

Cinemática directa e inversa de brazo robótico SCORBOT -ER V+

Carlos Arguello – Valeria Fuenzalida - Felipe González – Karina Muñoz – Alejandro Sanhueza

Curso: Laboratorio de Robótica Industrial

Docente: René Torres

Carrera: Ingeniería Civil Mecatrónica

Fecha: 6 de octubre de 2023



Tabla de contenido

| | |
|--|-----------|
| RESUMEN..... | 3 |
| OBJETIVOS | 4 |
| OBJETIVO PRINCIPAL | 4 |
| OBJETIVOS ESPECÍFICOS | 4 |
| INTRODUCCIÓN..... | 5 |
| BRAZO ROBÓTICO SCORBOT ER VPLUS | 6 |
| MÉTODO DENAVIT-HARTENBERG | 8 |
| MÉTODO DE GRADIENTE DESCENDIENTE..... | 9 |
| MATERIALES UTILIZADOS | 11 |
| METODOLOGÍA | 14 |
| RESULTADOS..... | 17 |
| SIMULACIÓN DE LA CINEMÁTICA DIRECTA: | 17 |
| SIMULACIÓN DE LA CINEMÁTICA INVERSA | 20 |
| LIMITACIONES Y CONSIDERACIONES ADICIONALES:..... | 21 |
| CONCLUSIONES..... | 22 |
| REFERENCIAS BIBLIOGRÁFICAS..... | 23 |

Resumen

En este informe, se aborda la cinemática directa e inversa de un brazo robótico Scorbot ER Vplus utilizando Python y Matlab como lenguajes de programación. El brazo robótico, desarrollado por Intelitek, consta de cinco articulaciones de revolución y se utiliza en una variedad de aplicaciones, como ensamblaje y empaquetado. El objetivo principal del informe es comprender y modelar los movimientos de este robot a través de la cinemática directa e inversa.

La cinemática directa se calcula utilizando el método Denavit-Hartenberg, que implica asignar sistemas de coordenadas a cada articulación y calcular las matrices de transformación homogénea entre ellos. Por otro lado, la cinemática inversa se aborda mediante el método de gradiente descendente, que busca los ángulos de articulación necesarios para alcanzar una coordenada deseada en el espacio.

En resumen, este informe proporciona una base sólida para comprender y aplicar la cinemática en el contexto de un brazo robótico Scorbot ER Vplus, con énfasis en la simulación y el cálculo de los movimientos del robot, aunque se mencionan algunas limitaciones que deben considerarse en aplicaciones prácticas.



Objetivos

Objetivo principal

Obtener la cinemática Directa e Inversa del brazo robótico Scorbót ER Vplus, haciendo uso de los lenguajes de programación Python y Matlab

Objetivos específicos

- a) Cuantificar las dimensiones físicas del robot Scorbót ER Vplus
- b) Identificar la posición home default del robot Scorbót ER Vplus
- c) Determinar la cantidad de grados de libertad del robot Scorbót ER Vplus
- d) Modelar la cinemática directa e inversa del robot Scorbót ER Vplus mediante el uso del método Denavit-Hartenberg y método de la gradiente descendiente, respectivamente.



Introducción

En la era de la automatización y la robótica, la comprensión de la cinemática de los brazos robóticos es esencial para su diseño, control y aplicación en diversas industrias. Los robots manipuladores, en particular, desempeñan un papel crucial en tareas de ensamblaje, empaquetado, soldadura y muchas otras aplicaciones industriales.

Un robot es un dispositivo que integra sistemas mecánicos, eléctricos e informáticos para realizar determinada acción sin intervención humana, generalmente se compone de: sensores, controlador y actuador. Existe una amplia gama de robots que varían respecto a cuantas tareas realizan, el tipo de comunicación con el controlador y cómo interactúa con el usuario u objeto a manipular. Un robot manipulador está compuesto por una serie de eslabones conectados por articulaciones que le permiten moverse para sostener, levantar, rotar o desplazar objetos, este tipo de robots son utilizados para tareas de ensamblaje, empaquetado, pintura, entre otros.

Este informe se enfoca en la cinemática directa e inversa de un brazo robótico Scorbót ER Vplus. El Scorbót ER Vplus, desarrollado por Intelitek, es un robot manipulador con cinco articulaciones de revolución que proporciona una amplia gama de movimientos. El objetivo principal de este informe es abordar cómo calcular y comprender los movimientos de este brazo robótico utilizando los lenguajes de programación Python y Matlab.



Brazo robótico SCORBOT ER Vplus

El robot Scorbót ER Vplus (Figura 1) desarrollado por la compañía taiwanesa Intelitek es un robot manipulador compuesto por un brazo robot y un controlador-A ACL. El brazo robot, cuando no está sobre la base deslizante, o riel, posee 5 articulaciones o juntas de revolución sin contar la articulación de la pinza que permite su apertura y cerradura. El rango de movimiento del eje de la base son $0^{\circ}/250^{\circ}$, rotación del hombro $130^{\circ}/-35^{\circ}$, rotación del codo $130^{\circ}/-130^{\circ}$, inclinación de la muñeca $130^{\circ}/-130^{\circ}$, giro de la muñeca $570^{\circ}/-570^{\circ}$ [1]. A través del estudio realizado se considera únicamente aquellas 5 articulaciones por lo que se consideran 5 grados de libertad (G.D.L.)

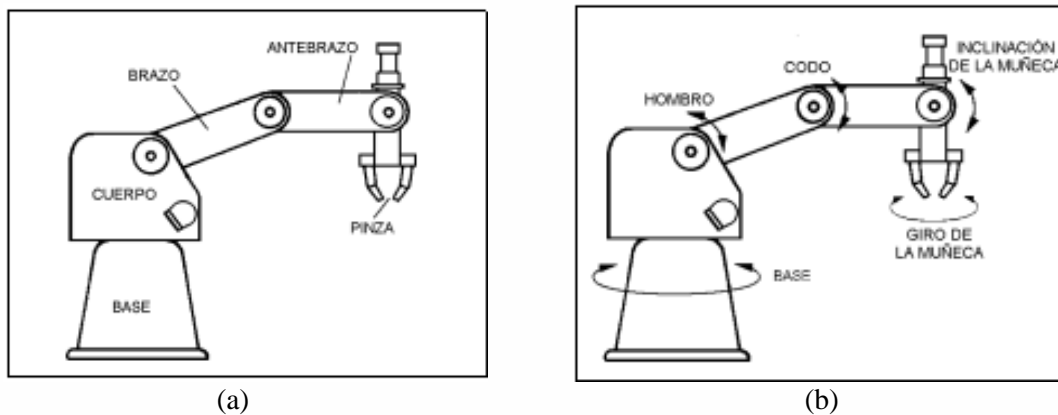


Figura 1: (a) Eslabones de brazo robótico [1] (b) Articulaciones del robot [1]

La posición de inicio o *home* posiciona a la articulación base en 0° , el hombro en -120° , el codo en $\sim 95^{\circ}$, la inclinación de muñeca en $\sim 88^{\circ}$ y el giro de la muñeca en 0° ; el punto centro del efector (T.C.P) se posiciona en las coordenadas (169,0,503) en escala de milímetros [4] respecto al sistema de coordenadas ubicado en el centro de la cara inferior de la base (Figura 2).

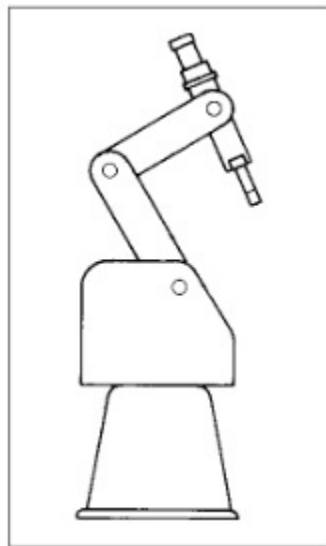


Figura 2: Posición de inicio SCORBOT ER Vplus [4]

Las dimensiones entre los eslabones (Figura 3) determinan el rango de operación del dispositivo, de manera que el máximo rango de operación desde la vista superior es de 610mm mientras que desde la vista lateral es de 1040mm [1].

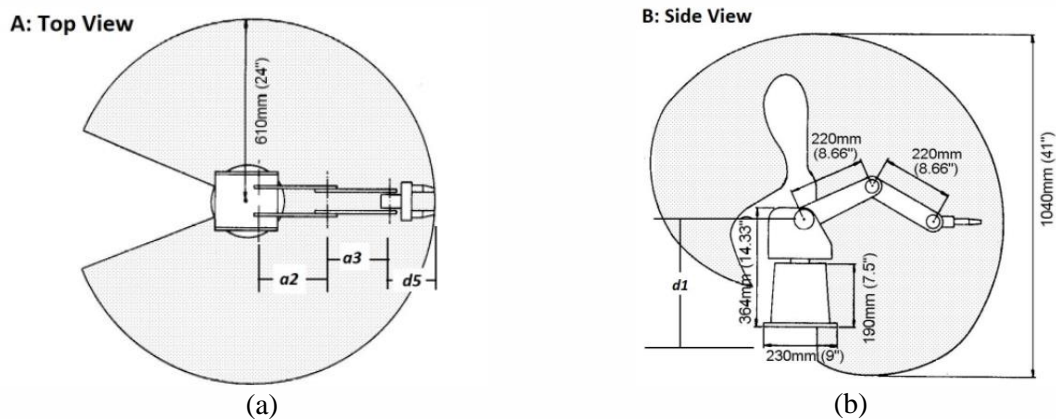


Figura 3: (a) Rango de operación (a) Vista superior (b) Vista lateral [1]

Se estudió cómo desplazar la pinza, o efector, a una coordenada específica del rango de operación utilizando teorías de cinemática, las cuales relacionan la posición del efector en el tiempo según los ángulos de rotación de cada articulación (o ángulos de rotación de cada eje con respecto a la articulación anterior). El análisis de problemas cinemático posee dos categorías dependiendo de las



variables que se busca obtener: La cinemática directa calcula la posición del efector a partir de los ángulos de rotación en cada articulación con respecto a un sistema de coordenadas de referencia, mientras que la cinemática inversa consiste en obtener los ángulos de rotación para cada articulación a partir de la posición del efector en el espacio. La resolución de ambos acercamientos requiere de metodologías distintas ya que, pese a que relacionan los mismos parámetros, la cinemática directa es de solución única mientras que la cinemática inversa posee múltiples soluciones, los cuales son vectores de ángulos de rotación que se han de comparar según cuántos movimientos realizan por articulación o cómo distribuyen las cargas internas. Por lo anterior, el cálculo de cinemática inversa requiere el uso de métodos de aproximación numérica el cual es realizado mediante un código automatizado en lugar de realizarlos manualmente, con el propósito de mejorar la precisión, la eficiencia y reducir posibles errores humanos.

Para la programación y planificación de trayectoria del efector se estudió la cinemática directa con el método Denavit-Hartenberg y la cinemática inversa con el método numérico de aproximación gradiente descendiente para determinar vectores de rotación de las articulaciones a partir de las coordenadas del efector (posición de la pinza en este caso).

Método Denavit-Hartenberg

Para aplicar el método, es necesario asignar sistemas de coordenadas por cada articulación del robot para después calcular las matrices de transformación homogénea entre sistemas de coordenadas consecutivos. Los datos necesarios para aplicar el método son las variables de ángulo, longitud y desplazamiento de cada articulación del robot.

La asignación de sistema de coordenadas comienza en la articulación de la base y termina en la articulación de la muñeca (articulación que sostiene a la pinza), en cada asignación de sistemas de coordenadas se siguen las siguientes condiciones:

- El eje Z debe estar alineado con el eje de rotación de la articulación.
- El eje X debe estar en la dirección común entre el eje Z de la articulación anterior y el eje Z de la articulación actual.
- El eje Y debe ser ortogonal tanto al eje X como al eje Z.

Una vez terminado de asignar todos los sistemas de coordenadas, se completa la matriz de transformación de cada eslabón (Ecuación 1).



$${}^{i-1}A_i = \begin{pmatrix} \cos \theta_i & -\cos \alpha_i \sin \theta_i & \sin \alpha_i \sin \theta_i & a_i \cos \theta_i \\ \sin \theta_i & \sin \alpha_i \cos \theta_i & -\sin \alpha_i \cos \theta_i & a_i \sin \theta_i \\ 0 & \sin \alpha_i & \cos \alpha_i & d_i \\ 0 & 0 & 0 & 0 \end{pmatrix} \quad (1)$$

Donde θ_i es el ángulo de rotación alrededor del eje Z desde el eje X del eslabón anterior hasta el eje X del eslabón actual.

Donde d_i es la distancia entre el origen del sistema de coordenadas del eslabón anterior hasta la intersección del eje Z del eslabón anterior con el eje Z del eslabón actual.

Donde α_i es la distancia de la normal común del eje Z del eslabón anterior con el eje Z del eslabón actual.

Donde α_i es el ángulo de rotación alrededor del eje X desde el eje Z del eslabón anterior hasta el eje Z del eslabón actual.

Una vez completadas las matrices de transformación de cada eslabón, estas se multiplican entre sí desde la base hasta el eslabón efector, al resultado se le conoce como matriz de transformación total (Ecuación 2).

$$T = {}^0A_1 {}^1A_2 \dots {}^{n-1}A_n \quad (2)$$

Método de Gradiente Descendiente

Para aplicar el método de optimización es necesario conocer una función que relacione la posición del efector con los ángulos de rotación de las articulaciones, además es necesario una estimación inicial de los ángulos de rotación de las articulaciones para alcanzar la posición del efector.

Se deriva la función de posición del efector respecto a sus variables, de forma que se obtiene una matriz jacobiana de 3 filas y 4 columnas (ya que la función de posición tiene componentes para el eje X,Y,Z mientras que solo hay 4 articulaciones del brazo robot que afectan a la posición del T.C.P), al reemplazar en ella la estimación inicial de los ángulos de rotación de las articulaciones se obtendrá el valor de la pendiente de la función de posición, utilizando el valor de la pendiente se varía la siguiente estimación inicial. Primero se calcula el error entre las coordenadas que se obtienen con la estimación inicial respecto a las coordenadas deseadas, después la matriz jacobiana evaluada con la estimación inicial es traspuesta para multiplicarse con el vector de error, el resultado es equivalente a la diferencia del vector de ángulos deseado y la estimación inicial de ángulos. Por último, el valor de la siguiente estimación inicial de ángulos es variado según una tasa de aprendizaje arbitraria (Ecuación 3). El

algoritmo es iterado hasta que se obtenga una solución que indique coordenadas dentro del margen de error seleccionado.

$$\overrightarrow{q_{n+i}} = \overrightarrow{q_i} + \alpha \cdot [\nabla f]^t \vec{e} \quad (3)$$

Donde $\overrightarrow{q_i}$ es la estimación inicial del vector de ángulos de rotación de las articulaciones.

Donde α es la tasa de aprendizaje.



Donde $[\nabla f]^t$ es la matriz traspuesta aplicada a la matriz jacobiana.

Donde \vec{e} es el vector de error entre las coordenadas deseadas y las resultantes a la estimación inicial.

El algoritmo varía la cantidad de iteraciones realizadas según la estimación inicial, la tasa de aprendizaje y la tolerancia de error seleccionado. Aquello es motivo de que se establezca un máximo de iteraciones como condición de finalización del algoritmo para evitar sobre exigir las herramientas de cálculo.



Materiales utilizados

| Hardware y herramientas | |
|---|--|
|  <p><i>Figura 4: Cinta métrica</i></p> | <p>Rango de medición: 0-205cm</p> <p>División: 1mm</p> <p>Peso neto: 50grs</p> <p>Ancho de cinta: 19mm</p> <p>Tipo de gancho: sin gancho</p> |
|  <p><i>Figura 5: Scorbot ER-Vplus</i></p> | <p>Ejes: 5 + pinza</p> <p>Grados de libertad</p> <p>Radio de operación 610mm</p> <p>Eje 1 base 310°</p> <p>Eje 2 Brazo inferior +130°/-30°</p> <p>Eje 3 Brazo superior 35°</p> <p>Eje 4 Elevación de pinza +/-130°</p> <p>Eje 5 Giro de pinza +/- 130°</p> <p>Velocidad de giro: 600mm/seg</p> <p>Capacidad de carga: 1 Kg en brazo extendido</p> <p>Servomotores: 6 servomotores DC con control de lazo cerrado</p> <p>Peso: 10.8Kg</p> |
| Software Utilizado | |



| | |
|---|---|
|  <p><i>Figura 6: Matlab</i></p> | <p>Matlab abreviado de Matrix Laboratory es un sistema de cómputo numérico que ofrece un IDE capaz de procesar el lenguaje de programación M, el cual está diseñado específicamente para trabajos de álgebra lineal y cálculos numéricos, así como representaciones espaciales. A través de los años se han implementado diversos framework útiles para modelación de sistemas mecatrónicos.</p> |
|  <p><i>Figura 7: Python</i></p> | <p>Python es un lenguaje de programación interpretado, cuyos objetivo y visión son crear un lenguaje de programación rápido útil para el prototipado capaz de incluso generar proyectos completos mediante el uso de Pyp (Python package index) se pueden implementar diversos módulos capaces de añadir nuevas funciones y métodos útiles para desarrollo web, bases de datos, interfaz de usuarios, cálculo numérico, desarrollo de software y videojuegos.</p> |
| <p>Librerías utilizadas</p> | |





Figura 8: Numpy

NumPy es la librería fundamental para computación científica. Es una librería en Python capaz de definir y operar con arreglos multidimensionales, además de generar operaciones rápidas sobre arreglos. Fourier transform, álgebra lineal y operaciones estadísticas.

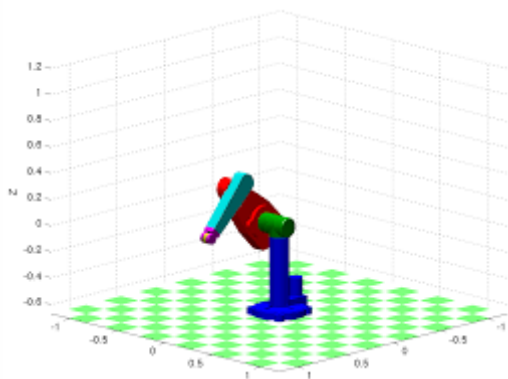


Figura 9: Robotics toolbox library

Robotics system toolbox es un grupo de herramientas y algoritmos especializados que pueden ser utilizados para cálculos de cinemática directa e inversa basándose en la cinemática del cuerpo rígido. La librería es capaz de generar conexiones directas con gazebo, así como ambientes virtuales con la finalidad de prototipar hardware robótico.



Metodología

Los cálculos de cinemática directa e inversa y también la simulación de brazo robótico Scorbobot ER V plus se realiza en el software MATLAB, para lo cual se hace uso de herramientas de modelación de robots de Robotics System Toolbox, de Peter Corke, la cual se instala de la página oficial del autor [3].

En primer lugar, se realiza el análisis de Denavit Hartenberg para el brazo robótico, siguiendo los pasos del procedimiento descrito por este método, de tal manera obtener la Figura 10, la cual contiene los sistemas de referencia de cada articulación y con lo cual se obtienen los parámetros de Denavit Hartenberg. Luego, con los parámetros calculados se realiza un Live Script en MATLAB, llamado “Scorbobot_DH.mlx”, el cual calcula las matrices de cada articulación, además de la matriz correspondiente al efector o garra del brazo respecto a su base.

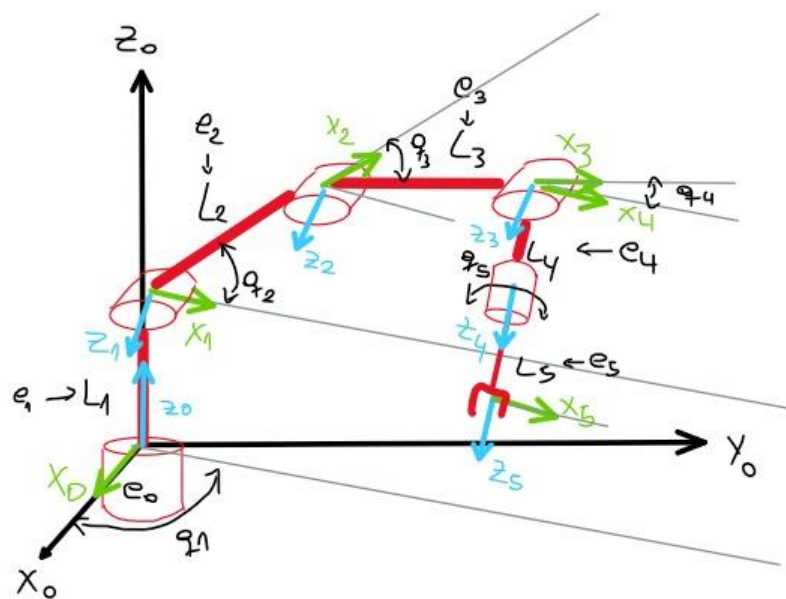


Figura 10: Análisis de Denavit Hartenberg

A continuación, para simular la cinemática directa del brazo se utiliza las herramientas de Robotic System Toolbox, las cuales por medio de los parámetros e indicando el tipo de articulación, ya sea de revolución o prismática, además de definir el espacio virtual, crea el robot en un entorno 3D el cual también cuenta con una Interfaz Gráfica de Usuario (GUI) la cual permite modificar el ángulo de cada articulación. Es por esto por lo que para la simulación de la cinemática del brazo robótico se debe ejecutar el archivo “Scorbobot_cinemática_directa.m” el cual abre el simulador con el brazo en la posición home, y el usuario tiene la opción de modificar el ángulo de cada articulación mediante deslizadores gráficos de la GUI.



Por otro lado, para la cinemática inversa del brazo robótico se realiza mediante el método de aproximación numérica de gradiente descendiente, donde el código se basa en el código proporcionado por el profesor para un brazo robótico de 2gdl. Para este código se usan las coordenadas de la posición del efector las cuales se obtienen de la matriz DH del brazo robótico que se encuentra en el archivo “Scorbot_DH.mlx” y corresponden a las descritas a continuación:

$$X = (\sin(q(4)) * (\sin(q(3)) * (\sin(q(1)) - \cos(q(1)) * \sin(q(2))) + \cos(q(1)) * \cos(q(2) * \cos(q(3))) - \cos(q(4)) * (\sin(q(1)) + \cos(q(3)) * (\sin(q(1)) - \cos(q(1)) * \sin(q(2))) - \cos(q(1)) * \cos(q(2)) * \sin(q(3)))) * (L4 + L5) + L2 * \cos(q(1)) * \cos(q(2)) + L3 * \sin(q(3)) * (\sin(q(1)) - \cos(q(1)) * \sin(q(2))) + L3 * \cos(q(1)) * \cos(q(2)) * \cos(q(3)));$$

$$Y = L2 * \cos(q(2)) * \sin(q(1)) - (\sin(q(4)) * (\sin(q(3)) * (\cos(q(1)) + \sin(q(1)) * \sin(q(2))) - \cos(q(2)) * \cos(q(3)) * \sin(q(1))) - \cos(q(4)) * (\cos(q(1)) + \cos(q(3)) * (\cos(q(1)) + \sin(q(1)) * \sin(q(2))) + \cos(q(2)) * \sin(q(1)) * \sin(q(3)))) * (L4 + L5) - L3 * \sin(q(3)) * (\cos(q(1)) + \sin(q(1)) * \sin(q(2))) + L3 * \cos(q(2)) * \cos(q(3)) * \sin(q(1)));$$

$$Z = L1 - (\cos(q(4)) * (\cos(q(2)) * \cos(q(3)) - \sin(q(2)) * \sin(q(3))) - \sin(q(4)) * (\cos(q(2)) * \sin(q(3)) + \cos(q(3)) * \sin(q(2)))) * (L4 + L5) + L2 * \sin(q(2)) + L3 * \cos(q(2)) * \sin(q(3)) + L3 * \cos(q(3)) * \sin(q(2)));$$

Siguiendo con el método de gradiente descendiente, se tiene que con esta coordenada se procede a calcular las derivadas parciales de cada componente respecto a los distintos ángulos obteniendo de esta forma el Jacobiano, mientras que el resto de código se mantiene invariante, dejando el máximo de iteraciones en 1000, la tolerancia en $1e-3$ y el tamaño del paso para el descenso del gradiente en 3. Luego, mediante el simulador creado para la cinemática directa se evalúan los ángulos calculados, y se grafica el brazo robótico, además de un punto amarillo el cual representa la coordenada objetivo o deseada.

Para usar este programa se tiene que ingresar la coordenada objetivo, la cual se ingresa en el arreglo “xd” y, además, se tiene el arreglo “q” el cual contiene los ángulos de la posición inicial con la cual se desea calcular la posición final.

Cabe destacar que el tamaño del brazo robótico se escala para realizar la simulación en un tamaño acorde, de tal manera que las longitudes de los eslabones con respecto a los de la simulación están en 100:1, lo cual corresponden a las siguientes dimensiones de la Tabla 1:

Tabla 1: Escalamiento de dimensiones del robot

| Eslabón | Largo real [mm] | Largo virtual [mm] |
|---------|-----------------|--------------------|
| 1 | 364 | 3.64 |
| 2 | 220 | 2.2 |
| 3 | 220 | 2.2 |
| 4 | 70 | 0.7 |
| 5 | 70 | 0.7 |



Resultados

Análisis parámetros Denavit-Hartenberg:

Para llevar a cabo el análisis de cinemática directa en el brazo robótico Scorbobot ER V plus, se procede a obtener los parámetros de Denavit-Hartenberg; cuyos valores se presentan en la tabla 2. Este paso es fundamental en el proceso de modelado y análisis del robot, ya que permite establecer un sistema de referencia para cada articulación y definir las relaciones geométricas entre ellas. La obtención de estos parámetros es esencial para la posterior formulación de las matrices de transformación homogénea. Estos parámetros proporcionan una representación matemática precisa de la estructura y geometría del robot, lo que facilita la planificación y control de sus movimientos en el espacio tridimensional; cabe notar que estos resultados se obtienen a partir del análisis que ilustra la figura 10 previamente presentada.

Tabla 2: Resultados parámetros Denavit-Hartenberg

| | θ_x | d_x | a_z | α_z |
|---------|-----------------|-------------|-------|------------|
| H_1^0 | $q_1 *$ | L_1 | | $r/2$ |
| H_2^1 | $q_2 *$ | 0 | L_2 | 0 |
| H_3^2 | $q_3 *$ | 0 | L_3 | 0 |
| H_4^3 | $(q_4 + r/2) *$ | 0 | 0 | $r/2$ |
| H_5^4 | $q_5 *$ | $L_4 + L_5$ | 0 | 0 |

Simulación de la Cinemática Directa:

En la simulación de la cinemática directa del brazo robótico Scorbobot ER V plus, se logró obtener una representación visual del brazo robótico en un entorno 3D utilizando las herramientas de Robotics System Toolbox, de Peter Corke. Esto permitió observar cómo el brazo robótico se mueve en función de los ángulos de sus articulaciones. Se presentan en la Figura 11, la simulación obtenida en su posición home y en la Figura 12, una variación en los parámetros q_n ; apreciando así el movimiento del brazo robótico en función de los ángulos seleccionados.



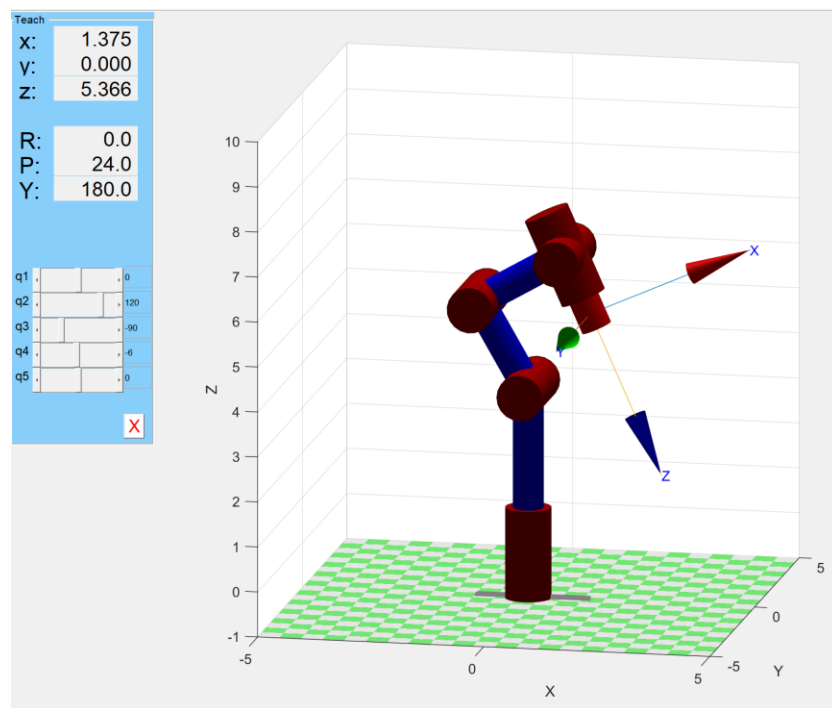


Figura 11: Simulación de la cinemática directa del brazo robótico Scorbot ER V plus.

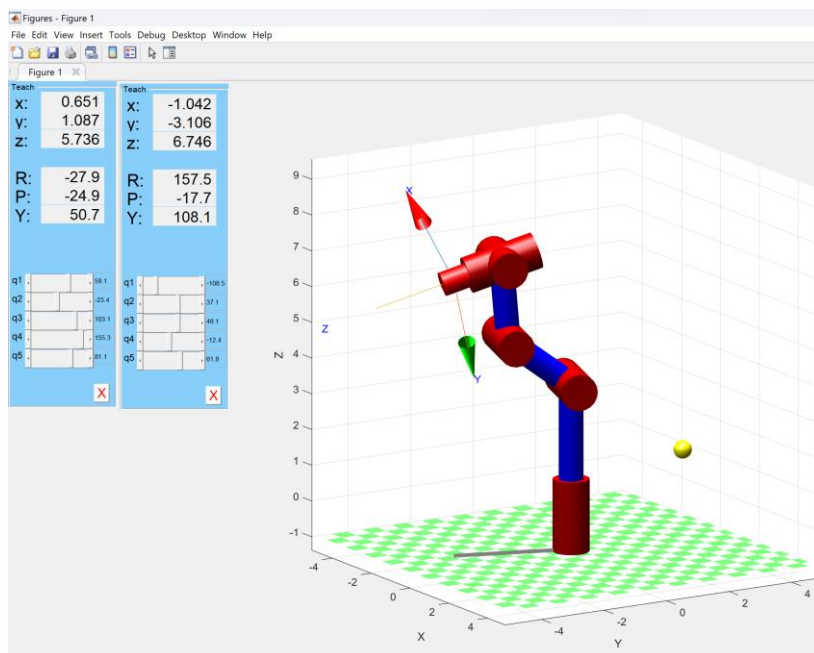


Figura 12: Variación en los parámetros en la simulación de la cinemática directa del brazo robótico Scorbot ER V plus.



Se encontraron varios aspectos notables en la simulación de la cinemática directa; la herramienta proporciona una visualización 3D que logra una representación gráfica precisa y realista del brazo robótico. Esto es útil para comprender cómo se mueve y se posiciona en el espacio tridimensional.

La inclusión de una Interfaz Gráfica de Usuario (GUI) permite al usuario modificar los ángulos de cada articulación mediante deslizadores gráficos. Esto facilita la exploración de diferentes configuraciones y su efecto en la posición final del efector.

Sin embargo, se observó una limitación importante en la simulación: la falta de restricciones en los ángulos de las articulaciones. Esto condujo a situaciones en las que los eslabones del brazo se superponían, como en la Figura 13 se aprecia, lo que en la realidad sería una configuración inviable. Sería beneficioso implementar límites en los ángulos de las articulaciones para evitar estas situaciones.

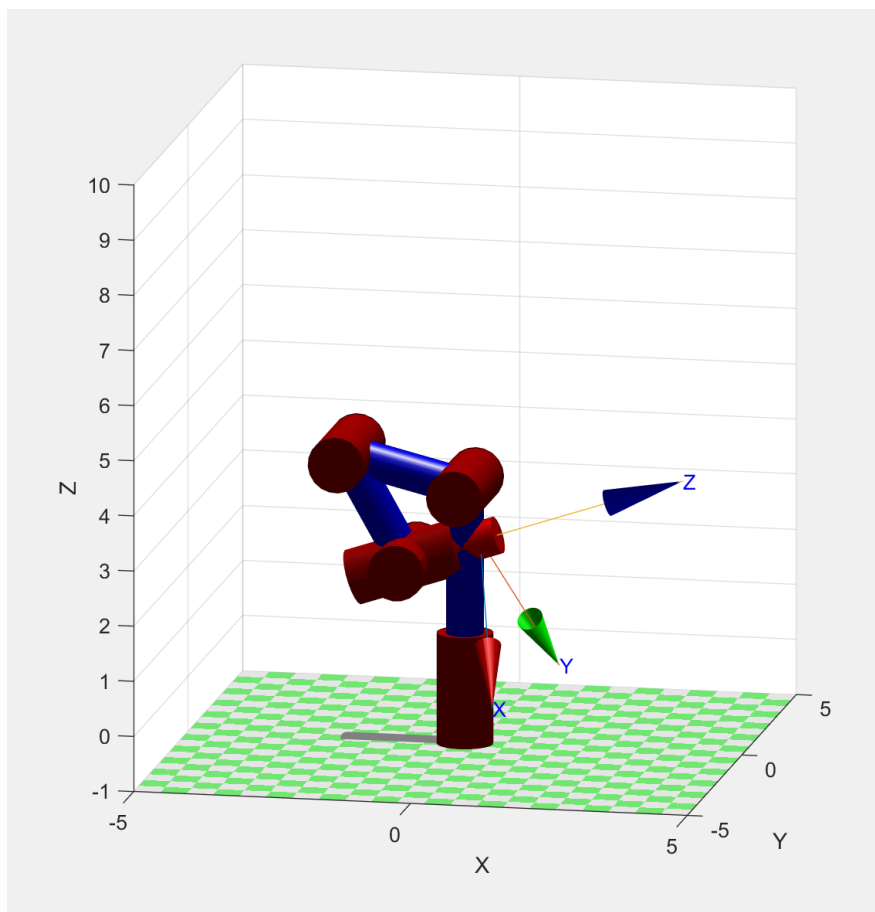


Figura 13: Eslabones superpuesto en simulación del brazo robótico Scorbot ER V plus

Simulación de la Cinemática Inversa

Para la cinemática inversa, se implementó el método de aproximación numérica de gradiente descendente para calcular los ángulos de articulación necesarios para posicionar el efector del brazo robótico en una coordenada deseada. El proceso incluyó calcular el Jacobiano y realizar iteraciones para encontrar los ángulos adecuados. Se presentan en la 14, los ángulos de rotación, para alcanzar la posición objetivo (2.2,2.2,2.0) en coordenadas cartesianas; también el ítem comprobación donde se aprecia la posición que tomaría el robot si se rotara en los ángulos obtenidos.

```
Command Window
Ángulos:
-4.7389
-2.1693
 2.4062
-11.4769

Comprobación (f(q)) :
 2.1999
 2.1997
 2.0002
```

Figura 144: Ángulos de rotación, para alcanzar la posición objetivo (2.2,2.2,2.0) en coordenadas cartesianas del brazo robótico Scorbot ER V plus.

Se observaron los siguientes aspectos en la cinemática inversa; con respecto a la Precisión de la Cinemática Inversa; el método de gradiente descendente demostró ser eficaz para encontrar los ángulos de articulación requeridos para alcanzar la coordenada objetivo. Esto es crucial en aplicaciones de robótica, donde se debe lograr una posición específica con precisión.

Se configuró un límite máximo de iteraciones (1000) y una tolerancia ($1e-3$) para controlar el proceso de búsqueda. Estas configuraciones permitieron encontrar soluciones dentro de un tiempo razonable y con precisión.

Se mencionó que las dimensiones del brazo robótico se escalaron (en una relación de 100:1) para la simulación. Esto es importante tenerlo en cuenta, ya que los resultados obtenidos se aplicarán a la escala real del brazo robótico.



Limitaciones y consideraciones adicionales:

1. Límites Físicos: Aunque la cinemática inversa calcula los ángulos necesarios para llegar a una coordenada objetivo, no tiene en cuenta posibles limitaciones físicas del brazo robótico, como obstáculos en el entorno.
2. Singularidades: En algunos casos, el brazo robótico puede encontrarse en configuraciones singulares donde no se puede encontrar una solución única para la cinemática inversa. Estas singularidades deben ser manejadas cuidadosamente en aplicaciones prácticas.
3. Eficiencia del Método: La eficiencia del método de gradiente descendente puede depender de la elección de la posición inicial y la configuración del brazo robótico. Se deben explorar diferentes estrategias para mejorar la convergencia en casos difíciles.

En resumen, la simulación de la cinemática directa e inversa del brazo robótico Scorbot ER V plus utilizando MATLAB y Robotics System Toolbox de Peter Corke proporciona una herramienta valiosa para comprender y planificar los movimientos del robot.

Sin embargo, es importante resaltar que, en la realidad, un brazo robótico posee limitaciones físicas en su espacio de trabajo. Tal como se ha observado durante la experimentación, al intentar alcanzar una coordenada objetivo que se encuentra más allá de las capacidades físicas del robot, los cálculos de la cinemática inversa pueden generar resultados imprecisos o, en algunos casos, inválidos.

Este fenómeno pone de manifiesto la crítica importancia de definir y restringir de manera adecuada el espacio de trabajo factible del robot. Esto conlleva la imposición de límites en los valores posibles de sus ángulos de articulación y posiciones finales. Establecer estas restricciones garantiza que el robot opere con seguridad y eficacia dentro de un rango limitado de movimientos, previniendo configuraciones inviables o peligrosas. La limitación del espacio de trabajo es una práctica esencial en la planificación y programación de robots, y se convierte en un factor determinante para asegurar un funcionamiento óptimo en aplicaciones del mundo real.



Conclusiones

La comprensión de la cinemática directa e inversa es esencial para el diseño y la operación efectiva de robots manipuladores como el Scorbobot ER Vplus. Este informe ha abordado estos conceptos fundamentales y ha proporcionado una visión general de cómo calcular y aplicar la cinemática en este contexto.

En la simulación de la cinemática directa, se logró una representación visual precisa del brazo robótico en un entorno 3D, lo que permitió comprender cómo las articulaciones afectan la posición y orientación del extremo del brazo. La inclusión de una Interfaz Gráfica de Usuario (GUI) facilitó la interacción y exploración de diferentes configuraciones de articulaciones. Sin embargo, se destacó la necesidad de implementar restricciones en los ángulos de las articulaciones para evitar configuraciones inviables.

La cinemática inversa se abordó mediante el método de gradiente descendente, que demostró ser eficaz para encontrar los ángulos de articulación requeridos para alcanzar una coordenada objetivo en el espacio. Se configuraron límites de iteraciones y tolerancia para controlar el proceso de búsqueda, lo que resultó en soluciones precisas y eficientes. Además, se mencionó la importancia de escalar las dimensiones del brazo robótico para la simulación, lo que facilitó la visualización de los movimientos.

Es importante destacar que, en aplicaciones prácticas, se deben considerar limitaciones adicionales, como obstáculos en el entorno y configuraciones singulares que pueden complicar la cinemática inversa. La programación y control de un robot manipulador en situaciones del mundo real a menudo involucran algoritmos más avanzados y sensores para abordar estos desafíos.

En última instancia, este informe proporciona una base sólida para comprender y aplicar la cinemática en el contexto de un brazo robótico Scorbobot ER Vplus. La simulación y el cálculo de movimientos son componentes esenciales para la programación y el control de robots manipuladores en aplicaciones industriales y más allá.



Referencias bibliográficas

- [1] Intelitek Inc. (1996). *Scorbot-ER V plus: User's Manual*. Recuperado de: <https://downloads.intelitek.com/Manuals/Robotics/Discontinued%20Machines/>
- [2] Ciencia_explicada. (2013). *Parametrización Denavit-Hartenberg para robots: teoría video y nueva aplicación libre*. Recuperado de: <https://www.ciencia-explicada.com/2013/02/parametrizacion-denavit-hartenberg-para.html>
- [3] Corke P. (s.f.) *Robotics Toolbox*. Recuperado de: <https://petercorke.com/toolboxes/robotics-toolbox/>
- [4] Intelitek Inc. (2016). *Scorbase: User's Manual*. Recuperado de: https://downloads.intelitek.com/Manuals/Robotics/ER-4u/Scorbase_USB_I.pdf
- [5] Castillo M. (2014). *Informe descriptivo Scorbot -ER Vplus*. Recuperado de: <https://studylib.es/doc/540097/informe-descriptivo-scorbot-er-v-plus---iaci>
- [6] MathWorks (s.f.). *Matlab*. Recuperado de: <https://la.mathworks.com/products/matlab.html>
- [7] Numpy (s.f.). *Numpy documentation*. Recuperado de : <https://numpy.org/doc/stable/>
- [8] Python (s.f.). *Python*. Recuperado de: <https://www.python.org/about/>

