

Seguimiento de trayectorias mediante cinemática diferencial aplicado a robots manipuladores

Trajectory tracking by differential kinematics applied to manipulator robots

L. A. Ramírez-López ^a, M Martínez-Aragón ^{b*}

^a Egresado de Ingeniería en Mecatrónica, Universidad Tecnológica de los Valles Centrales de Oaxaca, 71265, San Pablo Huixtepec, Oaxaca, México

^b División de carrera de Mecatrónica, Universidad Tecnológica de los Valles Centrales de Oaxaca, 71265, San Pablo Huixtepec, Oaxaca, México.

Resumen

El seguimiento de trayectorias, como aplicación al control de movimiento de un robot manipulador, puede desarrollarse con mecanismos matemáticos y computacionales. En este trabajo se presenta la implementación de una planificación de trayectorias mediante desplazamientos diferenciales (cinemática diferencial) formadas por segmentos de rectas para un robot manipulador KUKA LBR IIWA 14 R820, ocupando tres grados de libertad. En primera instancia se obtiene la cinemática directa usando el algoritmo de Denavit-Hartenberg, posteriormente se obtiene la matriz jacobiana para determinar los ángulos de las articulaciones en el que debe posicionarse el robot con base a pequeños desplazamientos. Finalmente se efectúan pruebas, realizando una co-simulación entre MATLAB y CoppeliaSim, se ingresa una trayectoria en MATLAB y el robot replica el trayecto en CoppeliaSim.

Palabras Clave: Brazo robótico, cinemática diferencial, planificación de trayectorias, seguimiento de trayectorias.

Abstract

Trajectory tracking as an application for motion control of a manipulator robot can be developed using mathematical and computational mechanisms. This paper presents the implementation of trajectory planning by differential displacements (differential kinematics) formed by segments of straight lines for a robotic manipulator KUKA LBR IIWA 14 R820 that takes three degrees of freedom. First, the forward kinematics are determined according to the Denavit-Hartenberg convention, then the Jacobi matrix is created to determine the angles of the joints where the robot should be positioned based on small displacements. Finally, tests are performed in which a co-simulation is performed between MATLAB and CoppeliaSim. Here, a trajectory is entered into MATLAB and the robot traces the trajectory in CoppeliaSim.

Keywords: Differential kinematics, robotic arm, trajectory planning, trajectory tracking.

1. Introducción

Derivado de la creciente evolución del ámbito industrial, las aplicaciones de la robótica se han extendido como una solución a los procesos que cotidianamente se desempeñan en el mundo moderno.

El rápido desarrollo tecnológico en conjunto con las crecientes necesidades humanas, han permitido materializaciones diversas de elementos mecánicos destinados a la realización de tareas específicas. Los robots manipuladores o brazos robóticos son autómatas de primordial importancia en este aspecto, debido a su versatilidad en múltiples contextos. Sus campos de acción incluyen: cirugías de alta precisión en áreas de la medicina, pintado y ensamblado dentro de la industria de manufactura de automóviles, manipulación de

procesos peligrosos, etc. Para cumplir con su objetivo, es necesario mover a través de trayectorias definidas a los eslabones que conforman la morfología del manipulador, de los cuales el efector final toma relevante importancia en el trayecto a describir debido a que es la parte encargada de realizar la tarea para la cual el mecanismo fue creado.

Existen métodos variados para planificar un trayecto a seguir. Desde el uso de caminos preprogramados, reconocimiento de patrones mediante visión artificial, algoritmos basados en interpolaciones (Porawagama, C. D., & Munasinghe, S. R., 2014), algoritmos que hacen uso del modelado del robot para la aproximación de posiciones subsecuentes iteradas de manera secuencial (Li, J., Gao, M., He, Z., & Yang, Y., 2018) o algoritmos que se basan en

*Autor para la correspondencia: mtz.marcelino@gmail.com

Correo electrónico: luisangelramlop3@gmail.com (Luis Angel Ramírez López), mtz.marcelino@gmail.com (Marcelino Martínez Aragón).

procesos de aproximación diferencial (Hossian, Carbajal, Alveal, & Bustamante), por mencionar algunos.

Una consideración importante es que las referencias se deben dar en tiempo real, teniendo en cuenta que cuando las posiciones y velocidades sean definidas en el espacio cartesiano, se deberá resolver la cinemática inversa para obtener los valores angulares (Braschi, E. L. A., 2015).

Con base a Braschi y compañía, en este trabajo se propone usar aproximaciones diferenciales para que el efector final siga una trayectoria, es decir, se calcula la cinemática inversa del punto inicial, en seguida se aplican incrementos infinitesimales (controlados mediante parámetros en el ambiente de programación) relativos a los ejes x e y . Se obtiene entonces el jacobiano a partir del punto de inicio, y los valores angulares correspondientes mediante el tratamiento vectorial de las ecuaciones que describen el posicionamiento actual del efector final para llegar a una nueva posición en tiempo real. Después se repite el proceso de forma iterativa a través del control de cambios de dirección que se logra aplicar con el algoritmo creado.

El manipulador debe de poder adaptarse a las diferentes formas de la trayectoria de entrada, motivo por el cual, el algoritmo de control es independiente de la forma de la trayectoria dada y puede adaptarse a la misma en tiempo de ejecución. Esta es otra forma de seguir trayectorias sin usar interpolaciones (Rodríguez Franco, García Rodríguez, Jara Ruiz, López Álvarez, & Orozco Soto), polinomios, elementos externos de reconocimiento (León, F. Morales, Altamirano, & R. Ruiz) o algún método citado anteriormente.

2. Robot Articulado de Tres Grados de Libertad

El modelo de manipulador KUKA LBR IIWA 14 R820 de CoppeliaSim, fue el modelo elegido para el desarrollo que se describe a continuación.

Para simplificar el proceso en estudio, y debido a la natural escalabilidad de los resultados obtenidos, se adaptó el diseño para accionar solamente tres grados de libertad, de sus seis variables articuladas originales.

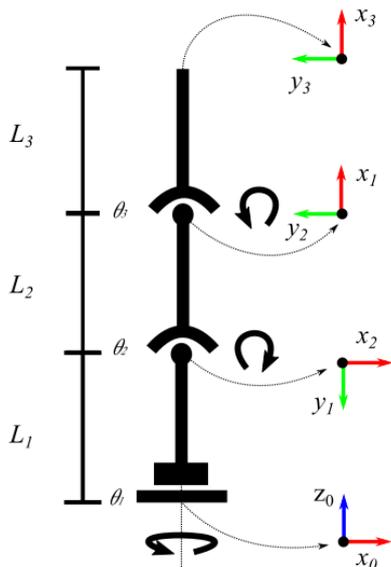


Figura 1: Asignación de tramas mediante el algoritmo Denavit Hartenberg.

2.1. Cinemática directa

Con base en el algoritmo Denavit Hartenberg para la asignación de tramas relativas a la estructura que conforma la morfología del manipulador (Barrientos, Peñín, Balaguer, & Aracil, 2017), se tiene la Figura 1, donde se muestran las tramas asignadas a los eslabones correspondientes, las direcciones positivas de movimiento angular, el sistema de referencia base y el sistema del extremo del robot.

Derivado de la Figura 1, y siguiendo el análisis de asignación de sistemas de referencia, se obtienen los parámetros de DH ilustrado en la Tabla 1.

J	θ (rad)	d (m)	a (m)	α (rad)
1	θ_1	L_1	0	$-\pi/2$
2	$\theta_2 - \pi/2$	0	L_2	π
3	θ_3	0	L_3	0

De la Tabla 1, se obtienen las siguientes matrices.

$$A_0^1 = \begin{pmatrix} C\theta_1 & 0 & -S\theta_1 & 0 \\ S\theta_1 & 0 & C\theta_1 & 0 \\ 0 & -1 & 0 & L_1 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$A_1^2 = \begin{pmatrix} S\theta_2 & -C\theta_2 & 0 & L_2 S\theta_2 \\ -C\theta_2 & -S\theta_2 & 0 & -L_2 C\theta_2 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$A_2^3 = \begin{pmatrix} C\theta_3 & -S\theta_3 & 0 & L_3 C\theta_3 \\ S\theta_3 & C\theta_3 & 0 & L_3 S\theta_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$T = \bar{T}(\theta_1, \theta_2, \theta_3) = A_0^1 A_1^2 A_2^3 = [\bar{n} \ \bar{o} \ \bar{a} \ \bar{p}] \quad (1)$$

Las matrices A_i^j generan las relaciones de movimiento de las tramas asignadas a eslabones consecutivos, mientras que la matriz T codifica la posición y orientación del efector final del manipulador con base al sistema fijo de referencia de su base, esto en función de las variables articulares.

La matriz T se conforma de cuatro vectores principales: tres de orientación, y uno de posición. Para el trabajo en cuestión, los vectores de orientación no serán relevantes. El vector de posición, en cambio, contiene la información de la posición del extremo del manipulador y será de primordial importancia en el desarrollo siguiente.

2.2. Cinemática Diferencial

A partir del modelado de la matriz T de la cinemática directa, sea P la función descrita en (2), contiene las coordenadas cartesianas del punto extremo de interés y a su vez proporciona la información posicional suficiente para el control diferencial del movimiento del efector final.

$$P = \bar{P}(\theta_1, \theta_2, \theta_3) = \bar{p} = (p_x, p_y, p_z) \quad (2)$$

Con el respaldo de la diferenciabilidad de las funciones componentes del vector \bar{p} , se obtiene la aproximación de

primer orden (Román, 1995) del incremento de la función P asociado al correspondiente incremento del vector de variables articulares $\vec{\theta} = (\theta_1, \theta_2, \theta_3)$.

$$P(\vec{\theta}_0 + \vec{h}) - P(\vec{\theta}_0) = [J(\vec{\theta}_0)] \vec{h} + \varepsilon(\vec{h}) \|\vec{h}\| \quad (3)$$

$$\lim_{\vec{h} \rightarrow 0} \varepsilon(\vec{h}) = 0$$

Considerando un error lo suficientemente pequeño, el cual se encuentre dentro de los parámetros deseados de acuerdo al contexto de aplicación, con base a (3), se obtiene:

$$[J(\vec{\theta}_0)]^{-1} [P(\vec{\theta}_0 + \vec{h}) - P(\vec{\theta}_0)] = \vec{h} \quad (4)$$

La ecuación vectorial (4) muestra el cambio \vec{h} necesario que se deberá aplicar a los valores actuales de las variables articulares $\vec{\theta}$ para llevar al efector final de la posición $P(\vec{\theta}_0)$ a la posición $P(\vec{\theta}_0 + \vec{h})$, a partir de la matriz jacobiana de la función P . De esta forma, se logra una aproximación aceptable del nuevo vector de coordenadas articulares $\vec{\theta}$ cuando el vector de posición asociado se desplaza una cantidad $P(\vec{\theta}_0 + \vec{h}) - P(\vec{\theta}_0)$.

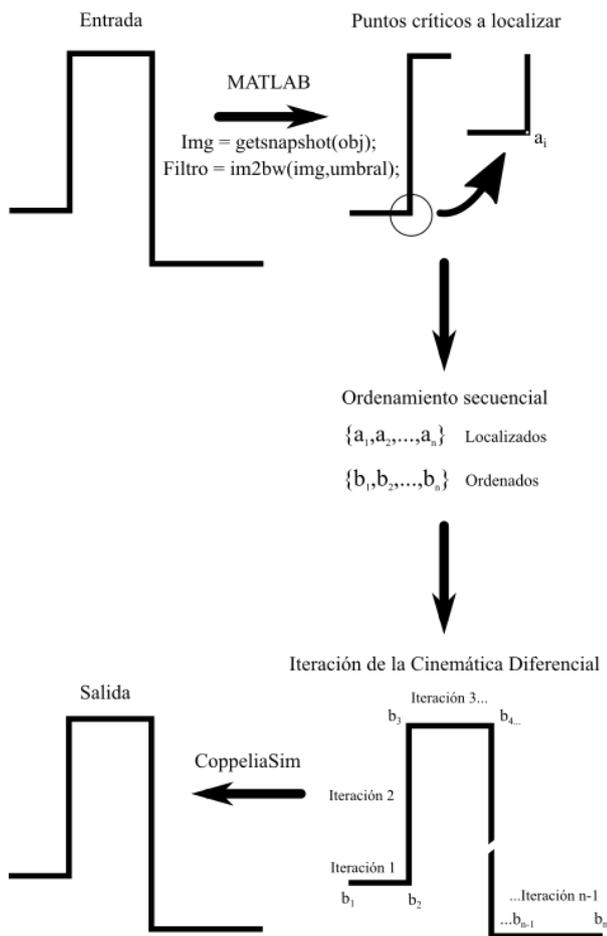


Figura 2: Esquema del proceso.

3. Desarrollo

Con el uso de MATLAB se desarrolló un algoritmo de búsqueda y planificación, donde se procesa la información necesaria para el reconocimiento de la trayectoria a seguir. El funcionamiento general se basa en una búsqueda de patrones en la imagen de entrada, de forma que puedan brindar las posiciones de los cambios de dirección críticos.

Una vez localizados los puntos en cuestión, se ordenan para formar una colección de elementos encadenados de manera secuencial, para que puedan ser adaptados dinámicamente al proceso de control mediante la cinemática diferencial. El ordenamiento es importante para asegurar la correcta ruta a seguir.

Posteriormente, la cinemática diferencial se itera para recorrer la trayectoria completa y generar un trazo continuo. La cinemática diferencial se aplica a cada trazo obtenido como resultado de la descomposición anterior. El trazo de salida se plasma de manera gráfica en el entorno de simulación de CoppeliaSim para validar el funcionamiento y resultados. La Figura 2 muestra un esquemático del proceso a seguir.

3.1. Trayectoria de entrada

La obtención de datos se inicia con la entrada de una trayectoria formada por la unión finita de segmentos de línea horizontales y verticales. Se debe tener el contraste necesario entre el fondo de la imagen y el trazo para facilitar el proceso de reconocimiento y filtrado.

La Figura 3 muestra la trayectoria de una primera prueba que cumple con las características requeridas.

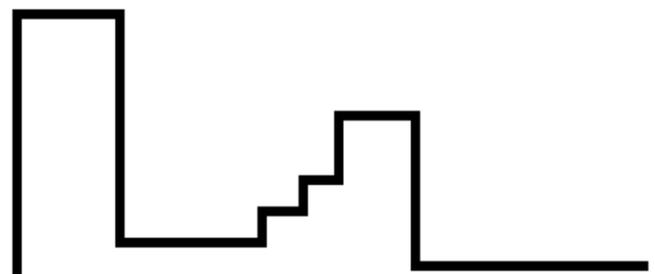


Figura 3: Trayectoria formada por la unión finita de segmentos de línea horizontales y verticales.

3.2. Procesamiento en MATLAB

Una de las formas de poder procesar la imagen de entrada y al mismo tiempo crear el objeto manipulable en MATLAB, es mediante la captura en imagen a través de las funciones de visión artificial que MATLAB proporciona. El objetivo se puede alcanzar mediante el proceso de creación de objetos en video y captura de *snapshots* para la manipulación de imágenes como objetos matriciales (MathWorks, MathWorks, s.f.).

Se usó el proceso de binarización (MathWorks, MathWorks, s.f.), para filtrar y reducir las impurezas de la imagen. La función binarizar de MATLAB proporciona como salida una matriz asociada a una imagen de entrada conformada por solamente dos tipos de valores ponderados. La Figura 4 muestra un esquema del tratamiento de la trayectoria de entrada.

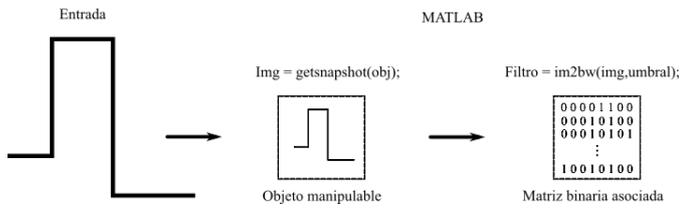


Figura 4: Tratamiento de la trayectoria de entrada.

El proceso posterior se basa en la descomposición de la imagen en segmentos como resultado de la búsqueda secuencial de patrones repetitivos, de manera que se puedan ubicar los puntos de cambio de dirección y clasificarlos de forma encadenada para tener la secuencia de trazos a seguir, y así poder recrear la trayectoria en totalidad. La búsqueda y clasificación se realizan manipulando directamente la matriz binaria creada.

La Figura 5 muestra los puntos críticos que el algoritmo de búsqueda es capaz de encontrar y clasificar.

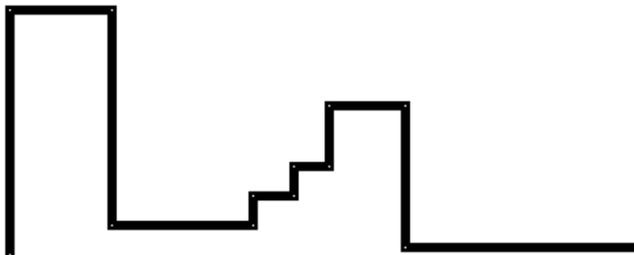


Figura 5: Trayectoria de entrada después del procesamiento y localización de puntos críticos para la planificación y seguimiento.

Finalmente, el seguimiento de la trayectoria requiere de un proceso iterativo de la cinemática diferencial, aplicado a cada segmento por separado. La aplicación del proceso base se ilustra en la Figura 6.

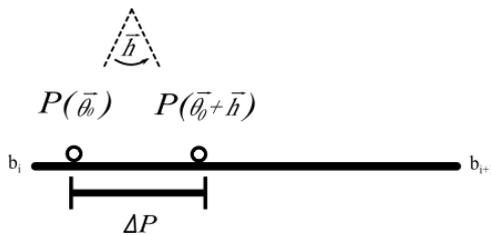


Figura 6: Funcionamiento base de la cinemática diferencial.

4. Co-simulación MATLAB/CoppeliaSim

Posterior al tratamiento de la trayectoria de entrada, se realizó el enlace con el software CoppeliaSim (Robotics, 2022) para la simulación en conjunto con MATLAB.

Tabla 2: Valores físicos para simulación

Parámetro	Valor (m)
L_1	0.36
L_2	0.42
L_3	0.51
$\Delta P = P(\vec{\theta}_0 + \vec{h}) - P(\vec{\theta}_0)$	0.00005

Utilizando tres de los seis grados de libertad del modelo original, los parámetros utilizados para simular el modelo KUKA LBR IIWA 14 R820 se muestran en la Tabla 2, al igual que el incremento diferencial; este último parámetro se puede ajustar para reducir el error mostrado en (3).

En la Figura 7 se muestra la salida generada posterior al proceso de simulación. Se puede observar una réplica de la trayectoria de entrada trazada por el robot.

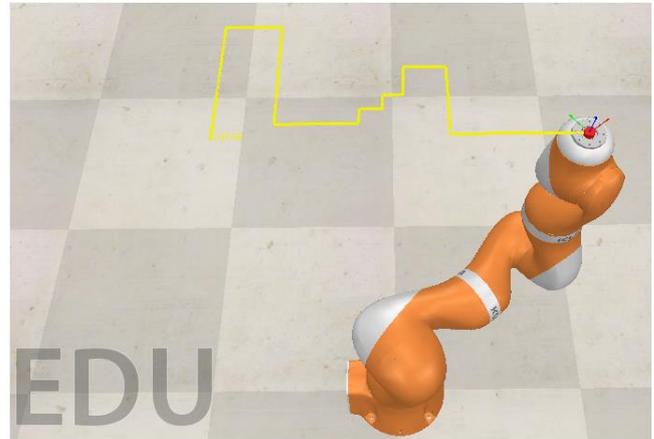


Figura 7: Trayectoria de salida después de la simulación en conjunto con CoppeliaSim.

5. Conclusiones

En este trabajo se obtiene la cinemática diferencial de tres grados de libertad del robot KUKA LBR IIWA 14 R820, para la aplicación en el seguimiento de trayectorias.

Utilizando la aproximación mediante cinemática diferencial, fue posible obtener una réplica de la trayectoria de entrada. Cabe señalar que no fue necesario el cálculo de la cinemática inversa del robot, proceso que en general requiere de mayor tiempo de procesamiento con el correspondiente aumento en la dificultad de los cálculos.

La escalabilidad a un modelo de más de tres grados de libertad se puede desarrollar con base en el mismo principio. De igual forma, el tratamiento de trayectorias que involucren segmentos o curvas diferentes a los descritos con anterioridad será posiblemente una extensión del proceso de codificación de la planificación y del algoritmo de control.

A diferencia de otros trabajos, se evitó la complejidad de métodos de seguimiento tales como los interpoladores, a costa del error derivado de la aproximación diferencial utilizada como base en el control de movimiento.

Respecto a la trayectoria de salida, el seguimiento a tamaño real de la trayectoria de entrada proporcionada, requerirán de escalados que generen las relaciones entre las magnitudes de entrada y salida correspondientes.

El proceso puede implicar un gran número de cálculos numéricos, y por lo tanto ser sensible a redondeos dependiendo del software de control que se utilice, y de la precisión.

Este trabajo muestra un enfoque simple y directo a la solución del problema de seguimiento de trayectorias, enfoque que puede adaptarse para aplicaciones de impresión (como placas de circuito PCB) y recreación de modelos digitales una vez que la escalabilidad del proceso permita el refinamiento en el tratamiento de curvas y segmentos mucho más generales a los mostrados. Queda como trabajo futuro la escalabilidad

anteriormente mencionada, tanto en la parte física, como en la de control y programación.

Referencias

- Barrientos, A., Peñín, L. F., Balaguer, C., & Aracil, R. (2017). *Fundamentos de robótica*. España: McGrawHill.
- Hossian, A., Carbajal, R., Alveal, M., & Bustamante, P. (s.f.). Una propuesta metodológica para la generación de trayectorias articulares de un manipulador industrial. Una visión desde los criterios de diseño. 7.
- León, A., F. Morales, E., Altamirano, L., & R. Ruiz, J. (s.f.). *Teaching a Robot New Tasks through Imitation and Feedback*. Obtenido de semanticscholar: <https://www.semanticscholar.org/>
- MathWorks. (s.f.). *MathWorks*. Recuperado el 30 de Abril de 2022, de <https://es.mathworks.com/help/imaq/getsnapshot.html>
- MathWorks. (s.f.). *MathWorks*. Recuperado el 30 de Abril de 2022, de <https://es.mathworks.com/help/images/ref/im2bw.html>
- Robotics, C. (27 de Enero de 2022). *Coppelia Robotics*. Recuperado el 30 de Abril de 2022, de <https://www.coppeliarobotics.com/helpFiles/index.html>
- Rodríguez Franco, M. E., García Rodríguez, J., Jara Ruiz, R., López Álvarez, Y., & Orozco Soto, S. (s.f.). Planeación de trayectorias robóticas mediante interpoladores cúbicos. 6.
- Román, J. d. (1995). En *Cálculo Infinitesimal de Varias Variables* (págs. 64-116). España: McGraw-Hill.