

# INSTITUTO TECNOLÓGICO AUTÓNOMO DE MÉXICO

INGENIERÍA EN MECATRÓNICA

## Diseño e implementación de un robot dibujador planar de tres grados de libertad

**Alumnos:** Mauricio Verduzco Chavira - 195106, Carlos Lozada Rojas - 168506, Alejandro Terrazas Maeshiro - 187977

**Asignatura:** Robótica

**Profesor:** José Guadalupe Romero Velázquez

CDMX, MÉXICO  
2025

# Índice general

<b>Resumen</b>	<b>3</b>
<b>1. Descripción general del sistema</b>	<b>4</b>
1.1. Tipo de robot . . . . .	4
1.2. Función principal . . . . .	4
1.3. Diagrama general del sistema . . . . .	4
1.4. Parte mecánica . . . . .	6
1.5. Control . . . . .	7
1.6. Software . . . . .	7
<b>2. Diseño mecánico</b>	<b>8</b>
2.1. Arquitectura mecánica . . . . .	8
2.1.1. Configuración RRP . . . . .	8
2.1.2. Longitudes de los eslabones . . . . .	8
2.2. Selección de actuadores . . . . .	9
2.2.1. Motor NEMA 23 (base) . . . . .	9
2.2.2. Motor NEMA 17 (segundo eje) . . . . .	10
2.2.3. Justificación de torque, peso y distribución . . . . .	10
2.3. Diseño del mecanismo del plumón . . . . .	11
2.4. Materiales y fabricación . . . . .	11
<b>3. Modelado cinemático del robot</b>	<b>12</b>
3.1. Planteamiento del problema cinemático . . . . .	12
3.2. Espacio cartesiano válido de operación . . . . .	12
3.3. Solución de la cinemática inversa . . . . .	12
3.3.1. Consideraciones numéricas y de alcanzabilidad . . . . .	13
<b>4. Sistema electrónico</b>	<b>15</b>
4.1. Arquitectura electrónica . . . . .	15
4.2. Unidad de control y protocolo de comunicación . . . . .	15
4.3. Etapa de potencia y control de motores a paso . . . . .	15
4.3.1. Eje 1: NEMA 23 con driver DM556 . . . . .	16
4.3.2. Eje 2: NEMA 17 con driver TB6600 . . . . .	16
4.3.3. Formato de comando de movimiento (G-code) . . . . .	16
4.4. Control del mecanismo del plumón (solenoide 12 V con MOSFET) . . . . .	16

4.5. Alimentación y distribución de energía . . . . .	17
4.6. Asignación de pines . . . . .	17
<b>5. Sistema de control y software</b>	<b>18</b>
5.1. Enfoque general del sistema de control . . . . .	18
5.2. Flujo general del software . . . . .	18
5.3. Procesamiento de imagen . . . . .	19
5.4. Generación y optimización de trayectorias . . . . .	19
5.5. Mapeo cartesiano a ángulos articulares . . . . .	19
5.6. Generación de G-code . . . . .	20
5.7. Envío e interpretación del G-code . . . . .	20
5.8. Control del actuador del plumón . . . . .	20
<b>6. Implementación práctica</b>	<b>21</b>
6.1. Montaje del robot . . . . .	21
6.2. Integración de la electrónica . . . . .	21
6.3. Calibración inicial . . . . .	22
6.4. Posición <i>home</i> y referencia de trabajo . . . . .	22
<b>7. Lista de componentes</b>	<b>23</b>
7.1. Componentes electrónicos y de potencia . . . . .	23
7.2. Componentes electromecánicos . . . . .	24
7.3. Componentes mecánicos . . . . .	24

# Resumen

En este trabajo se presenta el diseño e implementación de un robot dibujador planar de tres grados de libertad, capaz de reproducir imágenes sobre una superficie plana a partir de trayectorias definidas en coordenadas cartesianas. El sistema transforma puntos  $(x, y)$  en ángulos articulares mediante un modelo de cinemática inversa, permitiendo el control preciso de los movimientos del robot.

El proyecto integra diseño mecánico, modelado cinemático, electrónica de control y programación, empleando motores a paso y un mecanismo de subida y bajada del plumón para diferenciar entre movimientos de trazado y desplazamientos sin dibujo. El desarrollo está orientado a la comprensión y aplicación práctica de conceptos fundamentales de robótica, así como a la replicabilidad del sistema en entornos académicos.

# Capítulo 1

## Descripción general del sistema

### 1.1. Tipo de robot

El sistema desarrollado corresponde a un robot manipulador planar de tipo RRP, con tres grados de libertad. Las dos primeras articulaciones son de tipo rotacional y operan dentro de un mismo plano de trabajo, permitiendo posicionar el efecto final en coordenadas cartesianas bidimensionales ( $x, y$ ), lo que resulta adecuado para aplicaciones de dibujo y trazado sobre superficies planas. La tercera articulación es de tipo prismático y corresponde al actuador encargado de subir y bajar el plumón, cuya función es controlar el contacto con la superficie de dibujo.

La orientación del estilógrafo se mantiene constante durante la operación, ya que el objetivo principal del robot es el posicionamiento preciso del punto de contacto sobre el plano de trabajo y el control del inicio y fin de los trazos, sin requerir manipulación de la orientación del efecto final.

### 1.2. Función principal

El proyecto consiste en el diseño e implementación de un robot tipo brazo mecánico cuya función principal es recibir una imagen digital, específicamente una fotografía de una persona, procesarla y replicarla sobre una hoja de papel mediante trazos continuos. El sistema transforma la imagen original en coordenadas cartesianas que son interpretadas por el robot para controlar el movimiento de sus articulaciones, permitiendo así reproducir la fotografía en forma de dibujo. De esta manera, el robot integra procesamiento digital de imágenes, control de movimiento y actuadores mecánicos para ejecutar un proceso de dibujo automatizado de manera precisa y repetible.

### 1.3. Diagrama general del sistema

De manera general, el funcionamiento del robot puede entenderse como una secuencia de bloques funcionales claramente definidos, que inicia con la adquisición de una imagen digital y finaliza con la ejecución física del dibujo. El proceso comienza cuando el usuario toma o selecciona una fotografía, la cual es procesada en MATLAB mediante varias etapas

de tratamiento de imagen, incluyendo conversión a escala de grises, ajuste de resolución, detección de bordes y vectorización.

Posteriormente, las trayectorias obtenidas son ajustadas en escala y desplazamiento para adaptarse al área de trabajo del robot, además de optimizar el recorrido en coordenadas cartesianas con el fin de reducir movimientos innecesarios. A partir de estos puntos cartesianos, se realiza el mapeo a los ángulos articulares correspondientes mediante la cinemática inversa del robot, generando instrucciones en formato G-code con una nomenclatura compatible con el sistema de control.

El G-code generado es enviado de manera incremental desde MATLAB hacia el microcontrolador a través de una conexión serial, evitando la saturación del búfer de comunicación. Finalmente, el Arduino R4 interpreta estas instrucciones para controlar las velocidades, los ángulos de las articulaciones y el actuador del plumón, permitiendo que el robot ejecute el dibujo con un estilógrafo sobre una hoja en blanco.

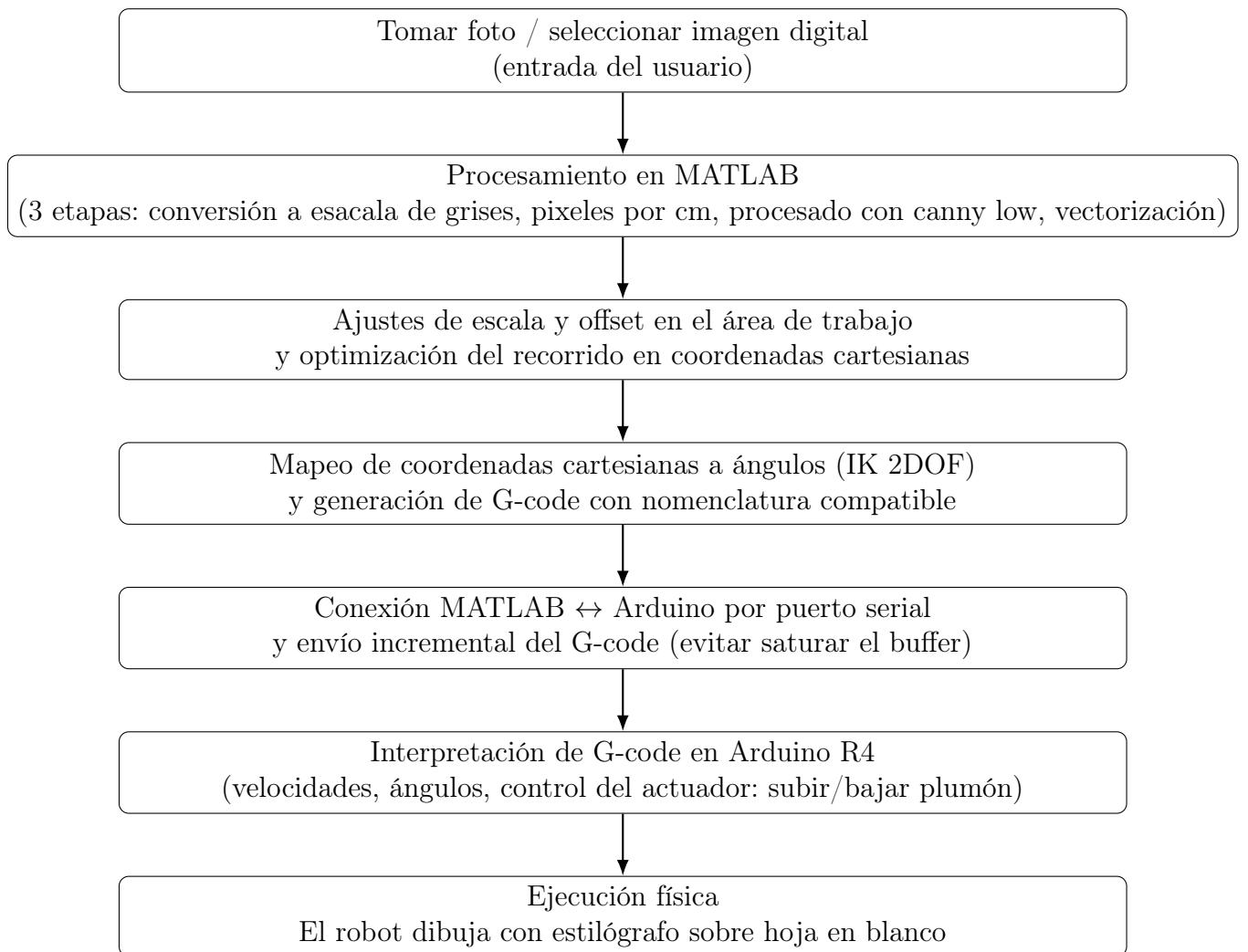
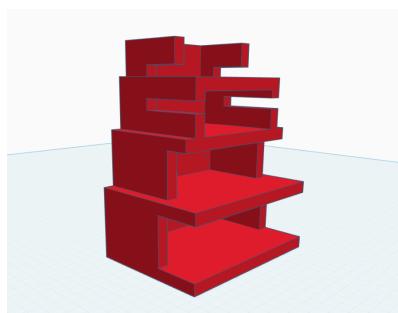


Figura 1.1: Diagrama secuencial del flujo completo: de fotografía a dibujo ejecutado por el robot.

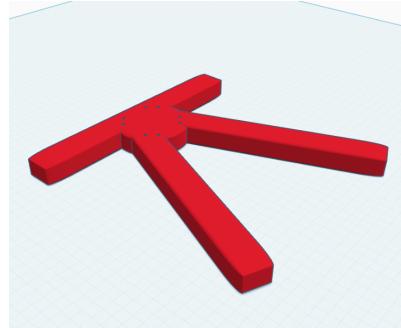
## 1.4. Parte mecánica

La parte mecánica del robot está compuesta por un brazo de dos eslabones rígidos unidos mediante articulaciones rotacionales. En el extremo del segundo eslabón se encuentra montado el mecanismo portapluma, encargado de realizar el contacto con la superficie de dibujo.

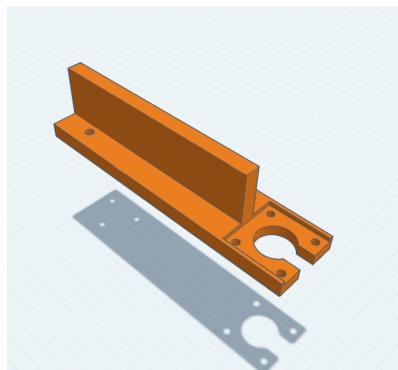
La estructura fue diseñada para ofrecer rigidez suficiente durante la operación, manteniendo al mismo tiempo un peso reducido. La mayoría de las piezas estructurales fueron fabricadas mediante impresión 3D, lo que permite una fácil replicación y modificación del diseño.



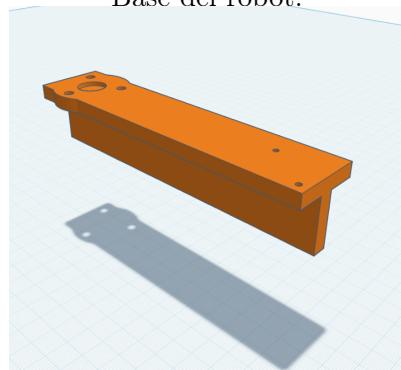
Torre que aloja la electrónica del sistema.



Base del robot.



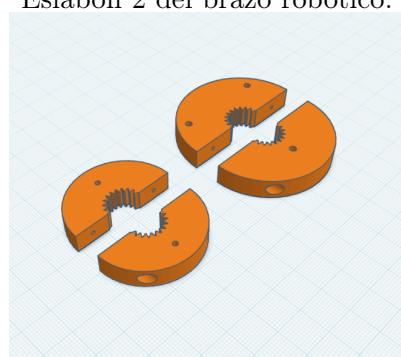
Eslabón 1 del brazo robótico.



Eslabón 2 del brazo robótico.



Mecanismo del actuador.



Abrazaderas para acoplar los motores con los links.

Cuadro 1.1: Principales piezas mecánicas impresas en 3D del robot dibujador.

## **1.5. Control**

El control del robot se basa en el posicionamiento articular calculado a partir de las coordenadas cartesianas deseadas. Para cada punto del dibujo, se calculan los ángulos correspondientes a las articulaciones utilizando el modelo de cinemática inversa del robot.

Los movimientos se ejecutan de manera secuencial siguiendo las trayectorias definidas, incorporando comandos específicos para activar o desactivar el mecanismo del plumón según corresponda.

## **1.6. Software**

El software del sistema se divide en dos niveles principales. El primero corresponde al procesamiento y la planificación, los cuales se ejecutan en el entorno MATLAB. En esta etapa se realiza la conversión de imágenes en trayectorias de dibujo, el ajuste de escala y offset en el área de trabajo, así como la generación de los archivos de control en formato G-code.

El segundo nivel corresponde a la ejecución del movimiento del robot y se implementa en el entorno de desarrollo del Arduino. En esta etapa, el microcontrolador interpreta las instrucciones recibidas, gestiona las velocidades y posiciones de los motores, y controla los actuadores del sistema, permitiendo la ejecución física del dibujo.

Esta separación permite una mayor modularidad del sistema, facilitando futuras modificaciones tanto en los algoritmos de procesamiento y generación de trayectorias como en el control directo del hardware del robot.

# Capítulo 2

## Diseño mecánico

### 2.1. Arquitectura mecánica

El robot presenta una arquitectura mecánica de tipo RRP, conformada por dos articulaciones rotacionales coplanares que conectan dos eslabones rígidos. Esta configuración permite posicionar el efecto final dentro de un plano bidimensional, lo que resulta adecuado para tareas de dibujo y trazado sobre superficies planas.

El sistema cuenta con tres grados de libertad, suficientes para cubrir el espacio de trabajo requerido sin introducir complejidad innecesaria en el análisis cinemático ni en el control del robot.

#### 2.1.1. Configuración RRP

La configuración cinemática del robot corresponde a un sistema RRP, ya que está compuesto por dos articulaciones rotacionales y una articulación prismática. Las dos primeras articulaciones rotacionales controlan el movimiento planar del brazo mecánico, permitiendo posicionar la punta del efecto final en el plano de trabajo mediante la rotación de los eslabones. La tercera articulación, de tipo prismático, corresponde al actuador lineal encargado de subir y bajar el plumón, cuya función no es posicionar el efecto en el plano, sino controlar el contacto con la superficie de dibujo. Esta configuración permite separar el control de posicionamiento en el plano del control de trazado, simplificando el análisis cinemático del sistema.

#### 2.1.2. Longitudes de los eslabones

Las longitudes de los eslabones fueron definidas de acuerdo con el alcance deseado del robot y la capacidad de los actuadores utilizados. Para el diseño final se emplearon las siguientes dimensiones:

$$L_1 = 14 \text{ cm}$$

$$L_2 = 14 \text{ cm}$$

donde  $L_1$  corresponde a la distancia entre el eje del motor de la base y el eje del segundo motor, y  $L_2$  corresponde a la distancia entre el eje del segundo motor y la punta del estilografo.

## 2.2. Selección de actuadores

Para accionar las articulaciones del robot se seleccionaron motores a paso con diferentes capacidades de torque, de acuerdo con las exigencias mecánicas de cada eje.

### 2.2.1. Motor NEMA 23 (base)

La primera articulación, ubicada en la base del robot, es accionada por un motor a paso NEMA 23 con un torque nominal de  $1.2 \text{ N}\cdot\text{m}$ . Este motor es responsable de soportar el mayor esfuerzo mecánico del sistema, ya que debe mover el peso total del brazo y contrarrestar los momentos generados cuando el robot se encuentra extendido.

La elección de este actuador garantiza un margen adecuado de torque para evitar pérdida de pasos y mantener un movimiento estable durante la ejecución de las trayectorias.



Figura 2.1: Motor a pasos tipo Nema 23

## 2.2.2. Motor NEMA 17 (segundo eje)

La segunda articulación es accionada por un motor a paso NEMA 17 con un torque nominal de  $0.4 \text{ N}\cdot\text{m}$ . Este motor se encarga de mover el segundo eslabón y el mecanismo del plumón.

El uso de un motor más compacto en esta articulación permite reducir la masa y la inercia del sistema, mejorando la respuesta dinámica del robot sin comprometer la capacidad de carga requerida para la aplicación de dibujo.



Figura 2.2: Motor a pasos tipo Nema 17

## 2.2.3. Justificación de torque, peso y distribución

La combinación de un motor NEMA 23 en la base y un motor NEMA 17 en el segundo eje permite una distribución eficiente del torque a lo largo del brazo. El motor de mayor capacidad se ubica en la base, donde los momentos son mayores, mientras que el motor de menor torque se coloca en una posición que minimiza su impacto en la inercia total del sistema.

Esta distribución contribuye a un movimiento más suave y preciso, reduciendo vibraciones y esfuerzos innecesarios en las articulaciones.

## 2.3. Diseño del mecanismo del plumón

El mecanismo encargado de subir y bajar el plumón se diseñó utilizando un actuador lineal tipo solenoide, con una carrera aproximada de 1 cm, suficiente para garantizar el contacto y la separación del estilógrafo respecto a la superficie de dibujo. Este actuador corresponde a la articulación prismática del sistema y su función es controlar exclusivamente el inicio y fin de los trazos, sin intervenir en el posicionamiento planar del robot.

Durante las primeras pruebas, el solenoide fue accionado directamente mediante un relé, lo cual provocó un rápido sobrecalentamiento del actuador. Esto se debía a que el solenoide permanecía alimentado con corriente continua al 100 % durante todo el tiempo de activación, condición que resultó inviable para un uso prolongado y repetitivo.

Para resolver este problema, se implementó un módulo de control basado en un MOSFET LR7843, el cual permitió aplicar una estrategia de alimentación en dos etapas. En la primera etapa, el solenoide recibe un pulso de corriente al 100 % para generar el golpe inicial necesario para el desplazamiento del núcleo. Posteriormente, la corriente se reduce aproximadamente al 50 %, valor que se determinó de forma experimental, ya que niveles cercanos al 40 % no proporcionaban la fuerza suficiente para mantener el plumón elevado de manera estable.

Con una alimentación al 50 %, el actuador logra sostener el plumón en posición incluso durante movimientos del brazo, evitando descensos no deseados por efecto de la gravedad. Al mismo tiempo, este nivel de corriente mantiene al solenoide en un estado térmico seguro, permaneciendo tibio incluso durante períodos prolongados de operación continua.

Esta solución permitió eliminar el problema de sobrecalentamiento, mejorar la confiabilidad del sistema y garantizar un funcionamiento estable del mecanismo del plumón durante ciclos prolongados de dibujo.

## 2.4. Materiales y fabricación

La mayoría de las piezas estructurales del robot fueron fabricadas mediante impresión 3D, utilizando filamento Rapid PETG de la marca Elegoo. Este material fue seleccionado por su buena resistencia mecánica, estabilidad dimensional y facilidad de impresión. Adicionalmente, este tipo de filamento es de fácil adquisición a través de plataformas comerciales como Amazon o Mercado Libre, donde se encuentra disponible en presentaciones de varios colores básicos.

El ensamblaje del robot se realizó utilizando tornillería estándar, lo que facilita el armado y desmontaje del sistema. La base del robot fue diseñada para proporcionar estabilidad durante la operación, reduciendo vibraciones y movimientos indeseados durante el dibujo.

Durante el diseño mecánico se consideraron aspectos como la rigidez estructural, la alineación de los ejes de rotación y las tolerancias mecánicas, con el objetivo de garantizar un funcionamiento confiable y reproducible del sistema.

# Capítulo 3

## Modelado cinemático del robot

### 3.1. Planteamiento del problema cinemático

El problema de la cinemática inversa consiste en determinar, para cada punto  $(x, y)$ , el par de ángulos articulares  $(\theta_1, \theta_2)$  que permiten alcanzar dicha posición, considerando las dimensiones reales del robot y sus limitaciones físicas.

Para el diseño final se utilizan las siguientes longitudes de eslabones:

$$L_1 = 14 \text{ cm}$$

$$L_2 = 14 \text{ cm}$$

donde  $L_1$  corresponde a la distancia entre el eje del motor de la base y el eje del segundo motor, y  $L_2$  a la distancia entre el eje del segundo motor y la punta del estilógrafo.

### 3.2. Espacio cartesiano válido de operación

Antes de aplicar la cinemática inversa, las trayectorias se trabajan completamente en el espacio cartesiano. Durante esta etapa se identifican regiones del plano que no son adecuadas para la operación del robot, ya sea por limitaciones geométricas o por interferencias mecánicas.

En particular, existe un área de singularidad cercana al eje del motor de la base donde el robot no puede operar de forma segura, debido a posibles colisiones del brazo o del mecanismo del estilógrafo con la base del motor NEMA 23. Por esta razón, el sistema considera un área de exclusión alrededor del origen cartesiano, la cual se implementa indirectamente mediante offsets y transformaciones geométricas aplicadas a las trayectorias.

Este filtrado previo permite que únicamente se procesen puntos cartesianos físicamente alcanzables y seguros antes de aplicar el modelo cinemático.

### 3.3. Solución de la cinemática inversa

Una vez definidos los puntos cartesianos válidos, se aplica la solución analítica de la cinemática inversa para un manipulador planar. Aunque el robot cuenta con una configuración

RRP, la tercera articulación de tipo prismático corresponde únicamente al mecanismo encargado de subir y bajar el plumón, por lo que no afecta la posición del efecto final en el plano ( $X, Y$ ). En consecuencia, el problema de cinemática inversa puede simplificarse considerablemente y tratarse como un manipulador de tipo RR para el cálculo de los ángulos articulares asociados al posicionamiento planar.

La distancia radial desde el origen hasta el punto objetivo se define como:

$$r = \sqrt{x^2 + y^2}$$

Utilizando la ley de los cosenos, el ángulo de la segunda articulación se obtiene a partir de:

$$\cos(\theta_2) = \frac{x^2 + y^2 - L_1^2 - L_2^2}{2L_1L_2}$$

Debido a la naturaleza del sistema, un mismo punto cartesiano puede alcanzarse con dos configuraciones geométricas distintas. En la implementación práctica se selecciona una rama fija del solucionador (configuración de codo), lo que evita cambios bruscos en la postura del robot durante la ejecución de trayectorias continuas. El ángulo  $\theta_2$  se calcula mediante:

$$\theta_2 = \text{atan2}(\sin(\theta_2), \cos(\theta_2))$$

donde el signo de  $\sin(\theta_2)$  se fija de acuerdo con la configuración seleccionada.

Definiendo los parámetros auxiliares:

$$\begin{aligned} k_1 &= L_1 + L_2 \cos(\theta_2) \\ k_2 &= L_2 \sin(\theta_2) \end{aligned}$$

el ángulo de la primera articulación se calcula como:

$$\theta_1 = \text{atan2}(y, x) - \text{atan2}(k_2, k_1)$$

Estas expresiones permiten transformar directamente cada punto cartesiano del dibujo en los ángulos articulares correspondientes.

### 3.3.1. Consideraciones numéricas y de alcanzabilidad

Las consideraciones relacionadas con singularidades y alcanzabilidad del robot no se abordan directamente dentro del cálculo de la cinemática inversa, sino durante la etapa de generación de los puntos cartesianos que definen las trayectorias de dibujo. Este enfoque permite simplificar el cálculo posterior de los ángulos articulares y evitar configuraciones problemáticas desde el origen del proceso.

De manera directa, se eliminan todos los puntos cartesianos que se encuentran dentro de un radio aproximado de 6 cm alrededor del origen del sistema de coordenadas. Esta región corresponde a una zona cercana a la base del robot donde pueden presentarse configuraciones cercanas a singularidades o movimientos no deseados de las articulaciones.

Adicionalmente, se descartan aquellos puntos que se encuentran fuera del alcance geométrico del robot, determinado por la longitud total de sus eslabones. De esta forma, únicamente

se conservan los puntos que pertenecen al espacio de trabajo alcanzable y seguro del mecanismo.

Al aplicar estas restricciones antes de la etapa de cinemática inversa, se garantiza que los puntos procesados sean físicamente realizable, se evita el cálculo de soluciones inválidas y se mejora la estabilidad numérica y operativa del sistema durante la ejecución del dibujo.

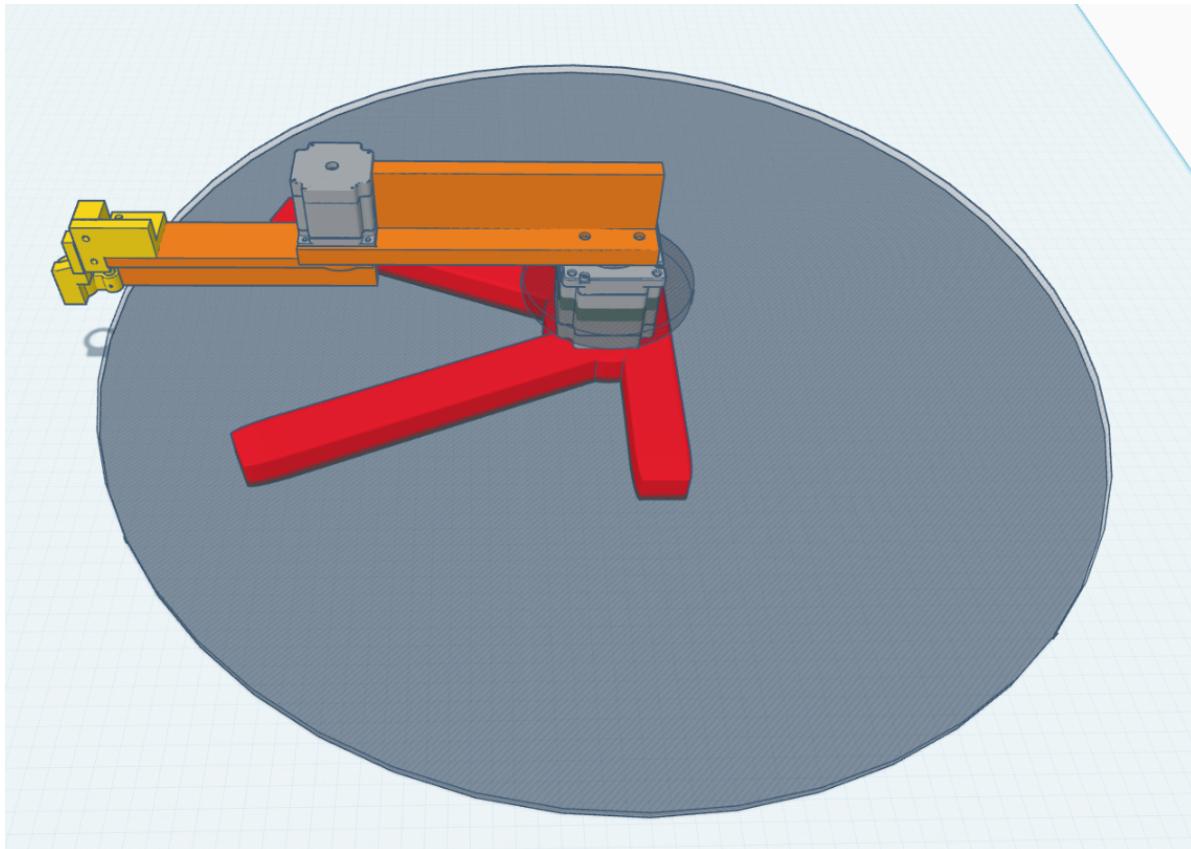


Figura 3.1: Área de trabajo: Singularidad en círculo pequeño sombreado, alcance brazo robot para dibujar en círculo grande sombreado

# Capítulo 4

## Sistema electrónico

### 4.1. Arquitectura electrónica

El sistema electrónico del robot se diseñó con una arquitectura centralizada: un microcontrolador interpreta comandos y genera señales digitales de control, mientras que la potencia de los actuadores se maneja mediante módulos dedicados. El sistema se divide en cuatro bloques: (i) unidad de control, (ii) control de motores a paso por drivers, (iii) control del actuador del plumón y (iv) alimentación y distribución de energía.

### 4.2. Unidad de control y protocolo de comunicación

La unidad de control utilizada es un **Arduino UNO R4 WiFi**. El microcontrolador recibe instrucciones en formato G-code desde una computadora mediante comunicación serial configurada a **115200 baudios**. La computadora envía el G-code de manera secuencial (línea por línea) y el firmware responde con un mensaje de confirmación **ok** tras la correcta recepción y procesamiento de cada instrucción.

Para este fin, se implementó un protocolo de comunicación bidireccional, en el cual MATLAB envía una línea de G-code y espera explícitamente la respuesta de **ok** por parte del Arduino antes de transmitir la siguiente instrucción. Este mecanismo permite verificar que la ejecución se esté realizando correctamente, evita la saturación del búfer de comunicación y garantiza una sincronización confiable entre el software de planificación y el sistema de control del robot.

### 4.3. Etapa de potencia y control de motores a paso

El robot cuenta con dos motores a paso, cada uno controlado por un driver independiente mediante señales **STEP/DIR** y una señal de habilitación **EN**. El firmware asume una configuración de microstepping equivalente a **6400 microsteps por revolución** (**STEPS\_PER\_REV=6400**), utilizada para convertir grados a pasos.

#### 4.3.1. Eje 1: NEMA 23 con driver DM556

El motor NEMA 23 del eje base es controlado mediante un driver **DM556**. El control se realiza por señales digitales de **paso (STEP)** y **dirección (DIR)** generadas por el Arduino, además de una señal de habilitación **EN**. En la implementación, la habilitación del driver se realiza con nivel lógico alto (**EN=HIGH**).

#### 4.3.2. Eje 2: NEMA 17 con driver TB6600

El motor NEMA 17 del segundo eje es controlado mediante un driver **TB6600**, también con interfaz **STEP/DIR** y señal **EN**. Debido al montaje mecánico del motor, el firmware invierte el sentido de giro del eje 2 mediante software (**INVERT\_DIR\_MOTOR2=true**).

#### 4.3.3. Formato de comando de movimiento (G-code)

El firmware interpreta comandos de movimiento con el siguiente formato:

G1 X<angulo1> A<angulo2> Y<vel1\_deg\_s> Z<vel2\_deg\_s>

donde **X** representa el ángulo objetivo del eje 1 en grados, **A** el ángulo objetivo del eje 2 en grados, y **Y** y **Z** las velocidades objetivo en grados por segundo para cada eje. El movimiento se ejecuta con una micro-rampa trapezoidal para reducir aceleraciones bruscas, mejorar la estabilidad mecánica y disminuir vibraciones.

### 4.4. Control del mecanismo del plumón (solenoide 12 V con MOSFET)

El control del actuador se basa en la técnica de modulación por ancho de pulso (*Pulse Width Modulation*, PWM), la cual permite regular la potencia entregada a una carga variando el ciclo de trabajo de una señal digital. Mediante PWM es posible controlar la corriente promedio aplicada al solenoide sin necesidad de elementos disipativos, lo que resulta especialmente útil para aplicaciones donde se requiere un control eficiente de fuerza y temperatura.

El sistema de subida y bajada del plumón se implementa mediante un solenoide de 12 V controlado por un módulo basado en MOSFET LR7843. El control se realiza desde el Arduino mediante PWM para aplicar un impulso inicial de activación (*golpe*) y posteriormente reducir la potencia para mantenimiento. En el firmware se implementan los siguientes estados de control:

- **Golpe inicial:** PWM al 100 % (**PWM\_FULL = 255**) durante **GOLPE\_MS = 80 ms**.
- **Mantenimiento:** PWM reducido al 50 % (**PWM\_MAINTAIN ≈ 127**).
- **Apagado:** PWM a 0 para desactivar el solenoide.

Los comandos utilizados para el control del solenoide son **M5** y **M3**. En la implementación del firmware, el comando **M5** activa el solenoide (*golpe + mantenimiento*), mientras que **M3** lo apaga.

## 4.5. Alimentación y distribución de energía

El sistema se alimenta mediante una fuente tipo slim de **24 V, 300 W, 12 A**. Esta fuente suministra energía a los drivers de motores a paso. El solenoide opera a **12 V** y se controla con el módulo LR7843. Se recomienda mantener **tierra común** entre Arduino, drivers y módulos de potencia para evitar diferencias de potencial y problemas de referencia en señales STEP/DIR y PWM.

## 4.6. Asignación de pines

La Tabla 4.1 resume la asignación de pines utilizada en el sistema electrónico, según el firmware implementado en Arduino.

Cuadro 4.1: Asignación de pines del sistema electrónico (Arduino UNO R4 WiFi).

Función	Pin Arduino	Descripción
Eje 1 STEP	D7	STEP hacia DM556 (NEMA 23)
Eje 1 DIR	D6	DIR hacia DM556 (NEMA 23)
Eje 1 EN	D5	Enable DM556 (EN=HIGH en firmware)
Eje 2 STEP	D10	STEP hacia TB6600 (NEMA 17)
Eje 2 DIR	D9	DIR hacia TB6600 (NEMA 17)
Eje 2 EN	D8	Enable TB6600 (EN=HIGH en firmware)
Solenoide (PWM)	D3	Control LR7843 (golpe + mantenimiento)

# Capítulo 5

## Sistema de control y software

### 5.1. Enfoque general del sistema de control

El sistema de control del robot se diseñó bajo un enfoque jerárquico y distribuido. Las tareas de mayor complejidad computacional, como el procesamiento de imágenes, la generación de trayectorias y el cálculo de la cinemática inversa, se realizan en una computadora mediante MATLAB. El microcontrolador se encarga exclusivamente de la ejecución en tiempo real de los movimientos y del control de los motores en el Arduino IDE.

Este enfoque permite simplificar el firmware del microcontrolador, facilitar la depuración del sistema y modificar los algoritmos de alto nivel sin afectar la estabilidad del control físico del robot.

### 5.2. Flujo general del software

El flujo completo del software se implementa como una cadena de procesos secuenciales que inicia con una imagen digital y finaliza con la ejecución física del dibujo. Dicho flujo puede resumirse en las siguientes etapas:

1. Procesamiento de la imagen de entrada.
2. Conversión de la imagen a trazos en coordenadas cartesianas.
3. Optimización de trayectorias en el espacio cartesiano.
4. Mapeo de coordenadas cartesianas a ángulos articulares a partir de la cinemática inversa.
5. Generación de código G.
6. Envío secuencial del G-code al microcontrolador con confirmación.
7. Ejecución del movimiento del robot.

Cada etapa se implementa de forma modular, permitiendo ajustes independientes y facilitando la extensión del sistema.

### 5.3. Procesamiento de imagen

El procesamiento de imagen se realiza en MATLAB mediante una serie de scripts organizados por etapas (`Process_Pict_Pt1` a `Process_Pict_Pt4`). En estas etapas, la imagen original se transforma en un conjunto de trazos representados por puntos cartesianos.

Durante este proceso se aplican técnicas de detección de bordes, conversión a representación tipo *sketch* y segmentación de trazos, con el objetivo de obtener trayectorias adecuadas para ser reproducidas físicamente por el robot.

---

### 5.4. Generación y optimización de trayectorias

Una vez obtenidos los trazos en coordenadas cartesianas, se realiza una optimización orientada al comportamiento físico del robot. Esta etapa incluye:

- Re-muestreo de los trazos con un paso cartesiano constante, definido mediante el parámetro `paso_cm`, para evitar cambios bruscos de posición.
- Reordenamiento de los trazos, seleccionando de manera iterativa el siguiente trazo más cercano al punto final del trazo anterior, con el fin de reducir desplazamientos innecesarios.
- Aplicación de offsets y transformaciones geométricas globales para alinear el dibujo con el sistema de referencia del robot.
- Exclusión implícita de regiones del espacio cartesiano no adecuadas para la operación, asociadas a singularidades provocando colisiones potenciales.

Estas acciones permiten una ejecución más fluida, reduciendo tiempos muertos y movimientos no productivos.

### 5.5. Mapeo cartesiano a ángulos articulares

El mapeo de las trayectorias cartesianas a ángulos articulares se realiza en MATLAB mediante el script `Mapeo_XY_Robot.m`, el cual implementa el modelo analítico de cinemática inversa del robot con  $L_1 = L_2 = 14$  cm.

En la implementación, el archivo de coordenadas se procesa realizando un intercambio de ejes, asignando  $x$  a partir de la segunda columna del archivo y  $y$  a partir de la primera. Para cada punto  $(x, y)$  se calculan los ángulos  $(\theta_1, \theta_2)$  utilizando una rama fija del solucionador, con el fin de mantener continuidad en la postura del robot durante la ejecución de un trazo.

Los puntos no alcanzables o fuera del espacio de trabajo válido son descartados antes de generar comandos de movimiento.

## 5.6. Generación de G-code

A partir de los ángulos articulares calculados, MATLAB genera un archivo de control en formato G-code. El formato de comando utilizado es:

```
G1 X<angulo1> A<angulo2> Y<vel1_deg_s> Z<vel2_deg_s>
```

donde **X** corresponde al ángulo objetivo del eje 1 (en grados) y **A** al ángulo objetivo del eje 2 (en grados). Las variables **Y** y **Z** representan velocidades angulares en grados por segundo para cada eje.

Las velocidades se definen de manera independiente mediante los parámetros `speedTheta1` y `speedTheta2`. En la configuración utilizada en este proyecto, ambos valores se establecieron iguales; sin embargo, el sistema permite asignar velocidades distintas para cada motor si se requiere.

En el G-code generado se incluyen comandos específicos para el control del mecanismo del plumón. El comando `M5` se utiliza para activar el actuador lineal, provocando que el plumón se eleve y deje de dibujar durante los desplazamientos sin trazo. Por otro lado, el comando `M3` se emplea para desactivar el actuador, permitiendo que el plumón descienda por gravedad y aplique contacto con la superficie de dibujo, asistido por un resorte ligero que proporciona una presión adicional. Esta lógica permite separar claramente los movimientos de posicionamiento de los movimientos de dibujo, garantizando trazos definidos y evitando marcas no deseadas sobre el papel.

## 5.7. Envío e interpretación del G-code

El G-code generado se envía desde MATLAB al microcontrolador de manera secuencial a través de una conexión serial. Un script de conexión se encarga de transmitir cada línea del archivo y de esperar la confirmación correspondiente antes de enviar la siguiente instrucción.

En el firmware del microcontrolador, cada comando recibido se interpreta, se traduce a señales `STEP/DIR` para los drivers de los motores a paso y se ejecuta utilizando rampas de aceleración y desaceleración. Al finalizar la ejecución de una instrucción, el microcontrolador envía un mensaje de confirmación (`ok`), lo que permite mantener sincronizado el flujo de control entre MATLAB y el robot.

## 5.8. Control del actuador del plumón

El control del actuador del plumón se integra dentro del mismo flujo de ejecución del G-code. Los comandos específicos permiten activar o desactivar el solenoide en los momentos adecuados, coordinando el movimiento del robot con el contacto sobre la superficie de dibujo.

El uso de modulación por ancho de pulso en el control del solenoide permite aplicar un impulso inicial de activación y posteriormente reducir la potencia durante el mantenimiento, mejorando la eficiencia energética y reduciendo el calentamiento del actuador.

# Capítulo 6

## Implementación práctica

### 6.1. Montaje del robot

El montaje del robot se realizó de manera modular, ensamblando de forma independiente la base, el primer eslabón, el segundo eslabón y el mecanismo del plumón. Esta estrategia permitió verificar el correcto funcionamiento de cada subsistema antes de integrar el conjunto completo.

Los motores a paso se fijaron directamente a las piezas estructurales impresas en 3D, cuidando la alineación de los ejes de rotación para evitar esfuerzos mecánicos adicionales. Durante el montaje se prestó especial atención a la rigidez de las uniones y a la correcta sujeción de los motores, ya que pequeñas holguras pueden traducirse en errores apreciables durante el dibujo.

### 6.2. Integración de la electrónica

Una vez ensamblada la estructura mecánica, se procedió a la integración del sistema electrónico. Los drivers de los motores a paso se montaron de forma externa al robot, permitiendo una adecuada ventilación y facilitando el acceso a ajustes de corriente y microstepping.

El cableado de las señales STEP, DIR y EN se realizó manteniendo recorridos cortos y ordenados, reduciendo la probabilidad de interferencias. Asimismo, se verificó la correcta referencia de tierra común entre el microcontrolador, los drivers y el módulo de control del solenoide.

Antes de realizar cualquier movimiento, se comprobó el correcto funcionamiento del solenoide de manera independiente, validando tanto el golpe inicial como el nivel de mantenimiento por PWM. Adicionalmente, se realizaron pruebas desde el monitor serial del Arduino para verificar el correcto funcionamiento de los motores, enviando comandos con ángulos deseados y velocidades específicas, así como los comandos del actuador (M3 y M5), asegurando la correcta respuesta del sistema antes de su operación conjunta.

### 6.3. Calibración inicial

La calibración inicial del sistema consiste en colocar manualmente el brazo del robot en una posición conocida antes de iniciar cualquier operación. Este procedimiento es indispensable debido a que los motores utilizados no cuentan con encoders, por lo que el control de posición se basa en el conteo de pasos a partir de una referencia inicial.

Una vez realizada esta calibración, el sistema puede iniciar la ejecución del proceso de dibujo con una referencia consistente, garantizando que los cálculos de movimiento se realicen de forma coherente durante toda la operación.

### 6.4. Posición *home* y referencia de trabajo

La posición *home* se define como la configuración en la cual el brazo del robot se encuentra en un punto determinado deseado. Esta posición se utiliza como referencia absoluta para el inicio y finalización de todos los movimientos del sistema.

A partir de la posición *home*, el control del robot se realiza mediante el conteo de pasos de los motores, lo que permite determinar la posición del brazo en cualquier momento del proceso de dibujo. Al finalizar la ejecución, el robot regresa nuevamente a esta posición de referencia como mecanismo de verificación.

Si al regresar a la posición *home* el brazo no queda en el punto deseado y perpendicular a la base, se considera que ha ocurrido una pérdida de pasos en los motores, provocando un desfase en la trayectoria. En este caso, es necesario apagar el sistema, recolocar manualmente el brazo en la posición *home* y reiniciar el sistema para restablecer la referencia de trabajo.

# Capítulo 7

## Lista de componentes

En este capítulo se presentan los principales componentes utilizados en la construcción del robot dibujador. La selección de cada elemento se realizó considerando disponibilidad, confiabilidad y compatibilidad con los requerimientos del sistema.

### 7.1. Componentes electrónicos y de potencia

Componente	Descripción
Arduino R4 WIFI	Microcontrolador principal encargado del control del robot y la interpretación del G-code
Driver TB6600	Driver para motor a pasos utilizado en una de las articulaciones rotacionales
Driver DM556	Driver para motor a pasos de mayor capacidad, utilizado en la articulación principal
Módulo MOSFET LR7843	Módulo de potencia utilizado para el control eficiente del actuador solenoide
Buck Step-Down XL4016 (200 W)	Regulador DC-DC encargado de reducir y estabilizar los niveles de voltaje
Fuente de poder 24 V DC	Fuente de alimentación principal del sistema, de operación silenciosa

Cuadro 7.1: Componentes electrónicos y de potencia del sistema

## 7.2. Componentes electromecánicos

Componente	Descripción
Motor a pasos NEMA 23	Motor utilizado para una de las articulaciones del brazo robótico
Motor a pasos NEMA 24	Motor utilizado en la articulación principal del robot
Actuador solenoide 12 V	Actuador lineal con carrera aproximada de 1 cm para el control del plumón

Cuadro 7.2: Componentes electromecánicos del sistema

## 7.3. Componentes mecánicos

Los elementos estructurales del robot fueron fabricados mediante impresión 3D, lo que permitió una alta flexibilidad en el diseño, facilidad de modificación y reducción de costos de manufactura. Las principales piezas impresas en 3D que conforman la estructura del sistema se listan a continuación:

- Base del robot.
- Eslabón 1 del brazo robótico.
- Eslabón 2 del brazo robótico.
- Mecanismo del actuador para el control del plumón.
- Soporte para eje y motor NEMA 17.
- Soporte para eje y motor NEMA 23.
- Torre de control para la electrónica y el cableado.

Estas piezas proporcionan soporte estructural al sistema, permiten el montaje preciso de los actuadores y facilitan la integración de los componentes electrónicos, contribuyendo a la estabilidad y funcionalidad general del robot dibujador.