

Interpolation Strategies for Motion Planning in a 2-Degree-of-Freedom SCARA Robot

Estrategias de Interpolación para la Planificación de Movimientos en un Robot SCARA de 2 Grados de Libertad

Fermín Marcelo R. Maciel Barboza^{*1}, Antonio Concha Sánchez¹, Leonel Soriano Equigua¹, Luis Alberto Contreras Aguilar¹, José Luis Álvarez Flores¹, and Víctor H. Castillo Topete¹

¹Universidad de Colima,
Facultad de Ingeniería Mecánica y Eléctrica,
Carretera Colima - Coquimatlan km 9, Valle de las Huertas, 28400 Coquimatlan, Col.
{fermin_maciel, aconcha, lsoriano, luisc, alvarez_jose, victorc}@ucol.mx

Abstract

This article presents a systematic methodology for trajectory planning in 2-degree-of-freedom SCARA robots using linear and circular interpolation. The robot's kinematics are addressed, describing both direct and inverse equations, and algorithms for generating smooth trajectories are detailed. The G-code is explored as a tool for motion control, and Matlab software is utilized for simulations. The versatility of these concepts is emphasized, showcasing how the methodology can be applied to various types of robots. Linear and circular interpolation algorithms, along with the integration of G-code for motion control, offer a didactic approach providing a practical experience to comprehend the fundamental principles of trajectory planning.

Keywords— Robot Path, Interpolation, Gcode

Resumen

Este artículo presenta una metodología sistemática para la planificación de trayectorias en robots SCARA de 2 grados de libertad mediante interpolación lineal y circular. Se aborda la cinemática del robot, describiendo las ecuaciones directas e inversas, y se detallan algoritmos para generar trayectorias suaves. Se explora el código G como herramienta de control de movimiento y se utiliza el software Matlab para simulaciones. La versatilidad de estos conceptos se destaca, mostrando cómo la metodología puede aplicarse a diversos tipos de robots. Los algoritmos de interpolación lineal y circular, junto con la integración del código G para el control de movimiento, ofrecen un enfoque didáctico que proporciona una experiencia práctica para comprender los principios fundamentales de la planificación de trayectorias.

Palabras clave— Trayectoria del robot, Interpolación, Código G

I. Introducción

Los robots de brazo robótico articulado de cumplimiento selectivo (SCARA) se utilizan ampliamente en las operaciones de montaje y manipulación en la industria [1].

El autor [2] presenta una revisión del modelo cinemático y dinámico, así como una descripción del modelo de

aceleración de un manipulador industrial de 2 grados de libertad (DoF). Los autores en [1] presentan un diseño de un controlador barato y de alta calidad, se derivan las ecuaciones cinemáticas del robot SCARA y se construye la planificación de la trayectoria utilizando el método de interpolación circular del método del analizador diferencial digital vectorial. En [3] se implementa un brazo manipulador que se ejecuta a lo largo de una secuencia de ruta específica de acuerdo con la solicitud de planificación de ruta y la restricción del brazo del robot manipulador.

* Autor de correspondencia

El autor en [4] desarrolla un programa que determina las leyes de movimiento de los actuadores que permiten que el robot genere el movimiento requerido sobre la base de las características geométricas del robot. En [5] los autores implementan estos tres movimientos en el controlador industrial CompactRIO (interpolación conjunta, interpolación lineal y arcos circulares), como parte del proceso de reacondicionamiento de un robot manipulador de cinco grados de libertad (5 GDL). Los autores en [6] diseñan e implementan un método de suavizado de ruta de robot que pueda satisfacer la continuidad G2, la interpolación del punto objetivo, la restricción de tolerancia de cuerda, la conservación de la forma y el rendimiento en tiempo real. En el artículo presentado por [7] se realiza una revisión de la situación actual de la planificación de trayectorias para evitar obstáculos espaciales y la planificación de trayectorias de movimiento, en este trabajo se discute el principio básico y la aplicación práctica del método de planificación de la trayectoria del brazo robótico en un ambiente espacial.

Este proyecto se enfoca en la planificación de trayectorias utilizando interpolación lineal y circular. La contribución de este trabajo es realizar una explicación clara, sencilla y sistematizada para la implementación de planeación de trayectorias aplicado a los movimientos de un brazo robot manipulador, en este caso se implementa en un brazo robot SCARA de 2 grados de libertad, sin embargo, no se limita al mismo ya que se puede adaptar fácilmente a cualquier otro tipo de robot con un modelo y desarrollo más complejos.

II. Modelado del sistema

Los robots son en sí mismos sistemas mecánicos, sin embargo, se han desarrollado métodos para representar los aspectos geométricos básicos de la manipulación robótica, así como los aspectos dinámicos de la manipulación. Mediante estos modelos matemáticos, es posible desarrollar métodos para planificar y controlar los movimientos de los robots para realizar tareas específicas.

De acuerdo con [8], los manipuladores robóticos están compuestos por eslabones conectados por articulaciones para formar una cadena cinemática. Las articulaciones suelen ser rotativas (revolutas) o lineales (prismáticas). Una articulación giratoria es como una bisagra y permite la rotación relativa entre dos eslabones. Una junta prismática permite un movimiento relativo lineal entre dos enlaces.

De acuerdo con [9] la cinemática es la parte de la dinámica que describe el movimiento de los cuerpos sin referencia a las fuerzas que lo causan ni a las que se generan a consecuencia de este. La cinemática de un robot corresponde al estudio de los movimientos de un robot, es decir mediante un análisis cinemático ya sea

la posición, velocidad y aceleración de cada uno de los elementos del robot son calculados sin considerar las fuerzas que causan el movimiento [10].

Los autores [11] describen métodos para obtener la cinemática directa e inversa en robot de 1 DoF; del robot planar, antropomórfico y cilíndrico de 3 DoF y 5 DoF, entre otros. La cinemática directa utiliza los parámetros comunes para calcular la configuración de la posición, y la cinemática inversa como bien dice su nombre invierte este cálculo para determinar los parámetros de conjuntos que logra una configuración deseada.

Por su parte, la investigadora Elena García Armada [12] en su trabajo "Los robots y sus capacidades" define el número de grados de movilidad (gdm) de un robot como el número de articulaciones que lo constituyen, por su parte los grados de libertad (DoF, por sus siglas en inglés) se definen como el número de movimientos independientes que puede realizar el extremo efectivo del robot en el espacio tridimensional.

En esta sección se describe el modelo cinemático y dinámico del sistema utilizado. En la Figura 1 se presenta el diagrama esquemático del robot el cual corresponde a un sistema que tiene 2 grados de libertad (DoF, por sus siglas en inglés) en un plano horizontal.

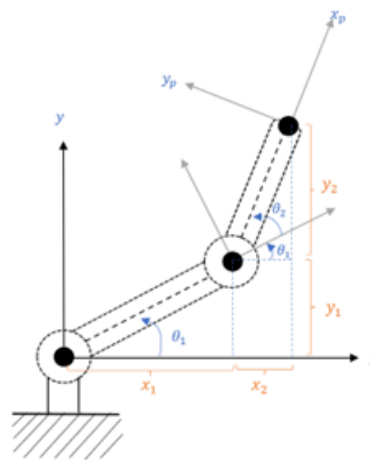


Figura 1: Estructura esquemática del robot de 2 DoF.

Las coordenadas conjuntas utilizadas para el análisis cinemático del sistema son los ángulos θ_1 y θ_2 , que representan la rotación absoluta del primer eslabón y la rotación relativa del segundo eslabón respecto al primero, respectivamente. Los parámetros del sistema se describen en la Tabla 1.

II.1. Cinemática directa

Considere la posición de la base (articulación 1) en $x_0 = 0$ y $y_0 = 0$ y una longitud de los eslabones l_1 y l_2 que se muestran en la Figura 1 donde se presentan los parámetros para el cálculo de la cinemática directa.

Tabla 1: Parámetros del robot de 2 DoF.

Parámetro	Descripción
l_1	Longitud del primer eslabón en cm
l_2	Longitud del segundo eslabón en cm
θ_1	Ángulo de movimiento del primer eslabón respecto al eje x, y
θ_2	Ángulo de movimiento del segundo eslabón respecto al eje x_1, y_1
(x_p, y_p)	Coordenadas del efector final

Como se observa en la Figura 1 en que la posición la articulación 2 en x_1 y y_1 los cuales se pueden obtener a partir de las funciones $\cos(\theta_1)$ y $\sin(\theta_1)$:

$$\cos(\theta_1) = x_1/l_1 \tag{1}$$

$$\sin(\theta_1) = y_1/l_1 \tag{2}$$

A partir de las ecuaciones (1) y (2) se pueden obtener las coordenadas x_1 como y_1 como se muestra a continuación:

$$x_1 = l_1 \cos(\theta_1) \tag{3}$$

$$y_1 = l_1 \sin(\theta_1) \tag{4}$$

Los valores de x_2 así y_2 se encuentran del mismo modo teniendo en cuenta que el ángulo del segundo eslabón está dado por $\theta_1 + \theta_2$ como se muestra en la Figura 1, por lo tanto, las coordenadas x_2 como y_2 están dadas por:

$$x_2 = l_2 \cos(\theta_1 + \theta_2) \tag{5}$$

$$y_2 = l_2 \sin(\theta_1 + \theta_2) \tag{6}$$

La posición final del robot dados por las coordenadas x_p así como y_p las cuales se definen como:

$$x_p = x_1 + x_2 = l_1 \cos \theta_1 + l_2 \cos(\theta_1 + \theta_2) \tag{7}$$

$$y_p = y_1 + y_2 = l_1 \sin \theta_1 + l_2 \sin(\theta_1 + \theta_2) \tag{8}$$

Mediante la cinemática directa se puede encontrar entonces la posición del deflector final definiendo valores de entrada de longitud de los eslabones (l_1 y l_2), las coordenadas de la posición inicial del brazo (x_0, y_0) y los ángulos de las articulaciones (θ_1 y θ_2). En la Figura 2 se presenta el diagrama de flujo para el cálculo de la cinemática inversa.

En la Figura 3 se muestra el resultado de la simulación de la cinemática directa en Matlab. En la simulación se consideran los parámetros $l_1 = 15, l_2 = 15, x_0 = 0, y_0 = 0, \theta_1 = 70$ y $\theta_2 = 60$. En la Figura ?? se representan

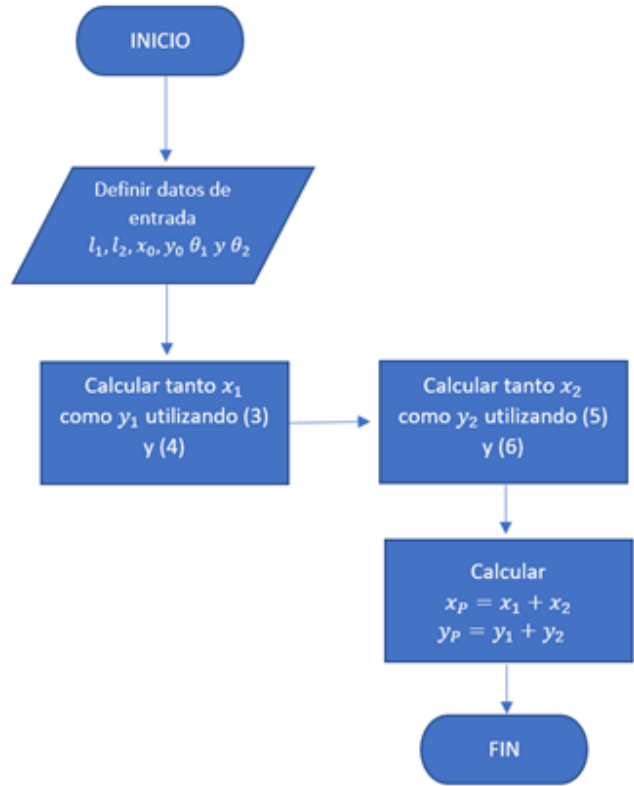


Figura 2: Diagrama de flujo para el cálculo de la cinemática directa.

los eslabones con líneas y las articulaciones con círculos. Se observa que los cálculos entregan un resultado de $x_p = -4.512, y_p = 25.59$ los cuales son correctos y se pueden fácilmente corroborar mediante la gráfica.

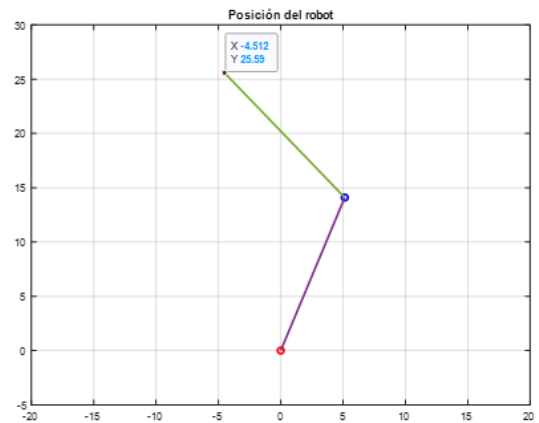


Figura 3: Resultado de la simulación de la cinemática directa en Matlab.

En la Figura 4 se muestra el resultado de la simulación de la cinemática directa considerando los parámetros

$l_1 = 15, l_2 = 15, x_0 = 0, y_0 = 0$ igual que la Figura 2 pero en este caso se utiliza un vector de ángulos con incrementos de 5 en cada tanto para $\theta_1 = \{0, 5, 10, \dots, 180\}$ y $\theta_2 = \{-180, \dots, -5, 0, 5, \dots, 180\}$. Esto se hace con el fin de encontrar el área de trabajo del robot. En este caso en vez de mostrar la posición del robot, se muestra la trayectoria de este.

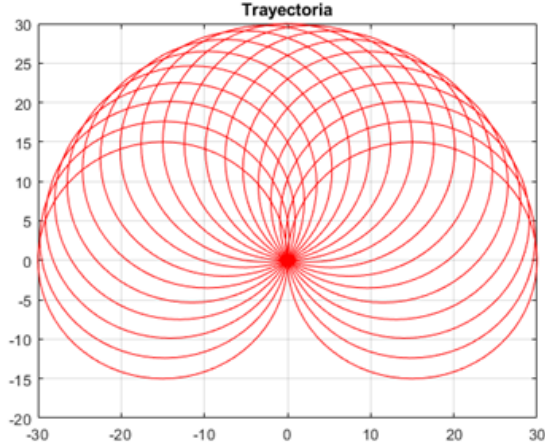


Figura 4: Simulación de trayectoria del robot mediante cinemática directa.

II.2. Cinemática inversa

Para la cinemática inversa, se obtienen las ecuaciones de la posición del efector final para obtener los ángulos de las articulaciones. Es decir, teniendo la posición del efector final (x_p, y_p) , se requiere obtener el valor de los ángulos θ_1 y θ_2 como se muestra en la Figura 1.

Dadas las ecuaciones (7) y (8), se calcula $r^2 = x_p^2 + y_p^2$, siendo r la distancia del origen al efector final teniendo en cuenta que $x_p = x_1 + x_2$ así como $y_p = y_1 + y_2$:

$$x_p^2 + y_p^2 = (l_1 C_1 + l_2 C_1 2)^2 + (l_1 S_1 + l_2 S_1 2)^2 \quad (9)$$

donde por simplicidad se escribe $C_1 = \cos(\theta_1)$, $C_{12} = \cos(\theta_1 + \theta_2)$, $S_1 = \sin(\theta_1)$ y $S_{12} = \sin(\theta_1 + \theta_2)$. Se calcula x_p^2 a partir la ecuación (9) y se tiene:

$$\begin{aligned} x_p^2 &= (l_1 C_1 + l_2 C_1 2)^2 \\ x_p^2 &= (l_1 C_1)^2 + 2(l_1 C_1)(l_2 C_1 2) + (l_2 C_1 2)^2 \\ x_p^2 &= l_1^2 C_1^2 + 2l_1 l_2 C_1 C_1 2 + l_2^2 C_1 2^2 \end{aligned} \quad (10)$$

También se calcula y_p^2 a partir de la ecuación (9) y se tiene que

$$\begin{aligned} y_p^2 &= (l_1 S_1 + l_2 S_1 2)^2 \\ y_p^2 &= (l_1 S_1)^2 + 2(l_1 S_1)(l_2 S_1 2) + (l_2 S_1 2)^2 \\ y_p^2 &= l_1^2 S_1^2 + 2l_1 l_2 S_1 S_1 2 + l_2^2 S_1 2^2 \end{aligned} \quad (11)$$

Sustituyendo (10) y (11) en la ecuación (9) se tiene:

$$\begin{aligned} x_p^2 + y_p^2 &= (l_1^2 C_1^2 + 2l_1 l_2 C_1 C_1 2 + l_2^2 C_1 2^2) + \dots \\ &+ (l_1^2 S_1^2 + 2l_1 l_2 S_1 S_1 2 + l_2^2 S_1 2^2) \end{aligned} \quad (12)$$

Se simplifica el resultado de la ecuación (12) de y se obtiene:

$$\begin{aligned} x_p^2 + y_p^2 &= l_1^2 (C_1^2 + S_1^2) + \dots \\ &+ l_2^2 (C_1 2^2 + S_1 2^2) + \dots \\ &+ 2l_1 l_2 (C_1 C_1 2 + S_1 S_1 2) \end{aligned} \quad (13)$$

Sabiendo que $C_1^2 + S_1^2 = 1$ y $C_1 2^2 + S_1 2^2 = 1$, se puede reducir más la ecuación (13) como sigue:

$$x_p^2 + y_p^2 = l_1^2 + l_2^2 + 2l_1 l_2 C_2 \quad (14)$$

Una vez teniendo lo más simplificada posible, despejamos C_2 :

$$\begin{aligned} l_1^2 + l_2^2 + 2l_1 l_2 C_2 &= x_p^2 + y_p^2 \\ 2l_1 l_2 C_2 &= x_p^2 + y_p^2 - l_1^2 - l_2^2 \\ C_2 = \cos(\theta_2) &= (x_p^2 + y_p^2 - l_1^2 - l_2^2) / (2l_1 l_2) \end{aligned} \quad (15)$$

Se obtiene el ángulo θ_2 de la siguiente manera:

$$\theta_2 = \cos^{-1}[(x_p^2 + y_p^2 - l_1^2 - l_2^2) / (2l_1 l_2)] \quad (16)$$

En la figura 5, se muestra la simulación de la cinemática inversa considerando los parámetros $l_1 = 10, l_2 = 10, x_0 = 0, y_0 = 0$. Después, asignando diferentes valores de $x_p = \{-10, -5, 5, 10\}$ y $y_p = \{-5, -5, -5, -5\}$ para observar el movimiento de los ángulos que calcula Matlab con las ecuaciones de la cinemática inversa según los valores de la posición del efector final.

III. Movimiento y trayectoria predefinida del robot

Cuando una carga bajo control debe seguir un camino específico para llegar desde su punto de partida hasta su punto de parada, los movimientos de los ejes deben estar coordinados o interpolados. Es común hablar de dos tipos de interpolación: lineal y circular.

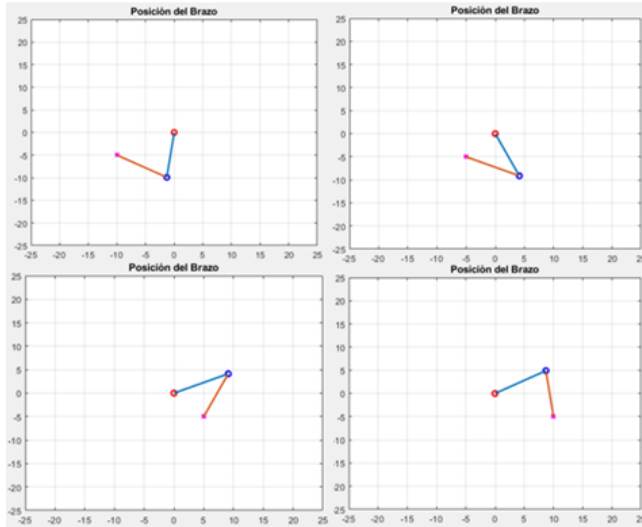


Figura 5: Simulación de la posición del brazo utilizando la cinemática inversa.

III.1. Interpolación lineal

La interpolación lineal es la capacidad de un sistema de control de movimiento que tiene dos o más ejes para mover la carga de un punto a otro en línea recta. El controlador de movimiento debe determinar la velocidad de cada eje para que pueda coordinar sus movimientos. La interpolación lineal verdadera requiere que el controlador de movimiento modifique la aceleración del eje, pero algunos controladores se aproximan a la interpolación lineal verdadera con perfiles de aceleración programados. La ruta puede estar en un plano o ser tridimensional. Para la interpolación lineal en términos de código G se utiliza comúnmente G01. La Fórmula de interpolación lineal es el proceso de encontrar un valor entre dos puntos en una línea. Esta herramienta, la interpolación, no solo es útil en estadísticas, sino que también es útil en ciencia, negocios o en cualquier momento en que sea necesario predecir valores que se encuentren dentro de dos puntos de datos existentes. Considerando que se quiere realizar un movimiento en un plano desde el punto inicial $P_1 = (x_1, y_1)$ al punto final $P_2 = (x_2, y_2)$. El valor desconocido en un punto se encuentra usando esta fórmula:

$$y = \frac{(y_2 - y_1)}{(x_2 - x_1)}(x - x_1) + y_1 \quad (17)$$

donde la ecuación (11) representa a la ecuación de la recta de la forma $y = mx + b$ con pendiente $m = \frac{y_2 - y_1}{x_2 - x_1}$ y cruce por $b = y_1$, el término $x - x_1$ se debe a que la recta no está trazada sobre las coordenadas (0,0), sino que a partir del punto inicial (x_1, y_1) . Consideremos por ejemplo que el $P_1 = (0, 3)$ y $P_2 = (6, 3)$ como se muestra en la Figura 7 donde el punto P_1 se ilustra con un cuadro de color rojo y el punto

P_2 con un círculo de color azul.

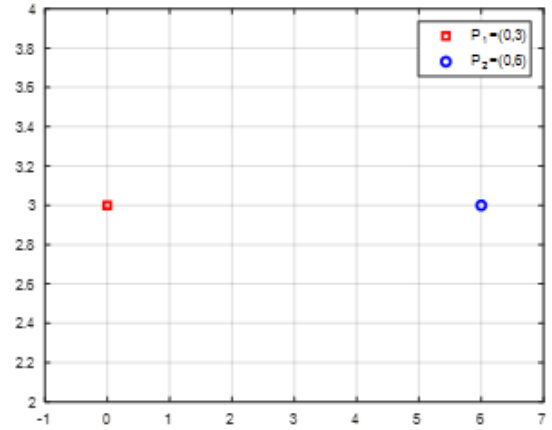


Figura 6: Definición del punto inicial y punto final de la interpolación lineal.

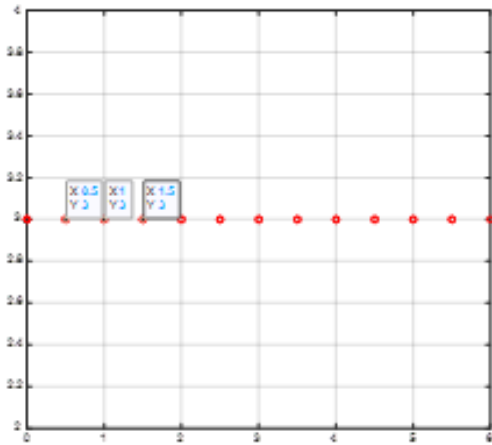
Podemos definir una resolución en x como $\delta x = \{0.5, 1\}$, utilizando la ecuación (17) encontramos los puntos δy . Como se muestra en la Figura 7 cuanto menor es el valor de δx , la interpolación se realiza en puntos más cercanos, podemos llamar al valor δx como la resolución de la interpolación.

Ahora bien, cuando utilizamos la interpolación para realizar el movimiento del robot, debemos ajustar que el movimiento siempre tenga una misma resolución cualquiera sea el sentido, en el caso por ejemplo de $P_1 = (6, 6)$ y $P_2 = (10, 9)$ se realiza un movimiento en diagonal, entonces debemos ajustar el valor δx para que se respete la resolución en la trayectoria dada. Esto se puede realizar obteniendo la distancia del movimiento en diagonal, la cual se define por el teorema de Pitágoras por $c = \sqrt{a^2 + b^2}$. Donde c representa la longitud de la hipotenusa, a y b son los lados del cateto.

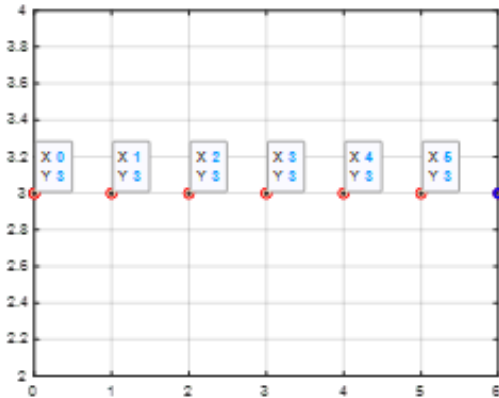
Para nuestro caso particular se definen como $a = |y_2 - y_1|$ y $b = |x_2 - x_1|$, donde $|x|$ representa el valor absoluto de x . En el caso que se comentó donde $P_1 = (6, 6)$ y $P_2 = (10, 9)$, los valores de los catetos están dados por $a = |9 - 6| = 3$ y $b = |10 - 6| = 4$ por ende $c = \sqrt{3^2 + 4^2} = \sqrt{25} = 5$. Realizando este factor de ajuste siempre que $y_1 \neq x_2$ tenemos el resultado que se muestra en la Figura 8. Se observa que en este caso $\delta x = 1$ ya que tenemos 5 puntos desde el punto inicial al punto final lo que corresponde al valor calculado de $c = 5$, entonces $\delta x_{ajustado}$ está dado de manera general por:

$$\delta x_{ajustado} = \frac{\delta x}{(c/b)} \quad (18)$$

Tenemos entonces en nuestro caso particular, $\delta x_{ajustado} = \frac{1}{5/4} = 0.8$ como se observa en la Fi-



(a) Interpolación lineal con $\delta x = 0.5$



(b) Interpolación lineal con $\delta x = 1$

Figura 7: Interpolación lineal entre dos puntos con diferente resolución δx .

gura 9 donde los incrementos en x están dados por $x = \{6, 6.8, 7.6, 8.4, 9.2, 10\}$.

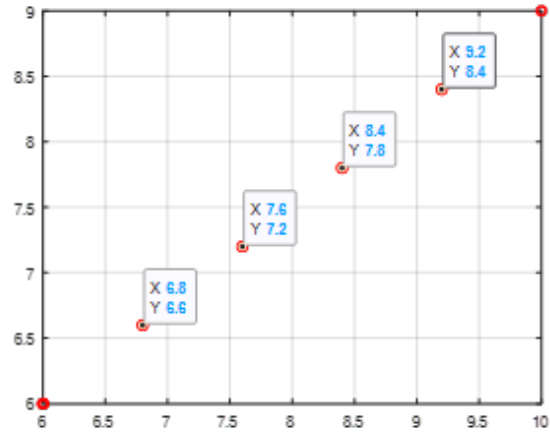


Figura 8: Interpolación lineal entre dos puntos con ajuste de δx .

La fórmula de interpolación de la ecuación (17) se puede utilizar siempre y cuando $x_1 \neq x_2$, ya que si $x_1 = x_2$ se tiene una indeterminación a la hora de calcular la pendiente m , que de ahora en adelante le renombraremos como m_x por el hecho de que se calcula la pendiente con respecto al eje x . Para encontrar la solución en los casos donde $x_1 = x_2$, se puede realizar la interpolación a través del eje de las ordenadas. La fórmula de interpolación, despejando la ecuación (17) entonces estará dada por:

$$x = \frac{(x_2 - x_1)(y_1 - y)}{(y_1 - y_2)} + x_1 \quad (19)$$

donde $m_y = -m_x = \frac{(y_1 - y_2)}{(x_2 - x_1)}$, así como la resolución $\delta y = \delta x$. En la figura 10 se muestra la interpolación en el eje y utilizando $\delta y = 1$.

En la Figura 10 se muestra la interpolación entre los puntos $P_1 = (6, 6)$ y $P_2 = (14, 12)$ con $\delta x = 2$, se definen los puntos de interpolación con una “x” y se puede observar el movimiento del brazo robot graficando los ángulos θ_1 y θ_2 mediante la obtención de la cinemática inversa en cada punto de interpolación. Cabe recalcar que cuanto menor es la resolución (δx) más suaves serán los movimientos del robot.

En la Tabla 2 se muestran los valores obtenidos en cada punto de interpolación y los valores de los ángulos θ_1 y θ_2 obtenidos de la cinemática inversa, puede observar que en este caso el $\delta x = 2$ y $\delta x_{ajustado} = 1.6$.

III.2. Interpolación circular

La interpolación circular es la capacidad de un sistema de control de movimiento que tiene dos o más ejes para mover la carga alrededor de una trayectoria circular. La

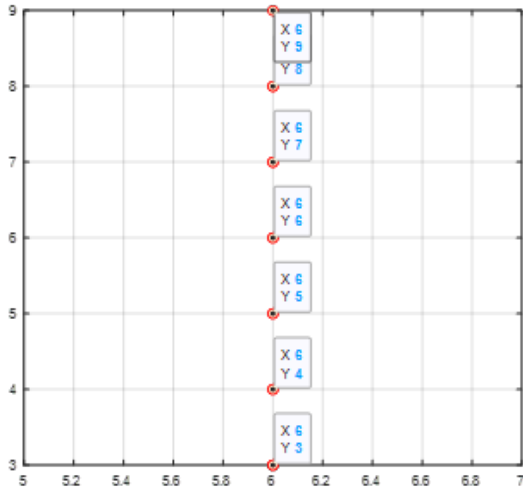


Figura 9: Interpolación lineal en el eje y considerando $\delta y = 1$.

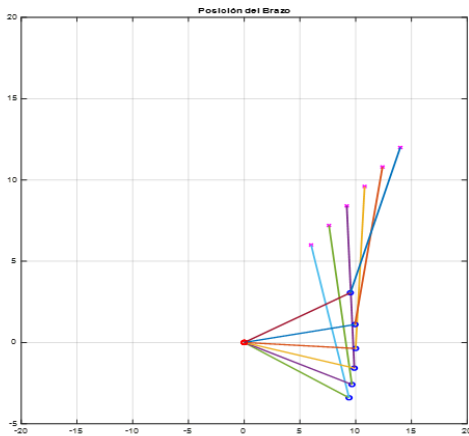


Figura 10: Movimiento de interpolación lineal del brazo robot de 2DoF.

interpolación circular se puede realizar en dos sentidos, en sentido horario y en sentido antihorario. A la interpolación en sentido horario se le identifica con el código G mediante G02 y a la interpolación en sentido antihorario se le identifica con el código G03.

Para la interpolación circular se considera (x_i, y_i) la posición inicial (start point), R el radio del centro del círculo (center) y (x_f, y_f) la posición final (end point) como se muestra en la figura 11.

A continuación, se describe paso a paso la realización de la interpolación circular:

1. Se calculan (x_c, y_c) , es decir, la posición del centro de la circunferencia.

Tabla 2: Valores para el movimiento del brazo robot mediante interpolación

Valor en eje x	Valor en eje y	θ_1	θ_2
6.0	6.0	-19.89	129.79
7.6	7.2	-14.98	116.87
9.2	8.4	-9.07	102.94
10.8	9.6	-2.1	87.47
12.4	10.8	6.35	69.39
14.0	12.0	17.81	45.57

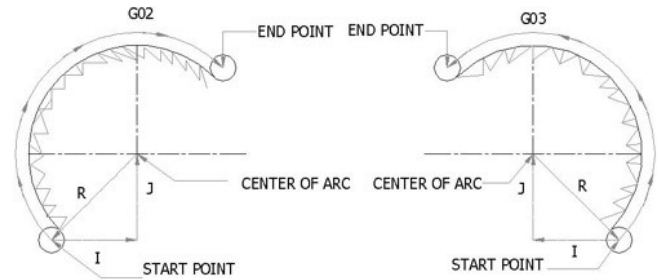


Figura 11: Interpolación circular.

$$\begin{aligned} x_c &= (x_i + x_f)/2 \\ y_c &= (y_i + y_f)/2 \end{aligned} \quad (20)$$

2. Se calcula el ángulo inicial (θ_i) y el ángulo final (θ_f):

$$\begin{aligned} \theta_i &= \text{atan2}(y_1 - y_c, x_1 - x_c) \\ \theta_f &= \text{atan2}(y_2 - y_c, x_2 - x_c) \end{aligned} \quad (21)$$

Obsérvese que en este caso se utiliza la función arco-tangente de dos parámetros, $\text{atan2}(a, b)$ en donde los parámetros a, b se utilizan para saber en qué cuadrante nos encontramos dependiendo del signo de cada parámetro.

3. Determinar si la dirección de rotación ya sea en sentido horario o en sentido antihorario, para ello es necesario realizar la verificación:

$$\begin{aligned} \text{Si } \theta_i < \theta_f, & \text{ la dirección es antihoraria} \\ \text{Si } \theta_i > \theta_f, & \text{ la dirección es horaria} \end{aligned} \quad (22)$$

4. Generar los puntos de interpolación, es decir los puntos intermedios, sabiendo los ángulos θ_i y θ_f se puede dividir en segmentos iguales y se encuentra el ángulo θ_m . Para respetar la velocidad de movimiento con respecto a la interpolación lineal se debe ajustar δx considerando la longitud del arco formado entre los puntos (x_i, y_i) y (x_f, y_f) y así obtener el $\delta\theta_m$, que corresponde al incremento de ángulos en la interpolación circular con respecto a θ_m . La longitud del arco se calcula como sigue:

$$L_{\text{arco}} = 2\pi R \frac{|\theta_i - \theta_f|}{360} \quad (23)$$

5. Habiendo calculado la longitud del arco se calcula la cantidad de divisiones o ángulos a recorrer entre θ_1 y θ_2 :

$$N_{div} = \frac{L_{arco}}{\delta x} \quad (24)$$

6. Entonces encontramos el incremento de ángulos $\delta\theta_m$ como sigue:

$$\delta\theta_m = \frac{|\theta_i - \theta_f|}{N_{div}} \quad (25)$$

7. Teniendo $\delta\theta_m$ se puede calcular el primer ángulo:

$$\theta_m = \theta_i + \delta\theta_m, \forall m \in 1, 2, \dots, N_{div} \quad (26)$$

8. Teniendo θ_m se deben calcular las coordenadas (x_m, y_m) :

$$\begin{aligned} x_m &= x_c + R \cos(\theta_m) \\ y_m &= y_c + R \sin(\theta_m) \end{aligned} \quad (27)$$

9. Teniendo las coordenadas del punto intermedio (x_m, y_m) al cual deseamos movernos desde $(x_i, y_i), \dots, (x_m, y_m), \dots, (x_2, y_2)$ se calcula la cinemática inversa para obtener los valores θ_1 y θ_2 que corresponden a los ángulos el cual deben moverse de los eslabones.

IV. Implementación y Resultados

La planificación de la trayectoria es para controlar el movimiento del brazo robot a lo largo de cierta trayectoria objetivo, para garantizar el seguimiento de la trayectoria del objetivo con un control de alta precisión ha sido el contenido principal del control de trayectoria del brazo manipulador. En la Figura 12 se describe el proceso de control de la planeación de trayectoria del brazo robot.

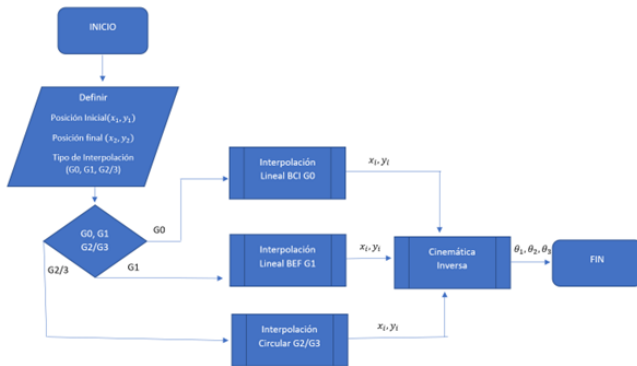


Figura 12: Diagrama de flujo de la planeación de trayectorias del brazo robot.

Se observa que al principio se deben definir las posiciones, tanto inicial en el punto (x_i, y_i) y la posición final en el punto (x_f, y_f) , obsérvese que no se define la posición

en el eje z , debido a que este va a depender del tipo de interpolación. En la interpolación lineal basada en coordenadas internas (BCI), se debe definir la posición z para levantar el brazo robot, por otro lado, en la interpolación lineal basada en el efector final (BEF) se debe definir posición z para bajar el brazo robot y realizar el trabajo. Después de las posiciones de paso en la interpolación definidas por (x_m, y_m) se obtiene la cinemática inversa para calcular los ángulos θ_1, θ_2 y θ_3 .

Como se observa en la Figura 12, para poder realizar la interpolación a considerar en la trayectoria del robot se requiere la posición inicial y final, así como el tipo de movimiento referente a:

- G0: Movimiento rápido de interpolación BCI.
- G1: Interpolación lineal BEF.
- G2/3: Interpolación circular BEF.

En la figura 13 se muestra que se utilizará de ejemplo en donde se definen las dimensiones y los movimientos a través del Código G.

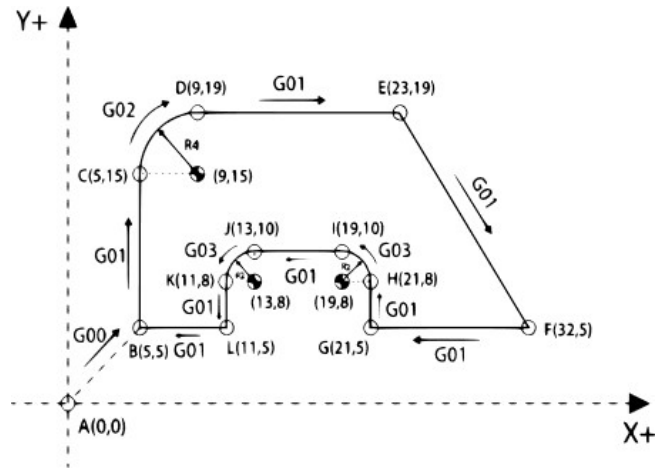


Figura 13: Figura base para definir el código G

A partir de la figura anterior, se define el código G resultante. Esto se realiza definiendo la posición inicial, final y el tipo de interpolación necesario en cada caso. En la Tabla 3 se observa este proceso de obtención del código G a partir de la figura 13.

En la Figura 14, se muestra la simulación en el software NC corrector el cual es un editor visualizador de programas gratuito para máquinas CNC (código G). En esta es posible corroborar que el código G ha sido ingresado de manera correcta para cumplir el requisito propuesto en la figura 14.

En la figura 15 se muestra la simulación en Matlab de la interpolación lineal y circular, en línea punteada se observa el movimiento G0, en línea continua de color rojo se muestra el movimiento G1 y en línea continua color azul se muestra el movimiento G2/G3 las unidades se

Tabla 3: Posición inicial, final y código G

Posición Inicial	Posición Final	Código G
(0,0)	(5,5)	G0 X5 Y5 Z1
(5,5)	(5,15)	G1 X5 Y15 Z-1
(5,15)	(9,19)	G2 X9 Y19 I4 J0
(9,19)	(23,19)	G1 X23 Y19
(23,19)	(32,5)	G1 X32 Y5
(32,19)	(21,5)	G1 X21 Y5
(21,5)	(21,8)	G1 X21 Y8
(21,8)	(19,10)	G3 X19 Y10 I-2 J0
(19,10)	(13,10)	G1 X13 Y10
(13,10)	(11,8)	G3 X11 Y8 I0 J-2
(11,8)	(11,5)	G1 X11 Y5
(11,5)	(5,5)	G1 X5 Y5 Z1

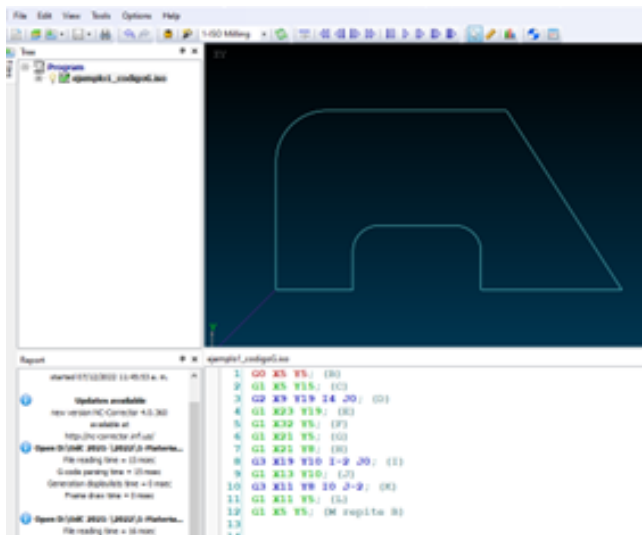


Figura 14: Simulación de código G en el software NC Corrector.

consideraron en cm por lo tanto se tuvieron que convertir a mm como se observa que el movimiento a la primera posición es (50, 50).

V. Conclusiones

En este artículo, se abordó una metodología para la planificación de trayectorias en robots SCARA de 2 grados de libertad, resaltando su aplicabilidad en entornos académicos, especialmente en el ámbito de la robótica y mecatrónica. La detallada descripción de la cinemática del robot, la implementación de algoritmos de interpolación lineal y circular, y la integración con el código G para proporcionar un funcionamiento correcto de un robot para seguir trayectorias suaves. Los ejemplos prácticos en Matlab permiten visualizar y experimentar con conceptos abstractos, facilitando la comprensión de los principios fundamentales de la planificación de trayectorias en el

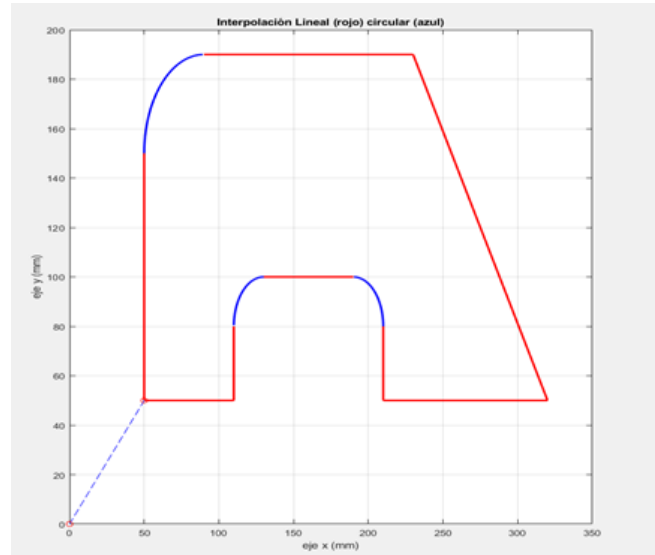


Figura 15: Simulación en Matlab de interpolación lineal y circular.

contexto de la robótica. Este enfoque didáctico, centrado en la aplicación práctica, no solo transmite los aspectos teóricos de la planificación de movimientos, sino que también promueve el desarrollo de habilidades prácticas para abordar desafíos reales referente al movimiento de Robots. La versatilidad de esta metodología puede enriquecer los programas académicos al proporcionar herramientas prácticas y conocimientos esenciales para resolver problemas en constante evolución en estos campos.

Referencias

- [1] G. F. Chen et al. «SCARA robot control system design and trajectory planning: a case study». En: *Advances in Electrical Engineering and Automation AISC* Vol. 139 (2012), págs. 171-176.
- [2] Patrik Axelson. *Simulation Model of a 2 Degrees of Freedom Industrial Manipulator*. Technical Report. Automatic Control at Linköpings Universitet, 2011.
- [3] Zhang Zhiyong et al. «Picking Robot Arm Trajectory Planning Method». En: *Sensors and Transducers* 162 (ene. de 2014), págs. 11-20.
- [4] Giovanni Incerti. «Motion Planning of SCARA Robots for Trajectory Tracking». En: *International Science Index on Excellence in Research and Innovation for Humanity* (2015).

- [5] Jorge Luis Aroca Trujillo, Alexander Pérez-Ruiz y Ruthber Rodríguez Serrezuela. «Generation and Control of Basic Geometric Trajectories for a Robot Manipulator Using CompactRIO». En: *Journal of Robotics* 2017 (2017), págs. 1-11.
- [6] Shanshan He et al. «A tolerance constrained G2 continuous path smoothing and interpolation method for industrial SCARA robots». En: *Robotics and Computer-Integrated Manufacturing* 63 (2020), pág. 101907. ISSN: 0736-5845.
- [7] Ye Dai et al. «A Review of Spatial Robotic Arm Trajectory Planning». En: *Aerospace* 9.7 (2022), pág. 361.
- [8] Mark W Spong, Seth Hutchinson y Mathukumalli Vidyasagar. *Robot Modeling and Control*. 2nd. John Wiley & Sons Inc., 2020.
- [9] J. L. Meriam y L. G. Kraige. *Mecánica para Ingenieros. Dinámica II Volumen 2*. Editorial Reverté S. A., 1998.
- [10] John J. Craig. *Robótica*. México: Pearson Education, 2006.
- [11] M. H. Hernández Ordoñez et al. *Robótica: análisis, modelado, control e implementación*. Omnia Publisher SL, 2015.
- [12] E. G. Armada. *Los robots y sus capacidades*. CSIC y Catarata, 2022.