un equilibrio entre eficiencia, precisión y seguridad en toda la operación. Para profundizar en la implementación de las trayectorias se puede consultar el anexo E

IV. ANÁLISIS DE LA SOLUCIÓN

La evaluación del desempeño del robot se dividirá en secciones que permitan analizar y comprender tanto las funciones implementadas como la solución propuesta al problema del robot pulidor. No obstante, el objetivo del proyecto no

es únicamente lograr que el sistema funcione, sino también que cumpla con posiciones y trayectorias específicas. Estas se validarán mediante distintos métodos de comprobación, y se clasificarán como exitosas, mejorables o fallidas, según corresponda.

IV-A. Comprobación de ángulos

Los ángulos fueron comparados de manera exhaustiva en todo el desarrollo del proyecto, principalmente por pasar del espacio cartesiano (cinemática inversa) al espacio de los actuadores (PWM de los servos), los cuales se trabajaron como "offsets" que luego se sumaron o restaron según correspondía. En el caso de comprobar su cercanía a lo enviado por la programación se optó por medir con respecto al cero del actuador y ver qué tanto se acercaba al ángulo deseado. Se midió con un transportador (ver figura 8), lo cual induce bastante error en las mediciones y además tiene una alta incertidumbre, los datos de estas mediciones se pueden observar en la tabla III.

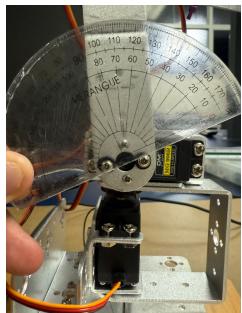


Figura 8: Comprobación de ángulos con transportador

Cuadro III: Comparación de ángulo teórico con ángulo práctico en el robot de 5 GDL

Medición	Art 1	Art 2	Art 3	Art 4
Med 1	70	65	155	110
Med 2	65	65	150	110
Med 3	65	70	150	105
Med 4	65	65	150	105
Med 5	70	70	150	110
Promedio	67.0	67.0	151.0	108.0
Teórico	60	67	130	97
Error [%]	11.7	0.0	16.2	11.3

* Todas las mediciones se realizaron con una incertidumbre estimada de $\pm 5^\circ$.

Estas mediciones, tomando en cuenta la incertidumbre y el error humano inducido, se considera **exitosa**.

IV-B. Comprobación de posiciones

Las posiciones se comprueban por medio de los puntos que se definieron para determinar la trayectoria de polinomios cúbicos, como la posición de home y acercamiento, que son su base. Las mediciones se realizarán con un juego de dos reglas de 40 cm las cuales tienen una incertidumbre de $\pm 0.1\text{cm}$.

En las 2 coordenadas medidas no hay un error mayor al 10 %, por lo que se puede dar esta sección como **exitosa**.

Cuadro IV: Mediciones de verificación de coordenadas - Punto de acercamiento

Posición Teórica (mm)	1	2	3	4	5
X = 100	11.0	10.8	10.8	10.9	11.0
Y = 0	-0.5	-0.7	-0.5	-0.7	-0.5
Z = 170	18.0	18.9	18.0	18.1	17.9
Promedio (X,Y,Z)	10.9	-0.58	18.18		
Teórico (X,Y,Z) (cm)	10.0	0.0	17.0		
Error (X,Y,Z) [%]	9.0	—	6.9		

Cuadro V: Mediciones de verificación de coordenadas - Punto de "Home"

Posición Teórica (mm)	1	2	3	4	5
X = -32.6	-2.9	-3.0	-3.0	-2.9	-3.2
Y = 0	1.0	0.7	0.8	0.8	0.7
Z = 409	39.0	36.8	36.7	37.0	36.8
Promedio (X,Y,Z)	-3.0	0.8	37.26		
Teórico (X,Y,Z) (cm)	-3.26	0.0	40.9		
Error (X,Y,Z) [%]	7.9	—	8.9		

IV-C. Comprobación de los Torques

Para determinar los torques articulares generados por una fuerza ejercida por el robot sobre el suelo se realizó una medición experimental colocando una balanza (observar figura 9) bajo el efecto final. Se registró la fuerza ejercida verticalmente para la configuración articular:

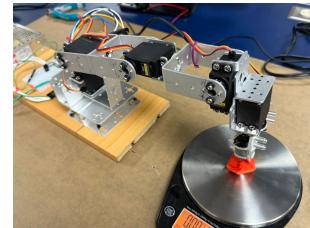


Figura 9: Medición de fuerza experimental

$$\mathbf{q} = [60^\circ, 0^\circ, 90^\circ, 100^\circ, 0^\circ]$$

La fuerza se midió en el eje Z en unidades de gramos-fuerza (gN). La Tabla VI muestra los valores obtenidos y su conversión a Newtons ($1 \text{ gN} = 9.81 \text{ m/s}^2 \div 1000$):

Cuadro VI: Mediciones experimentales y conversión a Newtons

Medición (gN)	Fuerza (N)
650.1	6.375
685.4	6.726
725.6	7.119
710.3	6.973
685.9	6.731
Promedio	6.782

La fuerza promedio fue de 6,782 N. Como el robot empujaba hacia abajo, se modeló como $\vec{F} = [0, 0, -6,782]^\top \text{ N}$.

Este valor y la configuración articular se utilizaron en el código `torques_experimentales.m` (ver Anexo D), el cual calcula los torques simbólicos a partir de la Jacobiana traslacional:

$$\tau = J_v^\top \cdot \vec{F}$$

El resultado fue:

$$\begin{aligned}\tau_1 &= 0,000 \text{ Nm}, & \tau_2 &= 0,000 \text{ Nm}, \\ \tau_3 &= 0,129 \text{ Nm}, & \tau_4 &= -0,247 \text{ Nm}, \\ \tau_5 &= 0,000 \text{ Nm}\end{aligned}$$

Análisis: Los torques no nulos se concentran en θ_3 y θ_4 , articulaciones responsables del movimiento vertical del efecto. El signo de τ_3 (positivo) y τ_4 (negativo) refleja la dirección del momento generado por la fuerza normal aplicada. La magnitud superior de los torques experimentales indica una fuerza de contacto mayor que la empleada en el caso teórico, lo cual sugiere que el robot ejerce un esfuerzo suficiente para borrar eficazmente la superficie.

IV-D. Comprobación de trayectorias

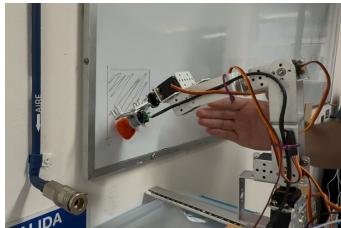


Figura 10: Comprobación de la trayectoria en la pizarra acrílica

Una vez generadas las trayectorias del robot utilizando polinomios cúbicos para interpolar posiciones articulares a lo largo del tiempo, se procedió a comprobar que los movimientos simulados coincidieran con los realizados físicamente por el brazo robótico.

Para ello, se dibujó sobre una pizarra acrílica la trayectoria esperada de la punta del efecto final, de acuerdo con los puntos vía definidos en el espacio cartesiano. Luego, se ejecutaron los comandos de movimiento generados por el script principal y enviados al robot por comunicación serial. En la Figura 10 se observa cómo el robot reproduce con precisión la trayectoria planeada.

La coincidencia entre lo simulado en MATLAB y la ejecución física confirma que la interpolación por polinomios cúbicos, la conversión de trayectorias cartesianas a articulares mediante cinemática inversa, y el modelo del robot basado en parámetros DH modificados fueron implementados correctamente.

La comprobación completa puede observarse en el siguiente video de demostración: <https://www.youtube.com/watch?v=dMsFVONa-UM>

V. CONCLUSIONES

1. Se logró diseñar e implementar un brazo robótico funcional de 5 GDL capaz de realizar trayectorias de pulido sobre una superficie plana.
2. Se determinó la capacidad de determinar la cinemática inversa de un robot de 5 GDL, a pesar de tener dificultad con la no linealidad de las ecuaciones algebraicas.
3. La implementación de trayectorias mediante polinomios cúbicos mejoró la suavidad de los movimientos y su control.
4. Los resultados experimentales para la posición son menores al 10 %, por lo cual se consideran aceptables. Los resultados de las mediciones angulares también se consideran aceptables, tomando en cuenta la incertidumbre de la herramienta y el error humano en la toma de datos.
5. Se comprobó que el robot puede ejercer fuerzas suficientes para borrar una superficie, cumpliendo con el objetivo práctico del proyecto.
6. El uso de *MATLAB* para cálculos y simulación, combinado con *Arduino* para la ejecución, permitió una arquitectura de control eficiente y replicable.

VI. RECOMENDACIONES

1. Mejorar el diseño mecánico incorporando una estructura más rígida o articulaciones con menor holgura para reducir errores acumulativos durante el movimiento.
2. Se puede integrar una forma de medir la distancia a la pizarra para poder borrar de forma segura, como un sensor láser o ultrasónico, esto con un control de lazo cerrado.
3. Explorar el uso de un manipulador de 6 GDL para permitir mayor flexibilidad en la orientación del efecto final en tareas más complejas.
4. Integrar sensores de fuerza o presión en el efecto final para realizar control en lazo cerrado y mantener constante la fuerza de contacto, mejorando el resultado del pulido.
5. Optimizar la herramienta de pulido con un motor más robusto o sistema de sujeción mejorado para aumentar la eficiencia y durabilidad.
6. Optimizar la planificación de trayectorias considerando no solo posiciones y velocidades, sino también el modelado de aceleraciones y tiempos mínimos, para reducir el desgaste mecánico y mejorar la eficiencia temporal del proceso.

REFERENCIAS

- [1] J. Craig, *Introduction to Robotics: Mechanics and Control*, 3rd ed., Pearson, 2005.
- [2] S. Surati, S. Hedao, T. Rotti, V. Ahuja, and N. Patel, “Pick and Place Robotic Arm: A Review Paper,” *International Research Journal of Engineering and Technology (IRJET)*, vol. 8, no. 2, pp. 2121–2125, Feb. 2021. [Online]. Available: <https://www.irjet.net/archives/V8/i2/IRJET-V8I2425.pdf>
- [3] ETC, “MG996R High Torque Metal Gear Dual Ball Bearing Servo Datasheet,” Alldatasheet. [Online]. Available: <https://www.alldatasheet.es/html-pdf/1131873/ETC2/MG996R/110/1/MG996R.html>.
- [4] NXP Semiconductors, “PCA9685 – 16-channel, 12-bit PWM Fm+ PC-bus LED controller Datasheet,” Alldatasheet, Rev. 02, Jul. 16, 2009. [Online]. Available: <https://www.alldatasheet.com/html-pdf/293576/NXP/PCA9685/54/1/PCA9685.html>.
- [5] X. Chen, Z. Huang, and W. Chen, “Force control in robotic wiping tasks for surface cleaning,” *IEEE Transactions on Automation Science and Engineering*, vol. 13, no. 3, pp. 1326–1337, 2016.
- [6] Occupational Safety and Health Administration (OSHA), *Ergonomics Guidelines*. [Online]. Available: <https://www.osha.gov/ergonomics>

ANEXOS

A continuación, se presenta un resumen de los scripts desarrollados y utilizados en MATLAB para el análisis cinemático y generación de trayectorias del robot, se puede acceder a estos scripts en el siguiente repositorio de GitHub: https://github.com/marcofv1/robot_Matlab_codes:

Cuadro VII: Listado de scripts utilizados en el proyecto

Anexo	Descripción	Archivo
A	Graficar el robot y los sistemas de referencia usando parámetros DH modificados.	proyDH.m
B	Cálculo simbólico de las matrices de transformación usando DH modificado.	proyT.m
C	Obtención de la matriz Jacobiana simbólica a partir de las matrices de transformación.	Jacobiano.m
D	Cálculo de torques a partir de datos experimentales.	torques.m
E	Código final implementado.	Proyecto_Brazo_Pulidor.m