

Interpolation strategies for motion planning on a 2-DOF SCARA robot

Estrategias de interpolación para la planificación de trayectorias en un robot SCARA de 2 GDL

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 articulado de cumplimiento selectivo (SCARA) son ampliamente utilizados en aplicaciones industriales, especialmente en tareas de ensamblaje y manipulación, debido a su rapidez, precisión y simplicidad mecánica [1]. Estos robots presentan ventajas considerables en entornos donde se requiere un

desplazamiento plano eficiente y controlado.

En el contexto educativo, los robots SCARA también ofrecen un escenario ideal para introducir conceptos clave de cinemática, interpolación y planificación de trayectorias, ya que su estructura plana simplifica el análisis sin sacrificar la complejidad de los problemas de control. Este trabajo se centra en la planificación de trayectorias utilizando algoritmos de interpolación lineal y circular, considerando su aplicación en un robot SCARA de dos grados de libertad (DoF).

^{*} Autor de correspondencia

El objetivo principal es proporcionar una metodología clara, sistemática y didáctica para generar trayectorias suaves que puedan ser seguidas por un brazo robótico, integrando tanto la cinemática directa e inversa como la codificación en lenguaje G (G-code), comúnmente utilizado en sistemas CNC. Además, se incorpora el uso del software *NC Corrector*, un visualizador gratuito de trayectorias, para validar los resultados obtenidos. Esta combinación de elementos busca facilitar la comprensión práctica de los principios fundamentales del movimiento robótico, haciendo de esta propuesta una herramienta útil tanto en entornos académicos como en aplicaciones de prototipado y validación de trayectorias.

II. Estado del arte

Diversos estudios han abordado la planificación de trayectorias en robots manipuladores, con especial atención a la interpolación y control de movimiento. En [2], se presenta un modelo cinemático y dinámico, junto con una descripción del perfil de aceleración para un manipulador industrial de 2 DoF. Por su parte, [1] propone un controlador de bajo costo y alta calidad para un robot SCARA, derivando las ecuaciones cinemáticas y utilizando interpolación circular mediante el método del analizador diferencial digital vectorial.

En [3], se implementa un brazo manipulador capaz de ejecutar rutas específicas bajo restricciones definidas por el diseño del robot. En cuanto a la generación de leyes de movimiento para actuadores, [4] desarrolla un programa basado en las características geométricas del robot. Asimismo, [5] emplea interpolación conjunta, lineal y circular en un entorno CompactRIO para controlar un robot manipulador de cinco grados de libertad.

En el ámbito del suavizado de trayectorias, [6] propone un método que garantiza continuidad G2, precisión geométrica y rendimiento en tiempo real. Finalmente, [7] realiza una revisión detallada sobre métodos actuales de planificación de trayectorias, enfocándose en la evasión de obstáculos y en la implementación práctica de estas técnicas en entornos tridimensionales.

A diferencia de los trabajos revisados, que se centran principalmente en la optimización de trayectorias o en la implementación de controladores industriales, la presente propuesta enfatiza el aspecto didáctico y formativo de la planeación de trayectorias en robots SCARA. Además, integra de manera novedosa el uso combinado del código G y el entorno Matlab como herramientas accesibles para la simulación, visualización y validación del movimiento, lo que permite una comprensión más intuitiva de los principios cinemáticos y de interpolación. Esta integración práctica, orientada a la enseñanza y prototipado, contrasta con los enfoques más técnicos o puramente teóricos de los estudios previos, reforzando el valor educativo de

la metodología desarrollada.

III. 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 en [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 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.

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

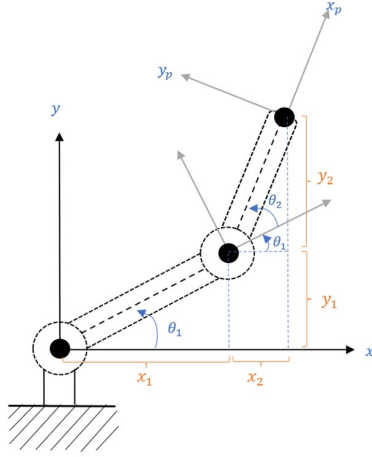


Figura 1: Estructura esquemática del robot de 2 DoF.

en la Tabla 1.

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

III.1. Cinemática directa

La cinemática directa de un manipulador puede obtenerse a través de diferentes métodos matemáticos, entre los cuales se encuentran los enfoques geométrico, matricial y algebraico. En este caso, el procedimiento utilizado se basa en las relaciones trigonométricas entre las longitudes de los eslabones y los ángulos articulares, por lo que puede considerarse un enfoque geométrico con formulación algebraica, ya que parte del análisis geométrico del manipulador y expresa las relaciones de posición mediante ecuaciones trigonométricas.

Considérese la posición de la base (articulación 1) en $x_0 = 0$ y $y_0 = 0$, con longitudes de los eslabones l_1 y l_2 , como se muestra en la Figura 1, donde se presentan los parámetros necesarios para el cálculo de la cinemática directa. En dicha figura, la posición de la articulación 2 se representa mediante las coordenadas (x_1, y_1) , las cuales se obtienen directamente a partir de las funciones trigonométricas del ángulo de la primera articulación θ_1 :

$$\cos(\theta_1) = \frac{x_1}{l_1} \quad (1)$$

$$\sin(\theta_1) = \frac{y_1}{l_1} \quad (2)$$

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

$$x_1 = l_1 \cos(\theta_1) \quad (3)$$

$$y_1 = l_1 \sin(\theta_1) \quad (4)$$

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:

$$x_2 = l_2 \cos(\theta_1 + \theta_2) \quad (5)$$

$$y_2 = l_2 \sin(\theta_1 + \theta_2) \quad (6)$$

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

$$x_p = x_1 + x_2 = l_1 \cos \theta_1 + l_2 \cos(\theta_1 + \theta_2) \quad (7)$$

$$y_p = y_1 + y_2 = l_1 \sin \theta_1 + l_2 \sin(\theta_1 + \theta_2) \quad (8)$$

Mediante la cinemática directa se puede encontrar entonces la posición del efector 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 directa.

En la Figura 3 se muestra el resultado de la simulación de la cinemática directa realizada en Matlab, considerando los siguientes parámetros: $l_1 = 15$, $l_2 = 15$, $x_0 = 0$, $y_0 = 0$, $\theta_1 = 70^\circ$ y $\theta_2 = 60^\circ$. En la Figura 3, los eslabones del robot se representan mediante líneas, mientras que las articulaciones se indican con círculos. Los cálculos arrojan un resultado para la posición del efector final de $x_p = -4.512$ y $y_p = 25.59$, valores que pueden corroborarse fácilmente mediante la inspección visual de la gráfica.

En la Figura 4 se presenta el resultado de la simulación de la cinemática directa considerando los parámetros $l_1 = 15$, $l_2 = 15$, $x_0 = 0$ y $y_0 = 0$. Para este análisis, se emplea un vector de ángulos con incrementos de 5° en ambos ejes articulares: $\theta_1 = 0^\circ, 5^\circ, 10^\circ, \dots, 180^\circ$ y $\theta_2 = -180^\circ, -175^\circ, \dots, 0^\circ, 5^\circ, \dots, 180^\circ$. El propósito de esta simulación es determinar el área de trabajo del robot. En lugar de representar posiciones específicas del brazo, se muestra la trayectoria resultante del movimiento del efector final.

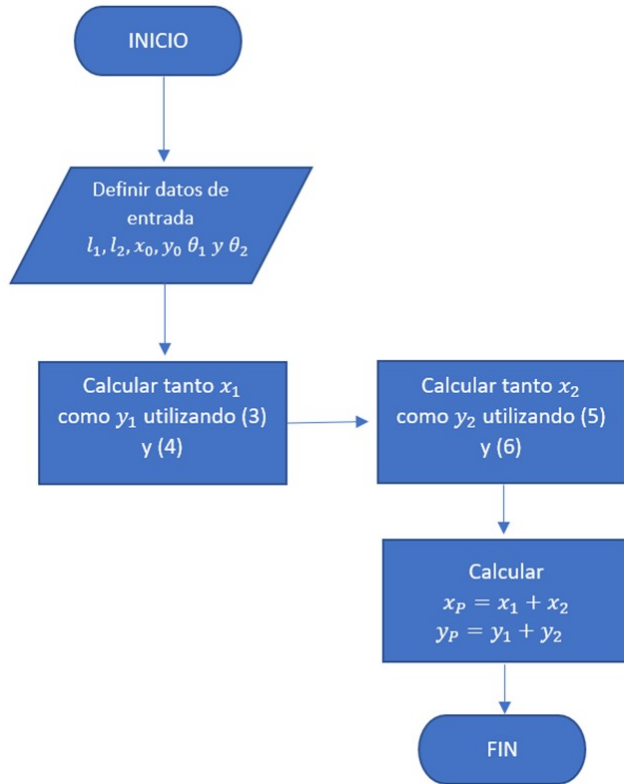


Figura 2: Diagrama de flujo para el cálculo de la cinemática directa.

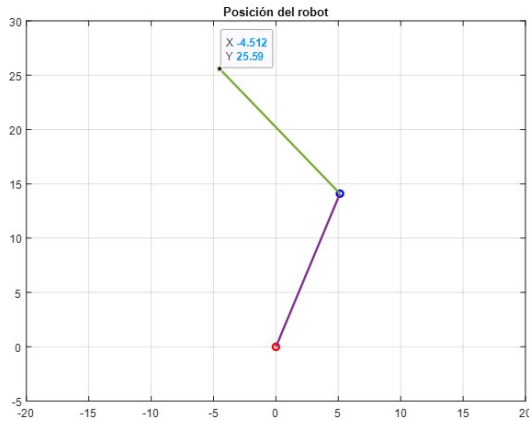


Figura 3: Resultado de la simulación de la cinemática directa en Matlab.

III.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.

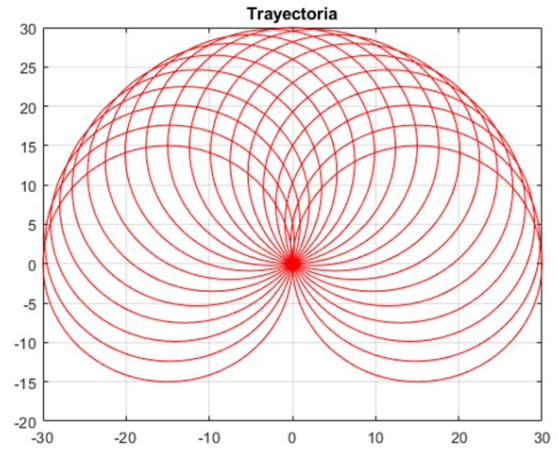


Figura 4: Simulación de trayectoria del robot mediante cinemática directa.

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 $y_p = y_1 + y_2$:

$$x_p^2 + y_p^2 = (l_1 C_1 + l_2 C_{12})^2 + (l_1 S_1 + l_2 S_{12})^2 \quad (9)$$

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:

$$\begin{aligned} x_p^2 &= (l_1 C_1 + l_2 C_{12})^2 \\ x_p^2 &= (l_1 C_1)^2 + 2(l_1 C_1)(l_2 C_{12}) + (l_2 C_{12})^2 \\ x_p^2 &= l_1^2 C_1^2 + 2l_1 l_2 C_1 C_{12} + l_2^2 C_{12}^2 \end{aligned} \quad (10)$$

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

$$\begin{aligned} y_p^2 &= (l_1 S_1 + l_2 S_{12})^2 \\ y_p^2 &= (l_1 S_1)^2 + 2(l_1 S_1)(l_2 S_{12}) + (l_2 S_{12})^2 \\ y_p^2 &= l_1^2 S_1^2 + 2l_1 l_2 S_1 S_{12} + l_2^2 S_{12}^2 \end{aligned} \quad (11)$$

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

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

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

$$\begin{aligned} x_p^2 + y_p^2 &= l_1^2 (C_1^2 + S_1^2) + \dots \\ &+ l_2^2 (C_{12}^2 + S_{12}^2) + \dots \\ &+ 2l_1 l_2 (C_1 C_{12} + S_1 S_{12}) \end{aligned} \quad (13)$$

Sabiendo que $C_1^2 + S_1^2 = 1$ y $C_{12}^2 + S_{12}^2 = 1$, se puede reducir más la ecuación (13) como sigue:

$$x_p^2 + y_p^2 = l_1^2 + l_2^2 + 2l_1l_2C_2 \quad (14)$$

Una vez teniendo lo más simplificada posible, despejamos C_2 :

$$\begin{aligned} l_1^2 + l_2^2 + 2l_1l_2C_2 &= x_p^2 + y_p^2 \\ 2l_1l_2C_2 &= x_p^2 + y_p^2 - l_1^2 - l_2^2 \\ C_2 &= \cos(\theta_2) = (x_p^2 + y_p^2 - l_1^2 - l_2^2) / (2l_1l_2) \end{aligned} \quad (15)$$

Se obtiene el ángulo θ_{22} de la siguiente manera:

$$\theta_2 = \cos^{-1}[(x_p^2 + y_p^2 - l_1^2 - l_2^2) / (2l_1l_2)] \quad (16)$$

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.

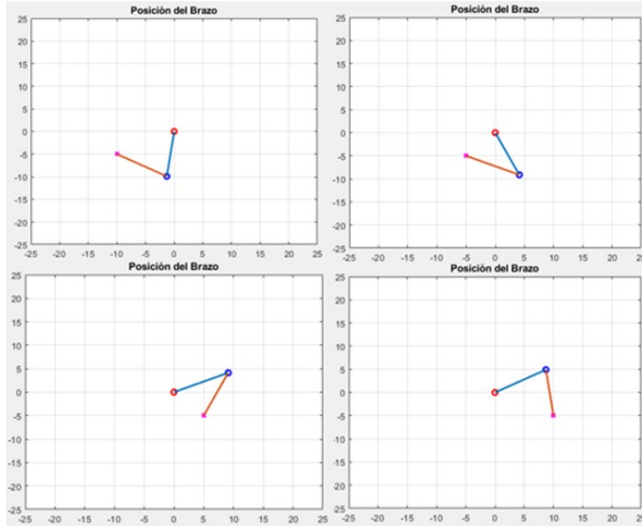


Figura 5: Simulación de la posición del brazo utilizando la cinemática inversa.

IV. 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.

IV.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 (17) 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 6 donde el punto P_1 se ilustra con un cuadro de color rojo y el punto P_2 con un círculo de color azul.

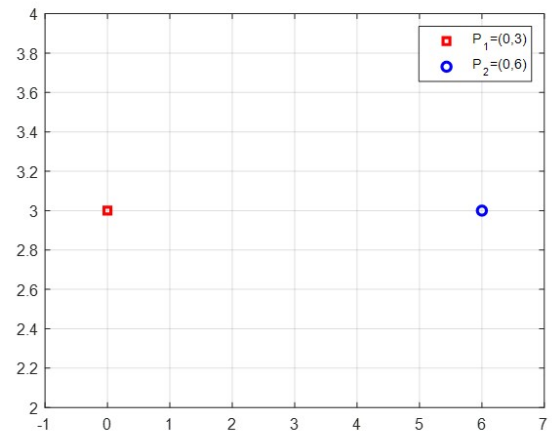
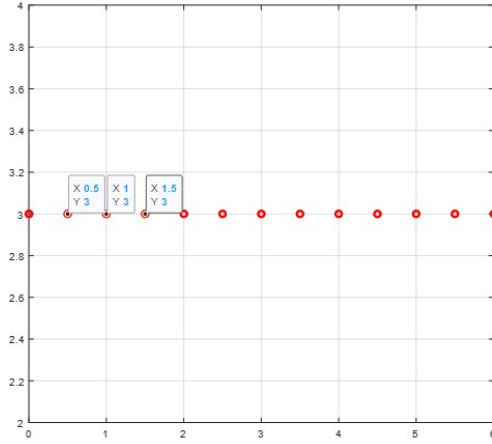
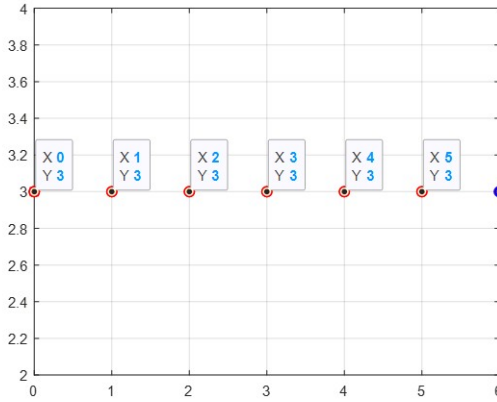


Figura 6: Definición del punto inicial y punto final de la interpolación lineal.

Podemos definir una resolución en x como $\delta x =$

(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 .

$\{0.5, 1\}$, 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.

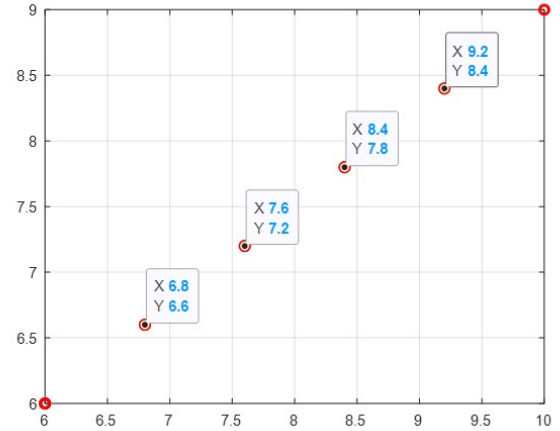
Ahora bien, cuando utilizamos la interpolación para realizar el movimiento del robot, debemos ajustar que el movimiento siempre tenga una misma resolución cualquiera sea el sentido, en el caso por ejemplo de $P_1 = (6, 6)$ y $P_2 = (10, 9)$ se realiza un movimiento en diagonal, entonces debemos ajustar el valor δ_x 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 Figura 8 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 9 se muestra la interpolación en el eje y 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

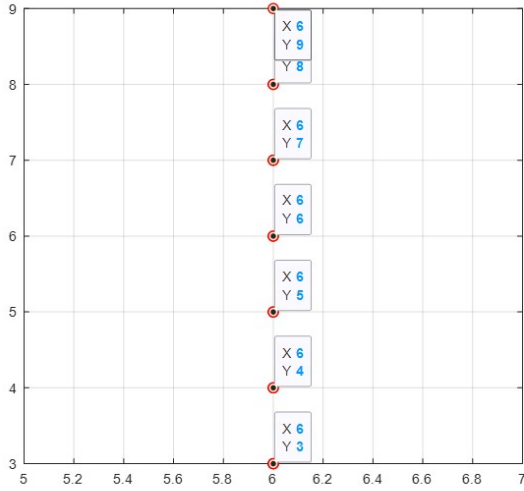


Figura 9: Interpolación lineal en el eje y considerando $\delta y = 1$.

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.

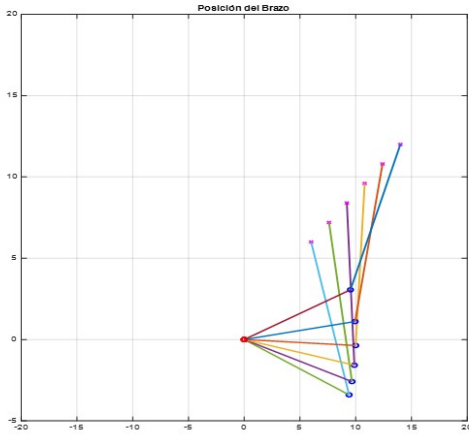


Figura 10: Movimiento de interpolación lineal del brazo robot de 2DoF.

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 $\delta x_{ajustado} = 1.6$.

IV.2. Interpolación circular

La interpolación circular es la capacidad de un sistema de control de movimiento que tiene dos o más ejes para

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

mover la carga alrededor de una trayectoria circular. La 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.

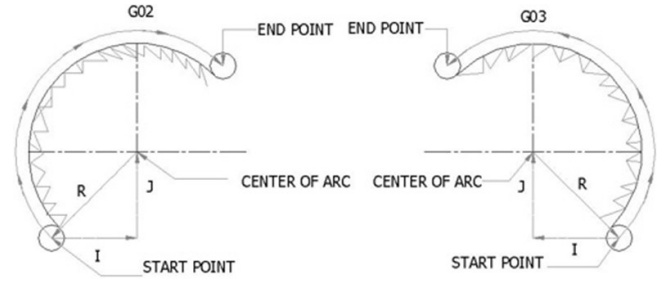


Figura 11: Interpolación circular.

A continuación, se describe paso a paso el procedimiento para realizar la interpolación circular:

1. Se calcula el centro de la circunferencia, (x_c, y_c) , a partir de las coordenadas iniciales y finales del arco:

$$x_c = \frac{x_i + x_f}{2} \quad (20)$$

$$y_c = \frac{y_i + y_f}{2}$$

2. Se determinan el ángulo inicial ϕ_i y el ángulo final ϕ_f mediante la función arco-tangente de dos argumentos:

$$\phi_i = \text{atan2}(y_i - y_c, x_i - x_c) \quad (21)$$

$$\phi_f = \text{atan2}(y_f - y_c, x_f - x_c)$$

Obsérvese que en este caso se utiliza la función arco-tangente de dos parámetros, $\text{atan2}(a, b)$, la cual permite determinar correctamente el cuadrante del ángulo resultante de acuerdo con el signo de cada parámetro.

- Determinar la dirección de rotación, ya sea en sentido horario o antihorario. Para ello, se realiza la siguiente verificación:

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

- Generar los puntos de interpolación, es decir, los puntos intermedios. Conociendo los ángulos ϕ_i y ϕ_f , se divide el arco en segmentos iguales para obtener el ángulo intermedio ϕ_m . Para respetar la velocidad de movimiento respecto a la interpolación lineal, se ajusta δx considerando la longitud del arco formado entre los puntos (x_i, y_i) y (x_f, y_f) , y así obtener $\delta\phi_m$, que corresponde al incremento angular en la interpolación circular. La longitud del arco se calcula como:

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

- Habiendo calculado la longitud del arco, se determina la cantidad de divisiones o ángulos a recorrer entre ϕ_i y ϕ_f :

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

- Se encuentra el incremento angular $\delta\phi_m$ como sigue:

$$\delta\phi_m = \frac{|\phi_i - \phi_f|}{N_{\text{div}}} \quad (25)$$

- Teniendo $\delta\phi_m$, se calcula el primer ángulo intermedio:

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

- A partir de ϕ_m , se calculan las coordenadas de los puntos intermedios (x_m, y_m) :

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

- Finalmente, con las coordenadas del punto intermedio (x_m, y_m) al cual se desea mover desde (x_i, y_i) hasta (x_f, y_f) , se calcula la cinemática inversa para obtener los valores de los ángulos articulares correspondientes, ϕ_1 y ϕ_2 , que definen el movimiento de los eslabones.

V. Implementación y Resultados

La planificación de trayectorias tiene como objetivo controlar el movimiento del brazo robótico a lo largo de una trayectoria definida, garantizando un seguimiento preciso de dicha ruta. Este proceso constituye uno de

los aspectos fundamentales en el control de trayectoria de un brazo manipulador. En la Figura 12 se ilustra el procedimiento seguido para la planeación y control de trayectorias del brazo robot.

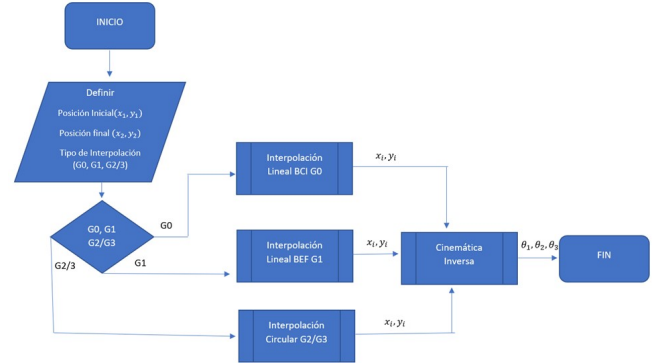


Figura 12: Diagrama de flujo de la planeación de trayectorias del brazo robot.

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.

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 y visualizador gratuito de programas CNC (código G) que permite validar gráficamente las trayectorias generadas a partir de instrucciones G-code. En esta simulación es posible corroborar que el código G ha sido ingresado correctamente para cumplir

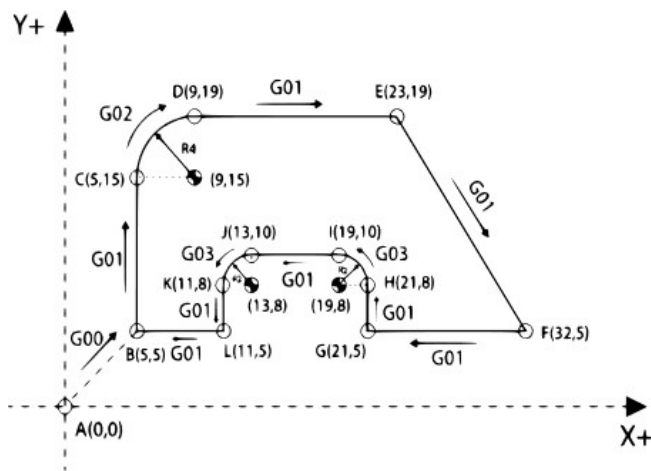


Figura 13: Trayectoria base utilizada para la generación del código G del robot SCARA de 2 grados de libertad.

Tabla 3: Posición inicial, final y código G de los puntos de la Figura 13

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
(32,5)	(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

con la trayectoria propuesta en la Figura 13. Cabe destacar que el uso del parámetro Z1 no corresponde a un movimiento real en el eje vertical del robot SCARA, sino que se emplea con un propósito representativo dentro del entorno de simulación. Este valor se utiliza para iniciar la trayectoria con una interpolación basada en coordenadas internas (BCI), lo cual simula el desplazamiento libre del efector hacia la posición inicial sin interacción con el plano de trabajo. Posteriormente, una vez alcanzado el punto de inicio, se activa la interpolación basada en el efector final (BEF), donde se definen los movimientos efectivos sobre el plano XY mediante los comandos G1, G2 y G3. Esta diferenciación entre la fase inicial (BCI) y la fase operativa (BEF) permite representar adecuadamente el ciclo completo de movimiento del robot dentro del software CNC, facilitando tanto la interpretación de los comandos G-code como la validación visual del algoritmo de planeación implementado.

En la figura 15 se muestra la simulación en Matlab de

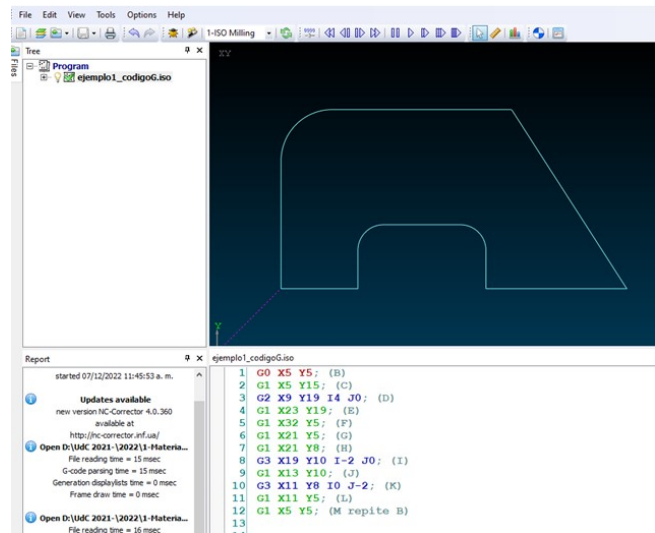


Figura 14: Simulación de código G en el software NC Corrector.

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 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).

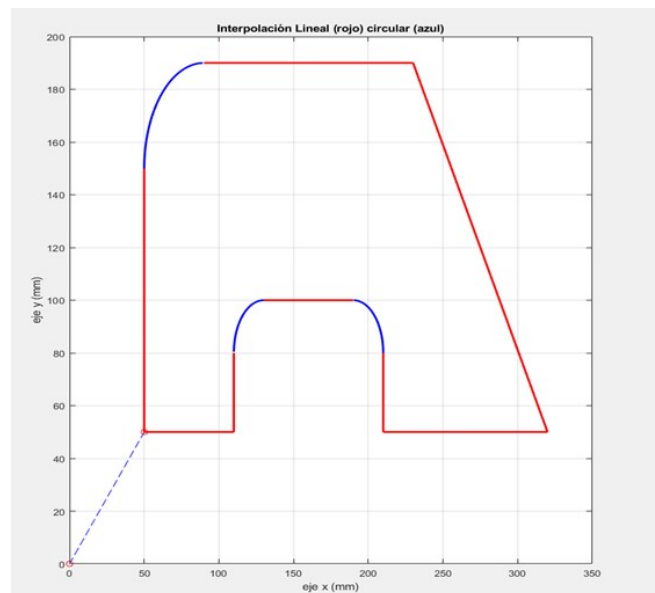


Figura 15: Simulación en Matlab de interpolación lineal y circular.

VI. 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 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.