



**UNIVERSIDAD ESTATAL DE MILAGRO**  
**VICERRECTORADO DE INVESTIGACIÓN Y POSGRADO**

**MAESTRÍA EN MATEMÁTICA**  
**CON MENCIÓN EN MODELACIÓN MATEMÁTICA**

**Tema:**

MODELADO Y SIMULACIÓN DE LA CINEMÁTICA DE UN ROBOT  
MANIPULADOR DE CINCO GRADOS DE LIBERTAD MEDIANTE MATLAB.

**Autor:**

**MSc. Alberto Andrés León Batallas**

**Director:**

**MSc. Byron Ramiro Romero Romero**

**Milagro, septiembre de 2022**

**ECUADOR**

## ACEPTACIÓN DEL TUTOR

Yo, **MSc. Byron Ramiro Romero Romero** en mi calidad de director del trabajo de titulación, elaborado por **Alberto Andrés León Batallas**, cuyo tema es “**Modelado y simulación de la cinemática de un robot manipulador de cinco grados de libertad mediante MATLAB.**”, que aporta a la Línea de Investigación **Desarrollo productivo**, previo a la obtención del Grado de **Magíster en Matemática** con Mención en Modelación Matemática.

Por la presente hago constar que he analizado el proyecto de grado presentado por el **MSc. Alberto Andrés León Batallas**, para optar al título de **Magister en Matemática** y que acepto tutorar al estudiante, durante la etapa del desarrollo del trabajo hasta su presentación, evaluación y sustentación.

Milagro, 6 de diciembre de 2021



Firmado electrónicamente por:  
**BYRON RAMIRO  
ROMERO ROMERO**

---

**NOMBRE:** MSc. BYRON RAMIRO ROMERO ROMERO

**CEDULA:** 0603003302

## DECLARACIÓN DE AUTORÍA DE LA INVESTIGACIÓN.

El autor de esta investigación declara ante el Comité Académico del Programa de Maestría en Matemática de la Universidad Estatal de Milagro, que el trabajo presentado es de mi propia autoría, no contiene material escrito de otra persona, salvo el que está referenciado debidamente en el texto; parte del presente documento o en su totalidad no ha sido aceptado para el otorgamiento de cualquier otro Título de una institución nacional o extranjera.

Milagro, 26 de septiembre de 2022



Firmado electrónicamente por:  
ALBERTO ANDRÉS  
LEÓN BATALLAS

---

**NOMBRE:** ALBERTO ANDRÉS LEÓN BATALLAS

**CÉDULA:** 0704304450

## VICERRECTORADO DE INVESTIGACIÓN Y POSGRADO CERTIFICACIÓN DE LA DEFENSA

El TRIBUNAL CALIFICADOR previo a la obtención del título de **MAGÍSTER EN MATEMÁTICA CON MENCIÓN MODELACIÓN MATEMÁTICA**, otorga al presente proyecto de investigación en las siguientes calificaciones:

TRABAJO DE TITULACION	59.00
DEFENSA ORAL	40.00
PROMEDIO	99.00
EQUIVALENTE	<b>Excelente</b>



Firmado electrónicamente por:  
**LUIS ANGEL  
BUCHELI**

---

**Msc. BUCHELI CARPIO LUIS ANGEL  
PRESIDENTE/A DEL TRIBUNAL**



Firmado electrónicamente por:  
**RAUL RUPERTO  
PANCHEZ  
HERNANDEZ**

---

**Msc. PANCHEZ HERNANDEZ RAUL RUPERTO  
VOCAL**



Firmado electrónicamente por:  
**JUAN JOSE  
PAREDES  
QUEVEDO**

---

**Mgs PAREDES QUEVEDO JUAN JOSE  
SECRETARIO/A DEL TRIBUNAL**

## DEDICATORIA

Dedico este trabajo principalmente a Dios, por haberme dado la vida y permitirme haber llegado hasta este momento tan importante de mi formación profesional.

A mi esposa Milly por su amor incondicional, que durante todo este tiempo ha sido paciente y me ha apoyado en todas mis decisiones.

A mi hijo Maxito, que con su amor y ternura, me ha motivado a seguir adelante.

A mis padres, Jorge y Mirtha, por darme el mejor ejemplo, y por demostrarme siempre su cariño y apoyo en cada decisión que he tomado.

A mis hermanos Jorge y Michelle, que siempre han estado junto a mí, brindándome su cariño y consejos.

## **AGRADECIMIENTO**

A Dios, por haberme dado las fuerzas y salud para realizar este trabajo.

A mi esposa Milly y a mi hijo Maxito, por su amor y comprensión en todo momento.

A mis padres, Jorge y Mirtha, por ser mi ejemplo y apoyo en cada etapa.

A mis hermanos Jorge y Michelle, por su apoyo incondicional.

Al personal docente por brindarme la oportunidad de adquirir nuevos conocimientos, especialmente al MSc. Byron Romero, que en su calidad de tutor y amigo, me ha guiado de la mejor manera en el desarrollo de este trabajo

A las autoridades de la Universidad Estatal de Milagro por su constante apoyo al crecimiento profesional de la planta docente.

## **CESIÓN DE DERECHOS DE AUTOR**

**Sr. Dr.**

**Fabricio Guevara Viejo.**

Rector de la Universidad Estatal de Milagro

Presente.

Mediante el presente documento, libre y voluntariamente procedo a hacer entrega de la Cesión de Derechos del Autor de Trabajo realizado como requisito previo a la obtención de mi Título de Cuarto Nivel cuyo tema fue “**Modelado y simulación de la cinemática de un robot manipulador de cinco grados de libertad mediante MATLAB.**” y que corresponde al Vicerrectorado de Investigación y Postgrado.

Milagro, 16 de junio de 2022.

---

**NOMBRE: ALBERTO ANDRÉS LEÓN BATALLAS**

**CÉDULA: 0704304450**

## ÍNDICE GENERAL

1.	Introducción.....	1
1.1.	Situación problemática.....	1
1.2.	Formulación del Problema.....	2
2.	Objetivos.....	3
2.1.	Objetivo general.....	3
2.2.	Objetivos específicos.....	3
3.	Planteamiento hipotético.....	3
4.	Marco teórico.....	4
4.1.	Antecedentes de la investigación (estado del arte).....	4
4.2.	Bases teóricas.....	9
4.2.1	Robot manipulador.....	9
4.2.2	Cinemática directa.....	10
4.2.3.	Método de Denavit - Hartenberg.....	11
4.2.3	Cinemática inversa.....	13
5.	Materiales y métodos.....	14
5.1	Robot CRS CataLyst-5.....	15
5.2	Desarrollo de la Cinemática Directa.....	17
5.3	Desarrollo de la cinemática inversa.....	22
6.	Resultados y discusión.....	32
7.	Conclusiones y recomendaciones.....	42
	Conclusiones.....	42
	Recomendaciones.....	43
8.	Referencias bibliográficas.....	43
9.	Anexos.....	46



## ÍNDICE DE FIGURAS

Figura 1 <i>Diseño de la estación de trabajo en el entorno RobotStudio</i> .....	7
Figura 2 <i>Manipulador robótico para servicio</i> .....	8
Figura 3 <i>Tipos de articulaciones empleadas en robots</i> .....	9
Figura 4 <i>Manipulador robótico y sus partes equivalentes en el cuerpo humano</i> .....	10
Figura 5 <i>Parámetros D-H en un eslabón giratorio</i> .....	13
Figura 6 <i>Límites angulares de cada articulación del robot CRS CataLyst-5.</i> .....	16
Tabla 1 <i>Espacio de trabajo del robot y velocidades máximas</i> .....	17
Figura 7 <i>Espacio de trabajo del robot CRS CataLyst-5</i> .....	17
Figura 8 <i>Configuración del robot y ejes utilizados en parámetros DH</i> .....	18
Tabla 2 <i>Tabla de Parámetros de Denavit Hartenberg</i> .....	18
Figura 9. <i>Desacoplo cinemático del robot manipulador</i> .....	23
Figura 10 <i>Plano XY del robot para cálculo de <math>q_1</math></i> .....	26
Figura 11 <i>Plano XZ del robot para cálculo de <math>q_3</math></i> .....	26
Figura 12 <i>Posición del robot para el cálculo de <math>q_2</math></i> .....	28
Figura 13 <i>Posición especial del robot</i> .....	29
Figura 14 <i>Simulación de la cinemática directa</i> .....	32
Figura 15 <i>Validación ajustada de la cinemática directa</i> .....	33
Tabla 3 <i>Combinación de ecuaciones de cinemática inversa</i> .....	34
Figura 16 <i>Resultados de la Validación de la cinemática inversa</i> .....	41

## ÍNDICE DE TABLAS

Tabla 1 <i>Espacio de trabajo del robot y velocidades máximas</i> .....	17
Tabla 2 <i>Tabla de Parámetros de Denavit Hartenberg</i> .....	18
Tabla 3 <i>Combinación de ecuaciones de cinemática inversa</i> .....	34

## Glosario de términos

${}^{i-1}A_i$	Matriz de transformación básica
$a_i$	Distancia de traslación a lo largo del eje $x_i$
$\alpha_i$	Ángulo de rotación alrededor del eje $x_i$ .
CAD	Computer-Aided Design
D-H	Denavit – Hartenberg
$d_i$	Distancia de traslación a lo largo del eje $z_{i-1}$
${}^0h$	Vector del punto central de la muñeca hacia el punto deseado
${}^{i-1}R_i$	Matriz de rotación
${}^0r_m$	Vector de coordenadas del punto central de la muñeca
${}^0r_p$	Vector de coordenadas del punto de posicionamiento deseado
STEM	Science, Technology, Engineering and Mathematics
T	Matriz de transformación homogénea
$\theta_i$	Ángulo de rotación alrededor del eje $z_{i-1}$ .

## Resumen

En la actualidad, los robots se han convertido en una parte importante para el crecimiento de los sistemas de fabricación industrial. El desarrollo de la Industria 4.0, impulsa a las organizaciones a la automatización de procesos, lo que ha provocado el aumento de los estudios académicos en el campo de la robótica industrial y de servicios, a menudo los investigadores necesitan comprender y analizar el aspecto cinemático de los robots. La obtención y planteamiento de los modelos cinemáticos directo e inverso de un robot manipulador, pueden ser procesos muy extensos y con una gran carga matemática, con cálculos complejos. Por lo tanto, en este trabajo se presenta el desarrollo detallado del modelamiento matemático de la cadena cinemática de un robot manipulador de cinco grados de libertad que, por su diseño estructural, se encuentra fácilmente en la industria y laboratorios de investigación. Para el problema cinemático directo se emplea el conocido Método de Denavit – Hartenberg, y la cinemática inversa se resuelve mediante el método de desacoplamiento cinemático. Los modelos son validados mediante MATLAB, en donde se podrá visualizar una simulación tridimensional del robot manipulador, la cual permitirá al usuario realizar una operación de los movimientos y trayectorias desarrolladas por las articulaciones. Y con esto, poder generar un recurso de aprendizaje en el campo de la robótica, el cual podrá tener fines académicos y de investigación.

**Palabras claves:** Cinemática directa, cinemática inversa, grados de libertad, robot manipulador.

## **Abstract**

Today, robots have become an important part of the growth of industrial manufacturing systems. The development of Industry 4.0, drives organizations to process automation, which has caused the increase in academic studies in the field of industrial and service robotics, often researchers need to understand and analyze the cinematographic aspect of the robots. Obtaining and approaching the direct and inverse kinematic models of a manipulator robot can be very extensive processes and with a great mathematical load, with complex calculations. Therefore, in this work, the detailed development of the mathematical modeling of the kinematic chain of a manipulator robot with five degrees of freedom is presented, which, due to its structural design, is easily found in industry and research laboratories. For the direct kinematics problem, the well-known Denavit-Hartenberg Method is used, and the inverse kinematics is solved by the kinematic decoupling method. They are validated using MATLAB, where a three-dimensional simulation of the manipulator robot can be visualized, the models that will allow the user to perform an operation of the movements and trajectories developed by the joints. And with this, to be able to generate a learning resource in the field of robotics, which may have academic and research purposes.

**Keywords:** direct kinematics, degrees of freedom, inverse kinematics, robotic manipulator.

## **1. Introducción**

La robótica es un área cada vez más explorada, la cual ha ido creciendo en las últimas décadas, ya que nos permite realizar movimientos más precisos y complejos en comparación a los efectuados por la acción humana, por lo tanto, se pueden desarrollar ciertas actividades de alto riesgo de una manera más segura y confiable.

La utilización generalizada de robots manipuladores durante la actual etapa de desarrollo industrial, ha llevado a mejoras en la productividad y en la calidad de los productos fabricados, principalmente debido a la mejor repetitividad de los movimientos del robot, lo que en consecuencia ha resultado en una mayor precisión en su desempeño.

En la actualidad, se busca planificar los movimientos de un robot en un entorno virtual, sin la necesidad de contar con el equipamiento físico, debido a que se lo puede realizar mediante una simulación en software, que permitirá conocer el accionamiento del manipulador y su respectivo estudio cinemático. Esto, tiene una gran connotación tanto en campo industrial como en el académico.

Cada vez más universidades e incluso colegios ofrecen cursos sobre robótica y otras tecnologías modernas de fabricación. Tanto los estudiantes como los docentes perciben estos cursos como valiosos e importantes, ya que preparan a la próxima generación de ingenieros para ingresar a un área altamente competitiva como la industria.

### **1.1. Situación problemática**

La robótica es una disciplina de la ingeniería muy atractiva y cada vez más importante. Con el creciente mercado de la robótica y las perspectivas muy brillantes para el futuro, es de esperar que la demanda de robótica educativa incluso crezca. Además, es muy recomendable que los estudiantes de ingeniería en general se familiaricen más con la tecnología robótica para afrontar los retos del futuro. Además, debido a su atractivo, la robótica juega un papel importante en la educación STEM (Science, Technology, Engineering and Mathematics).

Por lo tanto, es muy importante desarrollar metodologías de enseñanza que permitan un proceso educativo eficiente y generalmente accesible en el campo de la robótica.(Cehovin et al., 2017).

La implementación de la simulación en la robótica, se basa en el uso de una interfaz gráfica, donde se puede observar la interacción de los parámetros y variables del sistema (Cakir & Butun, 2007).

Para el desarrollo del problema cinemático se quieren conocimientos de algebra lineal, programación, física, electrónica y mecánica, de los cuales se basan para el planteamiento y desarrollo de los modelos matemáticos mediante uso de software (Gil et al., 2015).

Hay que tener en cuenta que una de las grandes preocupaciones en una clase de robótica es el presupuesto limitado (Jung, 2013). Por lo cual, es conveniente el uso de simuladores que permiten describir el comportamiento de los sistemas robóticos.

Las carreras de ingeniería de la Universidad Estatal de Milagro, dentro de las instalaciones de sus laboratorios no cuentan con el equipamiento físico de un robot manipulador, por lo que, con el desarrollo de este proyecto se pretende tener un insumo para el modelamiento y simulación de un robot manipulador, para reforzar el proceso de enseñanza aprendizaje en el campo de la robótica.

## **1.2. Formulación del Problema.**

El desarrollo de un entorno virtual de una estación de trabajo puede aportar muchas ventajas en la enseñanza de asignaturas técnicas. La simulación y monitoreo de sistemas de procesos tecnológicos, se ha convertido en una herramienta importante para el diseño de productos, la cual puede ser aplicada tanto en el campo industrial como en el de investigación, ya que de esta manera se pueden verificar y validar nuevos conceptos antes de ser aplicados en la práctica.

Cabe destacar que, ante la problemática actual, por la pandemia del COVID-19, no se ha permitido un retorno total a las aulas, por lo cual los estudiantes no pueden realizar las prácticas presenciales. En la actualidad, se busca el desarrollo de simuladores y entornos virtuales en el campo de la robótica para cumplir distintos objetivos, entre ellos, la enseñanza a estudiantes

en el uso y programación de robots manipuladores, que faciliten la planificación de tareas, que permitan hacer los cálculos necesarios y estudios cinemáticos, similares a los entornos reales como son las industrias y sus procesos productivos.

Para poder generar una herramienta de simulación robótica, primero se tiene que partir por el modelamiento matemático de la cinemática, donde se realiza la descripción de la posición y orientación del efector final tomando en cuenta la geometría del robot, se determinan matrices traslación y rotación, las cuales representan la posición y orientación de un elemento en el espacio, se reconocen las trayectorias de cada una de las articulaciones y los puntos de singularidad. Para el desarrollo de lo antes mencionado, se tienen que aplicar conocimientos en el campo de la robótica, modelación matemática, álgebra lineal, programación y simulación de procesos.

## **2. Objetivos**

### **2.1. Objetivo general.**

Modelar matemáticamente la cadena cinemática de un robot manipulador de cinco grados de libertad y simularlo mediante MATLAB.

### **2.2. Objetivos específicos.**

1. Determinar el modelamiento de la cinemática directa
2. Determinar el modelamiento de la cinemática inversa
3. Desarrollar el algoritmo para la simulación de la cinemática directa del robot manipulador mediante MATLAB
4. Desarrollar el algoritmo para la validación de la cinemática inversa del robot manipulador mediante MATLAB

## **3. Planteamiento hipotético**

Como se trata de un problema de modelación matemática de un robot manipulador, las preguntas que se plantean como hipótesis en la investigación son:

- ¿Cómo determinar el modelamiento matemático de la cinemática directa de un robot de cinco grados de libertad?



- ¿Cómo determinar el modelamiento matemático de la cinemática inversa de un robot de cinco grados de libertad?
- ¿Cómo simular la cinemática directa de un robot de cinco grados de libertad utilizando MATLAB?
- ¿Cómo validar el modelamiento matemático cinemática inversa de un robot de cinco grados de libertad utilizando MATLAB?

#### **4. Marco teórico**

##### **4.1. Antecedentes de la investigación (estado del arte).**

La modelación matemática es una herramienta muy potente que permite analizar, describir y optimizar el comportamiento de sistemas productivos, los cuales pueden ser representados en entornos gráficos de simulación. La robótica es una rama de la ingeniería que combina varias disciplinas como la mecánica, electrónica e informática, destacando un fuerte componente matemático como el álgebra lineal. El comportamiento de un robot puede ser representado mediante el modelamiento matemático de la cinemática directa e inversa del mismo.

El desarrollo de la innovación educativa en el campo de la robótica, tiene como finalidad que los estudiantes adquieran conceptos básicos y de la forma más práctica posible, ya que, en los últimos años, se ha incrementado considerablemente la tendencia sobre la implementación de dispositivos robóticos en la Industria, lo cual aporta el desarrollo productivo, debido a esto, las universidades han visto la necesidad de generar nuevos recursos para la enseñanza, y así los estudiantes puedan estar mejor preparados para el mundo laboral. Considerando lo antes mencionado, se puede destacar el desarrollo de uno de los primeros laboratorios remotos con entorno virtual presentado en (Torres et al., 2006), donde su componente principal es RoboLab, un sistema que permite al alumno interactuar con robots simulados y reales a través de Internet.

Con la finalidad de proporcionar herramientas para la enseñanza de la robótica, en (Jara et al., 2007) desarrollan un laboratorio para la simulación y teleoperación de robots industriales, en donde las aplicaciones fueran elaboradas con software de libre acceso (open-source), utilizando Java. Donde se permite que el

estudiante pueda ser adiestrado de una manera rápida y económica. De esta manera, estarán más preparados para manipular y resolver los problemas de los complejos dispositivos robóticos. Destacan que el procedimiento para transformar el sistema robótico en un interactivo laboratorio virtual fue más sencillo que con cualquier otro programa.

En (Urrea & Kern, 2012) se presenta el modelado de un robot manipulador redundante tipo SCARA con cinco grados de libertad. Se desarrolla en un entorno de simulación utilizando las herramientas de programación MATLAB/Simulink. Este entorno de simulación se emplea para realizar varias pruebas del manipulador redundante, donde logran realizar el modelo cinemático y dinámico utilizando los métodos de Denavit-Hartenberg y Lagrange-Euler, respectivamente.

En (Crespo et al., 2013) realizan un proyecto, en donde se puede destacar la innovación educativa, proponen un entorno en Realidad Virtual como herramienta para el aprendizaje inmersivo. Este trabajo abarca algunas asignaturas a nivel de ingeniería como robótica, mecánica y electrónica. Se destaca la utilización de dos técnicas de Realidad Virtual: Un ambiente virtual para dispositivos móviles, y otro ambiente virtual inmersivo e interactivo utilizando el Oculus Rift, un casco de realidad virtual. Para la materia de Robótica y Mecánica se desarrollaron objetos de aprendizaje en 3D, donde se utilizan brazos robóticos y generadores eólicos, que pueden ser visualizados por los estudiantes utilizando dispositivos móviles y ordenadores.

En (Medina Cervantes et al., 2014), se presenta un simulador en 3D desarrollado para el control de un brazo robot antropomórfico de cuatro grados de libertad, que tiene como objetivo apoyar el aprendizaje de estudiantes de ingeniería en el campo de robótica. El entorno gráfico del simulador en 3D del brazo robótico fue elaborado mediante un paquete de herramientas de realidad virtual de MATLAB, empleando V-Realm Builder.

El desarrollo de software para la simulación y control de un robot industrial KUKA IRB-2600, es presentado en (Lazaro-Arvizu et al., 2015) como una herramienta para controlar y simular los movimientos del robot, usando aplicaciones como SIMULINK/MATLAB en donde se ejecuta la simulación en un

interfaz gráfica que permite la interacción con el usuario y así evita movimientos peligrosos en el robot real, en caso de contar con él.

A nivel regional, en una prestigiosa universidad de Chile, (Urrea & Coltters, 2015), proponen una nueva metodología para el diseño e implementación de simulaciones tridimensionales de robots manipuladores, tanto en tareas individuales como cooperativas, utilizando diversas herramientas de programación y diseño 3D. Donde obtienen los modelos matemáticos de movimiento para los sistemas representados, independientemente del tipo de robot manipulador a considerar, también presentan el desarrollo de un algoritmo computacional. Además, incorporan la detección de colisiones entre elementos constitutivos del sistema, considerando elementos externos a los robots, colisiones entre eslabones de un mismo robot, o entre dos robots. Utilizan una plataforma de programación C++, también diseñan una interfaz gráfica de usuario flexible e intuitiva.

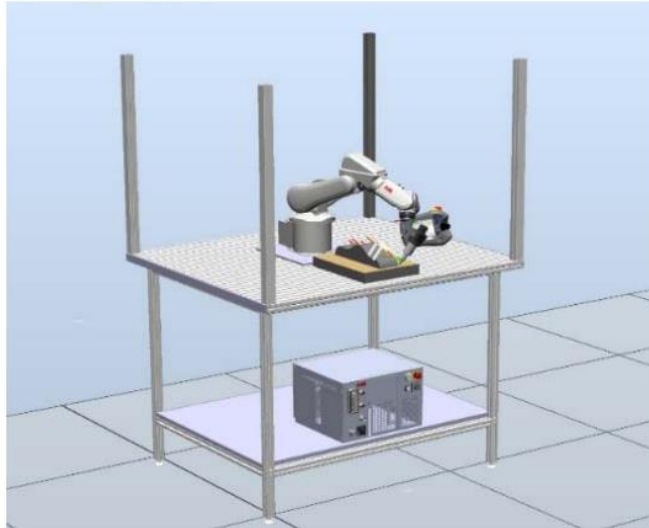
La robótica ha sido utilizada en aplicaciones de sistemas de manufactura flexible. Hay que considerar que el proceso de aprendizaje y capacitación de un ingeniero en el campo de la robótica requiere mucho tiempo, además de que resulta complicado enseñar en robots reales y en tiempo real. Por lo que en (Hwang et al., 2016) desarrollan un método de programación fuera de línea, para poder generar la codificación a partir de datos en formato de diseño CAD.

A nivel local se puede destacar (González Alvarez, 2017), la implementación de un entorno de simulación para un robot paralelo activado por ocho cables, inicialmente presenta conceptos básicos de robots paralelos para dar paso a la modelación de la cinemática directa e inversa, finalmente obtienen una herramienta que permite conocer posición factible de la estructura móvil del robot en una determinada posición y con cierta orientación, e indica las singularidades de ser el caso.

En (Velíšek et al., 2017), desarrollaron un proyecto en el cual presentan el diseño de una estación de trabajo robotizada haciendo uso de la integración de modelos CAD y software de simulación robótica como forma de emparejar y comparar entornos reales y virtuales; en este trabajo abordan la creación de las piezas y tiene como principal objetivo su integración en un entorno de simulación RobotStudio, dado según la concepción y lineamientos de la Industria 4.0, donde

la pieza en sí se convierte en la entidad portadora de la mayor cantidad de información necesaria en cualquier momento de un proceso de producción.

**Figura 1** *Diseño de la estación de trabajo en el entorno RobotStudio.*



*Fuente:* (Velíšek et al., 2017)

(Liu et al., 2017) presentan su trabajo sobre el diseño de un manipulador robótico de tres grados de libertad, cuyas ecuaciones cinemáticas se establecieron por el método Denavit - Hartenberg. La solución inversa de las ecuaciones cinemáticas del manipulador robótico, se establecieron a partir del análisis de la configuración estructural. Para la validación de los modelos, se obtuvieron las curvas de velocidad y aceleración, la cuales fueron desarrolladas en el simulador ADAMS.

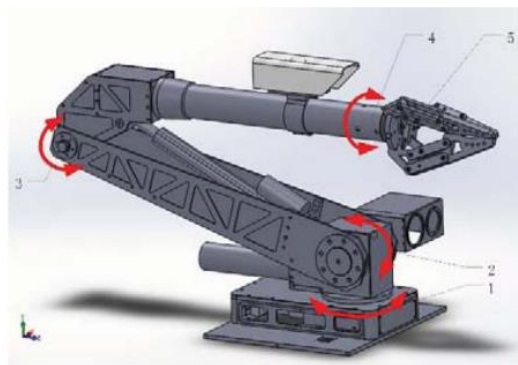
En (Ndawula & Assal, 2018), se han centrado en el diseño conceptual, los análisis cinemáticos, la síntesis dimensional óptima, el análisis del espacio de trabajo y la planificación de la trayectoria de un novedoso robot de trasplante de plántulas de maceta de campo abierto con pinzas múltiples de tres grados de libertad. Esto es un ejemplo que los robots manipuladores no necesariamente están aplicado a la industria de manufactura, también están inmerso en otras áreas como la agricultura.

En el trabajo realizado por (Vithanage et al., 2019), presentan una síntesis cinemática directa e inversa detallada de un robot comercial industrial articulado de seis ejes. Las soluciones derivadas de los problemas cinemáticos tanto

directos como inversos se validaron con el software de simulación basado en CAD proporcionado por el proveedor del robot, que imita con precisión el comportamiento real del robot. El problema cinemático directo se resuelve empleando la convención D-H clásica y las propiedades de la muñeca esférica del robot que satisfacen la condición de Piper se han utilizado para obtener soluciones analíticas de forma cerrada para el problema cinemático inverso.

En (Kang & Chou, 2019) analizan las características cinemáticas de un manipulador robótico de cinco grados de libertad diseñado para un robot de servicio y presentan su método de manipulación, ver Figura 2. La cinemática del sistema del brazo mecánico, incluye la cinemática directa e inversa y se realizan experimentos para validar su corrección. El mecanismo y la cinemática del efector final de tres dedos también se analizan y simulan. Debido a la dificultad para establecer con precisión la función del modelo mecánico para el efector final, también presentan un modelo de control del efector final basado en el algoritmo de red neuronal.

**Figura 2** Manipulador robótico para servicio



*Fuente:* (Kang & Chou, 2019)

Finalmente se puede destacar el trabajo de simulación realizado por (Jha et al., 2021), en donde seleccionan a el robot industrial KUKA 3R Arc, el cual es un manipulador de seis grados de libertad; en el que realizan el análisis de la cinemática directa e inversa, utilizan método Denavit-Hartenberg y utilizan para la simulación del robot, un software de bajo costo llamado Roboanalyzer, que sirve tanto para el diseño y análisis de la cinemática.

## 4.2. Bases teóricas.

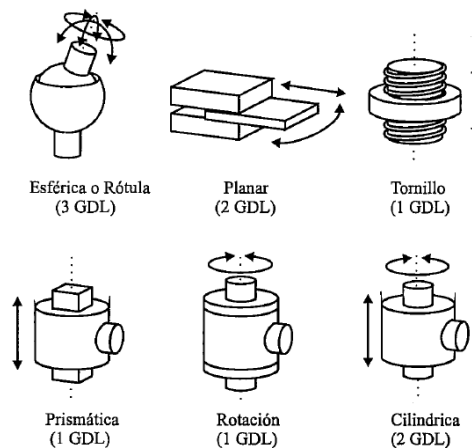
### 4.2.1 Robot manipulador

Un robot manipulador es un mecanismo multifuncional reprogramable, integrado por una serie de articulaciones, que tiene como principal objetivo el desplazamiento y orientación de herramientas, piezas y objetos, a través de movimientos variables programados, para el desempeño de diversas tareas. (Barrientos et al., 2007)

El robot manipulador está dentro de la clasificación de los robots tipo serie, el cual mecánicamente puede estar compuesto por eslabones, articulaciones y actuadores. Las articulaciones generalmente pueden ser prismáticas (de translación) y rotativas (de rotación), los eslabones son rígidos; el empleo de diferentes combinaciones de articulaciones de un robot, da como resultado diferentes configuraciones, según su aplicación. (López Cruz, 2018)

Cada componente de movimiento que permite una articulación se llama grado de libertad (GDL). En la Figura 3, se puede apreciar los diferentes tipos de articulaciones empleadas en robots.

**Figura 3** Tipos de articulaciones empleadas en robots

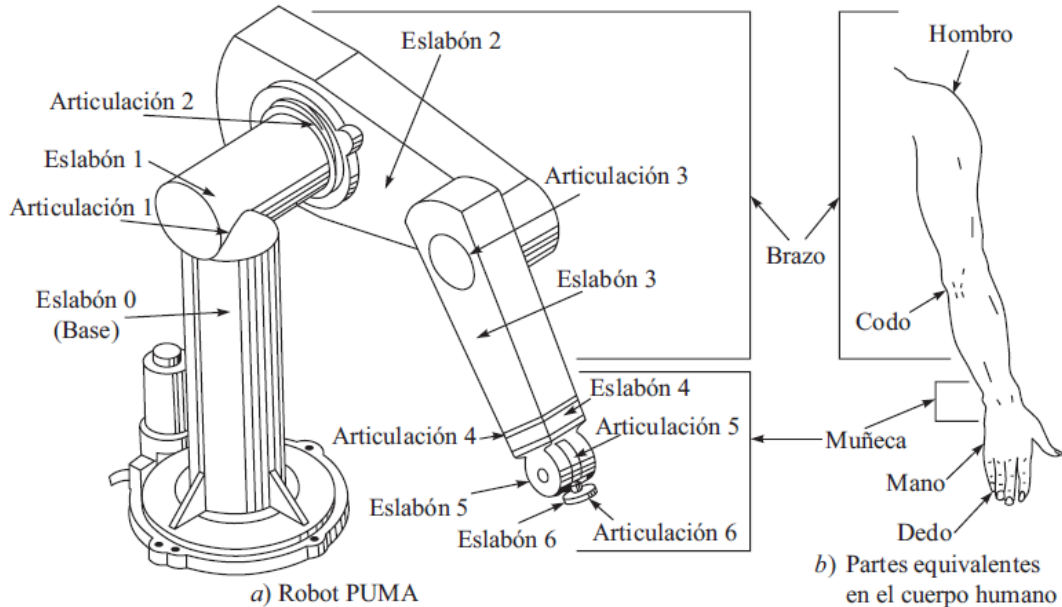


*Fuente:* (Barrientos et al., 2007)

La configuración física estructural de la mayor parte de los robots manipuladores industriales, tiene una gran similitud con la anatomía del brazo humano, por lo que se puede hacer analogía a los distintos elementos que

componen al manipulador, y se usan términos como hombro, brazo, codo y muñeca. (Subir, 2008)

**Figura 4** Manipulador robótico y sus partes equivalentes en el cuerpo humano



Fuente: (Subir, 2008)

#### 4.2.2 Cinemática directa

La cinemática del robot es la ciencia que estudia el movimiento del manipulador con respecto a un sistema de coordenadas de referencia fijo, sin considerar las fuerzas que originan el movimiento. El estudio pretende describir la analítica del desplazamiento espacial del robot como función del tiempo, y en particular las relaciones entre la posición y orientación del extremo final del robot y los valores que toman sus coordenadas articulares. (Fu et al., 1988)

La cinemática directa consiste en obtener la posición y orientación del efector final, con respecto al sistema de coordenadas de referencia, tomando en cuenta la geometría del robot, se plantean matrices de transformación homogéneas, las cuales representan la posición y orientación de un elemento en el espacio. (Ollero, 2001)

La matriz de transformación homogénea es una matriz de orden cuatro, que transforma un vector expresado en coordenadas homogéneas desde un sistema de coordenadas hasta otro sistema de coordenadas. La matriz de transformación homogénea tiene la siguiente estructura:

$$T = \begin{bmatrix} \text{rotación} & \text{traslación} \\ \text{perspectiva} & \text{escalado} \end{bmatrix}$$

$$T = \begin{bmatrix} n_x & o_x & a_x & P_x \\ n_y & o_y & a_y & P_y \\ n_z & o_z & a_z & P_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1)$$

$$T = \begin{bmatrix} \mathbf{n} & \mathbf{o} & \mathbf{a} & \mathbf{p} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2)$$

Donde  $\mathbf{n}$ ,  $\mathbf{o}$ ,  $\mathbf{a}$  es una terna de vectores ortogonales unitarios que representa la orientación y  $\mathbf{p}$  es un vector que representa la posición del origen del sistema actual respecto al sistema de referencia.

Existen métodos sistemáticos para situar los sistemas de coordenadas asociados a cada eslabón y determinar la cadena cinemática del robot, uno de los más utilizados es el Método de Denavit-Hartenberg, el cual nos permite resolver el problema cinemático directo.

#### 4.2.3. Método de Denavit - Hartenberg

El método de Denavit – Hartenberg (D-H), es un algoritmo que se aplica a robots manipuladores, que consiste en una serie de reglas, para situar los sistemas de referencia de cada eslabón del robot y obtener la cadena cinemática del robot. (Denavit & Hartenberg, 1955)

Permite el paso de un eslabón al siguiente mediante cuatro transformaciones básicas, que dependen exclusivamente de las características constructivas del robot.

Las transformaciones básicas que relacionan el sistema de referencia del elemento  $i$  con el sistema del elemento  $i-1$ , son:

- a) Rotación ( $\theta$ ) de cada uno de los ejes  $z_{i-1}$ .
- b) Traslación ( $d$ ) a lo largo de cada uno de los ejes  $z_{i-1}$ .
- c) Traslación ( $a$ ) a lo largo de cada uno de los ejes  $x_i$ .
- d) Rotación ( $\alpha$ ) de cada uno de los ejes  $x_i$ .



Para representación matemática del método D-H, hay que considerar que el producto de matrices no es conmutativo, las transformaciones se tienen que realizar en el orden indicado:

$${}^{i-1}A_i = T(z, \theta_i) T(0, 0, d_i) T(a_i, 0, 0) T(x, \alpha_i) \quad (3)$$

Por lo tanto, la expresión (1) representa la matriz de transformación básica:

$${}^{i-1}A_i = \begin{bmatrix} c\theta_i & -c\alpha_i s\theta_i & s\alpha_i s\theta_i & a_i c\theta_i \\ s\theta_i & c\alpha_i c\theta_i & -s\alpha_i c\theta_i & a_i s\theta_i \\ 0 & s\alpha_i & c\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4)$$

Donde  $\theta_i$ ,  $d_i$ ,  $a_i$ ,  $\alpha_i$  son los parámetros de D-H del eslabón  $i$ , ver Figura 5. De esta manera se procede al planteamiento del algoritmo para la resolución del problema cinemático directo, que se describe a continuación: (Barrientos et al., 2007)

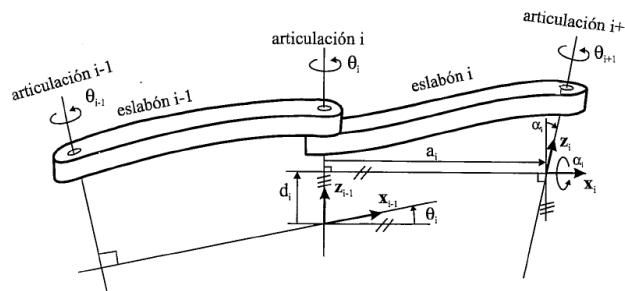
- 1) Numerar los eslabones comenzando con 1 (primer eslabón móvil de la cadena) y acabando con  $n$  (último eslabón móvil). Se numerará como eslabón 0 a la base fija del robot.
- 2) Numerar cada articulación comenzando por 1 (la correspondiente al primer grado de libertad) y acabando en  $n$ .
- 3) Localizar el eje de cada articulación. Si ésta es rotativa, el eje será su propio eje de giro. Si es prismática, será el eje a lo largo del cual se produce el desplazamiento.
- 4) Para  $i$  de 1 a  $n-1$  situar el eje  $z_i$  sobre el eje de la articulación  $i+1$ .
- 5) Situar el origen del sistema de la base  $\{S_0\}$  en cualquier punto del eje  $z_0$ . Los ejes  $x_0$  e  $y_0$  se situarán de modo que formen un sistema dextrógiro con  $z_0$ .
- 6) Para  $i$  de 1 a  $n-1$ , situar el sistema  $\{S_i\}$  (solidario al eslabón  $i$ ) en la intersección del eje  $z_i$  con la línea normal común a  $z_{i-1}$  y  $z_i$ . Si ambos ejes se cortasen se situaría  $\{S_i\}$  en el punto de corte. Si fuesen paralelos  $\{S_i\}$  se situaría en la articulación  $i+1$ .
- 7) Para  $i$  de 1 a  $n-1$ , situar  $x_i$  en la línea normal común a  $z_{i-1}$  y  $z_i$ .
- 8) Para  $i$  de 1 a  $n-1$ , situar  $y_i$  de modo que forme un sistema dextrógiro con  $x_i$  y  $z_i$ .
- 9) Situar el sistema  $\{S_n\}$  en el extremo del robot de modo que  $z_n$  coincida con la dirección de  $z_{n-1}$  y  $x_n$  sea normal a  $z_{n-1}$  y  $z_n$ .

- 10) Obtener  $\theta_i$  como el ángulo que hay que girar en torno a  $z_{i-1}$  para que  $x_{i-1}$  y  $x_i$  queden paralelos.
- 11) Obtener  $d_i$  como la distancia, medida a lo largo de  $z_{i-1}$ , que habría que desplazar  $\{S_{i-1}\}$  para que  $x_i$  y  $x_{i-1}$  quedasen alineados.
- 12) Obtener  $a_i$  como la distancia medida a lo largo de  $x_i$ , que ahora coincidiría con  $x_{i-1}$ , que habría que desplazar el nuevo  $\{S_{i-1}\}$  para que su origen coincidiese con  $\{S_i\}$ .
- 13) Obtener  $\alpha_i$  como el ángulo que habría que girar en torno a  $x_i$ , que ahora coincidiría con  $x_{i-1}$ , para que el nuevo  $\{S_{i-1}\}$  coincidiese totalmente con  $\{S_i\}$ .
- 14) Obtener las matrices de transformación  ${}^{i-1}A_i$ .
- 15) Obtener la matriz de transformación que relaciona el sistema de la base con el del extremo del robot:

$$T = {}^0A_1 {}^1A_2 \dots {}^{n-1}A_n \quad (5)$$

- 16) La matriz  $T$  define la orientación (submatriz de rotación) y posición (submatriz de traslación) del extremo referidas a la base en función de las  $n$  coordenadas articulares.

**Figura 5** *Parámetros D-H en un eslabón giratorio*



*Fuente:* (Barrientos et al., 2007)

### 4.2.3 Cinemática inversa

El problema de la cinemática inversa consiste en la determinación de las coordenadas de articulares del robot  $q = [q_1, q_2, \dots, q_n]^T$  correspondientes a una orientación y posición específicas del efector final. (Reyes Cortés, 2011)

En este caso, hay que considerar que la resolución no es sistemática, como en el caso de la cinemática directa que hizo uso de método D-H, ya que el

procedimiento de determinación de las ecuaciones depende de la configuración del robot, y no siempre existe una solución cerrada. (Siciliano et al., 2009)

Resolver matemáticamente el problema cinemático inverso es más complejo, debido a que se deben determinar un conjunto de ecuaciones no lineales, las cuales estarán representadas en términos de funciones trigonométricas. Además, hay que indicar que podrán existir múltiples soluciones, y en algunos casos hay la posibilidad que no exista una solución, debido a la arquitectura del manipulador.

Debido a que no existen algoritmos generales para la solución del problema cinemático inverso, se puede plantear los siguientes métodos que depende de la configuración del robot:

- **Método geométrico**, es adecuado para robots de pocos grados de libertad, permite obtener los valores de las tres primeras variables articulares, que son las que consiguen posicionar el robot, para esto, se plantean relaciones geométricas y trigonométricas. (Craig, 2006)
- Resolución a partir de **matrices de transformación homogénea**, en donde se pretende despejar las “ $n$ ” variables  $q_i$  en función de los vectores **n, o, a, p**.
- **Desacoplamiento cinemático**, es utilizado en robots de mayor grado de libertad. El método consiste en realizar una separación de la orientación y posicionamiento. Para realizar esto, dada una posición y orientación final deseadas, establece las coordenadas del punto de corte de los tres últimos ejes (muñeca del robot) calculándose los valores de las tres primeras variables articulares ( $q_1, q_2, q_3$ ) que consiguen posicionar este punto. A partir de los datos de orientación y de los ya calculados, se obtiene los valores del resto de las variables articulares. (Barrientos et al., 2007)

## 5. Materiales y métodos

En el presente trabajo de titulación, la metodología de investigación aplicada es de tipo analítico con enfoque cuantitativo, esto porque se busca resolver los problemas cinemáticos: directo e inverso, de un robot manipulador

de cinco grados de libertad, y de esta manera evidenciar mediante las expresiones matemáticas el modelamiento de la cadena cinemática.

Para resolver el problema cinemático directo existen métodos alternativos como el de Matrices de transformación homogénea y Cuaternios, ya que permiten representar las transformaciones de rotación y desplazamiento. Pero el método más utilizado y el que se escoge para el presente trabajo es el algoritmo de Denavit-Hartenberg, ya que es un método sistemático que permite situar los sistemas de coordenadas asociados a cada eslabón y determinar la cadena cinemática directa del robot.

Para resolver el problema cinemático inverso existen métodos como el geométrico, matrices de transformación homogénea y el de desacoplamiento cinemático. Para hacer la cinemática inversa se decide utilizar la metodología del desacoplo cinemático, debido a que el robot cuenta con cinco grados de libertad, para lo cual se procede a dividir al manipulador en dos partes para luego acoplarlos vectorialmente.

Para la obtención de la validación de los modelos planteados, se utiliza el MATLAB. Además, se hace uso del mismo software para la simulación gráfica de la cinemática directa del robot manipulador.

Para el desarrollo de este proyecto, se ha escogido al robot CRS CataLyst-5, debido a que es un manipulador muy utilizado en el ámbito académico para el estudio y enseñanza en el campo de la robótica.

### **5.1 Robot CRS CataLyst-5**

El robot manipulador CRS CataLyst-5 de cinco grados de libertad es producido por el fabricante Thermo Electron Corporation.

El manipulador posee cinco ejes articulados que le permiten adoptar distintas orientaciones.

El robot Catalyst-5 tiene las siguientes características, que fueron tomadas de la hoja de especificaciones:

- Está constituido por cinco articulaciones rotacionales que le aportan 5 grados de libertad.
- El brazo tiene un peso de 19 kg.
- La capacidad de carga nominal es de 1 kg.

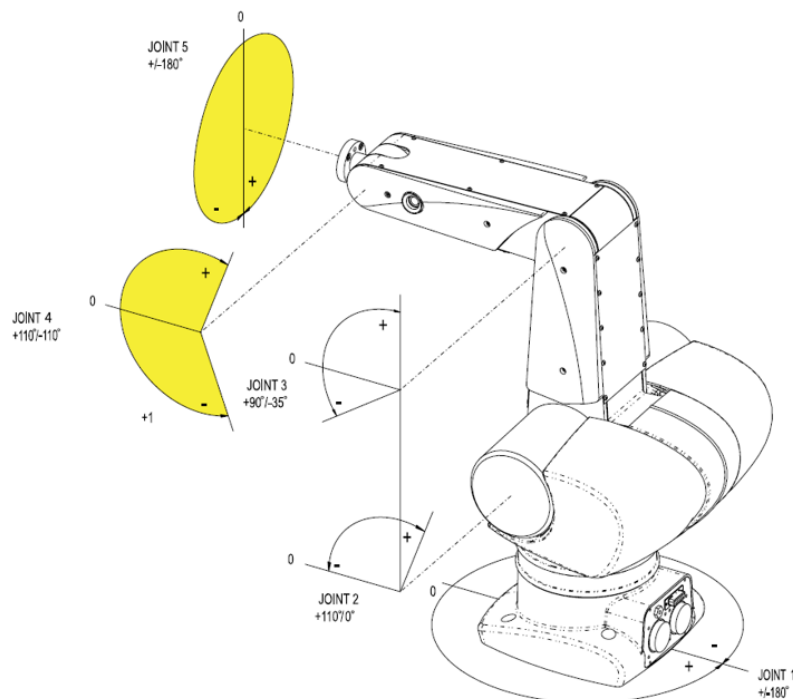
- La pinza es de accionamiento hidráulico. Permite coger piezas cilíndricas con diámetros comprendidos entre 35 mm y 40 mm y piezas prismáticas con espesores entre 16,2 mm y 29 mm, aproximadamente.
- El radio de alcance es de 560 mm.
- El giro del eje inferior de hasta 360°.
- La repetitividad del robot es  $\pm 0,05$  mm.
- El robot posee frenos en todos sus ejes.
- Los motores son de corriente continua con encoders ópticos incrementales en cada uno de sus ejes.

En el robot CRS CataLyst-5 se observan varias características diferenciales respecto de otros robots similares, entre las que se pueden citar las siguientes:

- Es rápido, robusto y de bajo costo.
- La posición “Home” se gestiona de forma automática.
- Es fácil de integrar con otros sistemas.

En la figura 6 se muestran los límites de giro de cada una de las articulaciones del robot.

**Figura 6** Límites angulares de cada articulación del robot CRS CataLyst-5.



*Fuente:* hoja de especificaciones CRS CataLyst-5

En la tabla 1 se observa la velocidad y el espacio de las articulaciones del robot:

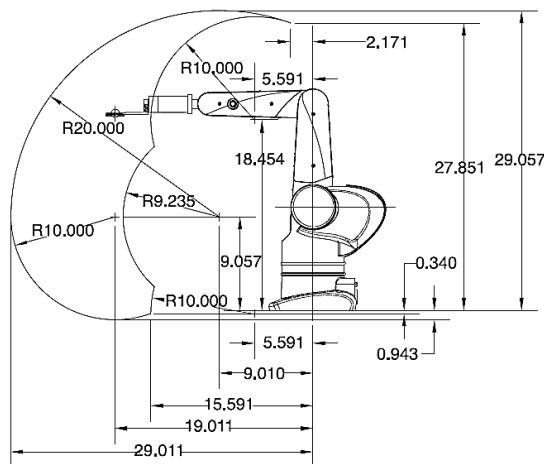
**Tabla 1** Espacio de trabajo del robot y velocidades máximas

EJE	ESPACIO DE TRABAJO	VELOCIDAD MÁXIMA
J1 (cintura)	360°	210°/s
J2(hombro)	110°	210°/s
J3(codo)	125°	210°/s
J4(lanzamiento de la muñeca)	220°	552°/s
J5(giro de la muñeca)	360°	1102°/s

Fuente: hoja de especificaciones CRS CataLyst-5

Si se quiere definir el rango de movimiento del robot CRS CataLyst-5, en la figura 7 se puede observar el espacio de trabajo, el cual es el máximo en que sus brazos pueden llegar a alcanzar sin tener en cuenta las pinzas, considerar que las dimensiones de la figura vienen dadas en pulgadas.

**Figura 7** Espacio de trabajo del robot CRS CataLyst-5.



Fuente: hoja de especificaciones CRS CataLyst-5

## 5.2 Desarrollo de la Cinemática Directa

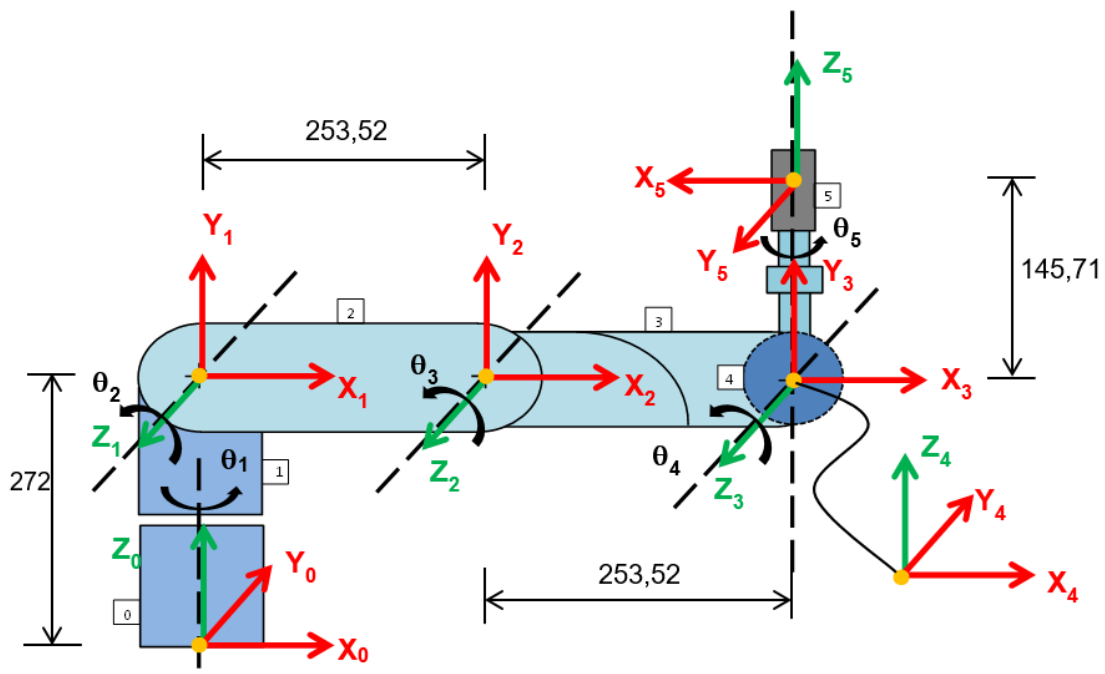
Para realizar la cinemática directa se utilizó el método de Denavit-Hartenberg. El cual se deben seguir dieciséis pasos y el resultado de los parámetros depende de la configuración inicial del robot y el posicionamiento de los ejes de cada articulación. Como resultado se obtiene una tabla que identifica los siguientes parámetros:

- Rotación ( $\theta_i$ ) de cada uno de los ejes  $z_{i-1}$ .
- Traslación ( $d_i$ ) a lo largo de cada uno de los ejes  $z_{i-1}$ .

- c) Traslación ( $a_i$ ) a lo largo de cada uno de los ejes  $x_i$ .
- d) Rotación ( $\alpha_i$ ) de cada uno de los ejes  $x_i$ .

Para la realización del algoritmo de Denavit-Hartenberg se fijó una configuración inicial del robot que permitiera que los cálculos de la cinemática inversa se pudieran realizar de manera más sencilla. La configuración del robot y la colocación de los ejes quedaron de la siguiente manera. Ver Figura 8, considerar que las dimensiones del esquema de articulaciones están en milímetros.

**Figura 8** Configuración del robot y ejes utilizados en parámetros DH.



Fuente: Elaboración propia

Como resultado del desarrollo del algoritmo de Denavit – Hartenberg, se obtuvo la Tabla 2, que describe los parámetros de rotación y translación.

**Tabla 2** Tabla de Parámetros de Denavit Hartenberg

	$\theta$	$d$	$a$	$\alpha$
1	$q_1$	272	0	$90^\circ$
2	$q_2$	0	253,52	0
3	$q_3$	0	253	0
4	$q_4$	0	0	$-90^\circ$
5	$q_5+180^\circ$	145,71	0	0

Fuente: Elaboración propia

Ya que la cinemática directa se realizó con una configuración del robot que no es la original, simplemente se debe corregir la postura y mantenerla para todos los efectos de cálculos posteriores, los ángulos que se deben corregir y que se tomarán en cuenta para posteriores cálculos son los siguientes:

- Al ángulo de  $q_2$  se debe sumar  $+90^\circ$ .
- Al ángulo de  $q_3$  se debe sumar  $-90^\circ$ .
- Al ángulo de  $q_4$  se debe sumar  $-90^\circ$ .

Sumando éstos ángulos, el Robot se mantendrá la posición original de HOME.

Las matrices de transformación básica  ${}^{i-1}A_i$  de cada articulación se calculan con la utilización directa de los parámetros de D-H en la matriz (4).

$${}^{i-1}A_i = \begin{bmatrix} c\theta_i & -c\alpha_i s\theta_i & s\alpha_i s\theta_i & a_i c\theta_i \\ s\theta_i & c\alpha_i c\theta_i & -s\alpha_i c\theta_i & a_i s\theta_i \\ 0 & s\alpha_i & c\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4)$$

Por lo tanto, las matrices de transformación básica de cada articulación quedan de la siguiente manera:

$${}^0A_1 = \begin{bmatrix} C_1 & 0 & S_1 & 0 \\ S_1 & 0 & -C_1 & 0 \\ 0 & 1 & 0 & 272 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (6)$$

$${}^1A_2 = \begin{bmatrix} C_2 & -S_2 & 0 & 253,52 C_2 \\ S_2 & C_2 & 0 & 253,52 S_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (7)$$



$${}^2A_3 = \begin{bmatrix} C_3 & -S_3 & 0 & 253 C_3 \\ S_3 & C_3 & 0 & 253 S_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (8)$$

$${}^3A_4 = \begin{bmatrix} C_4 & 0 & -S_4 & 0 \\ S_4 & 0 & C_4 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (9)$$

$${}^4A_5 = \begin{bmatrix} -C_5 & S_5 & 0 & 0 \\ -S_5 & -C_5 & 0 & 0 \\ 0 & 0 & 1 & 145,71 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (10)$$

Finalmente, para obtener la matriz  $T$  se realiza la multiplicación de cada una de las matrices anteriores y se obtiene la matriz de orientación y posición del extremo operativo con respecto a la base en función de las coordenadas articulares.

$$T = {}^0A_1 {}^1A_2 {}^2A_3 {}^3A_4 {}^4A_5 \quad (11)$$

Para realizar la validación de la cinemática directa, se hace uso de la librería de MATLAB desarrollada por (Corke, 2015), para esto, solo es necesario conocer los números de Denavit Hartenberg como parámetros de entrada y utilizar las siguientes funciones:

### Función (Link)

Un objeto Link contiene toda la información relacionada con un enlace de robot, tales como parámetros cinemáticos, parámetros de inercia de cuerpo rígido, parámetros de motor y transmisión.

Para el caso de estudio fue utilizado de la siguiente forma utilizando los números de DH.

$$L_n = \text{Link}([theta_n \quad d_n \quad a_n \quad alpha_n \quad sigma_n \quad offset_n])$$

Donde:

theta, d, a, alpha => Son los parámetros de DH.

sigma => Indicar "0" para articulación rotacional y "1" para articulación prismática.

offset => valor angular que acompaña a parámetro theta.

n => Articulación.

```
% Link([Theta, d , a , alpha , sigma , offset ])  
L1=Link([ 0 272 0 pi/2 0 0 ]);  
L2=Link([ 0 0 253.52 0 0 0 ]);  
L3=Link([ 0 0 253 0 0 0 ]);  
L4=Link([ 0 0 0 -pi/2 0 0 ]);  
L5=Link([ 0 145.71 0 0 0 pi ]);
```

### Función (SerialLink).

Con ésta función se hace la construcción del robot uniendo cada uno de los eslabones creados con la función "Link" en una estructura. Se utilizó de la siguiente manera:

```
bot=SerialLink([L1 L2 L3 L4 L5], 'name', 'Robot CRS');
```

Donde,

L1 L2 L3 L4 L5 => son los eslabones del robot.

name => se utiliza para darle nombre al robot.

### Función (n).

Es utilizada para realizar la verificación de los grados de libertad del robot que se está construyendo. Se utilizó de la siguiente manera:

```
bot.n
```

### Función (fkine).

La función fkine realiza el cálculo de la matriz  $T=[noap]$  dependiendo de los valores de entrada de cada una de los ángulos del robot en radianes. Se utilizó de la siguiente manera, dando como parámetros de entrada la posición home del robot real.

```
bot.fkine([0 pi/2 -pi/2 -pi/2 0])
```

### Función (plot).

La función plot dibuja en un entorno 3D el robot descrito por las funciones anteriores, los valores de entrada para este caso serían los ángulos del robot en posición home en radianes. Se utilizó la función de la siguiente manera.

```
bot.plot([0 pi/2 -pi/2 -pi/2 0]);
```

### Función (teach).

La función teach dibuja en un entorno 3D el robot descrito por las funciones anteriores y se pueden modificar los valores de cada ángulo directamente en el entorno tridimensional. Se utilizó la función de la siguiente manera.

```
bot.teach([0 pi/2 -pi/2 -pi/2 0]);
```

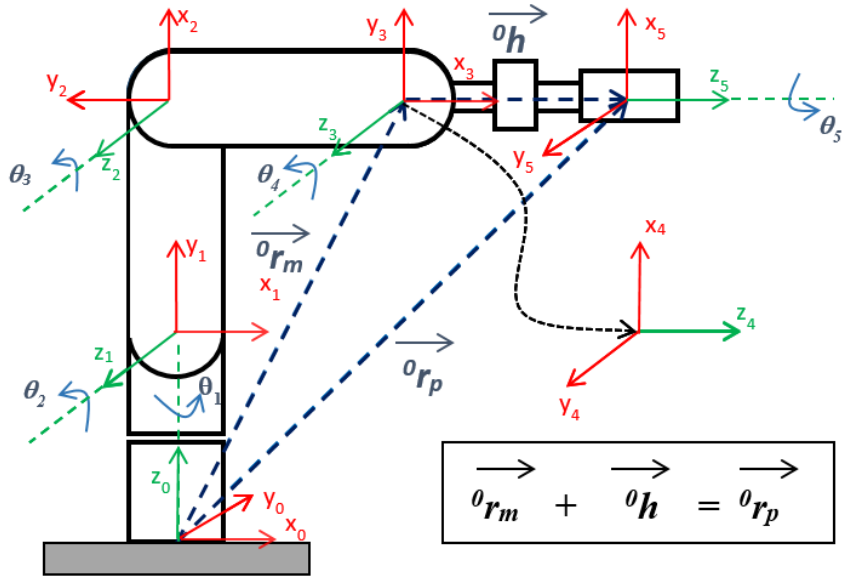
El código completo empleado para la verificación de la cinemática directa, utilizando las funciones antes descritas, se puede observar en Anexos.

## **5.3 Desarrollo de la cinemática inversa**

Para hacer la cinemática inversa se decide utilizar la metodología del desacoplo cinemático, en el cual se divide el robot en dos partes para luego acoplarlos vectorialmente.

En la Figura 9, se puede observar desacoplo realizado del robot manipulador de cinco grados de libertad.

Figura 9. Desacoplo cinemático del robot manipulador.



Fuente: Elaboración propia

Se calculan por separado las ecuaciones para obtener  $q_4$  y  $q_5$  y luego se obtiene  $q_1$ ,  $q_2$  y  $q_3$  mediante el método geométrico de la siguiente manera:

$$\Rightarrow {}^0R_5 = {}^0R_3 {}^3R_5 = [\text{noa}] \quad (12)$$

$$\Rightarrow {}^3R_5 = {}^3R_4 {}^4R_5 \quad (13)$$

(12) en (13)

$${}^0R_3 {}^3R_4 {}^4R_5 = [\text{noa}] \quad (14)$$

$${}^3R_4 {}^4R_5 = ({}^0R_3)^{-1} [\text{noa}] \quad (15)$$

$$\boxed{{}^3R_4 {}^4R_5 = ({}^0R_3)^T [\text{noa}]} \quad (16)$$

Luego partiendo de la ecuación del desacoplo cinemático,

$$\boxed{\vec{0}r_m + \vec{0}h = \vec{0}r_p}$$

(17)

Y conociendo que:

$$T = \begin{bmatrix} {}^0R_5 & {}^0r_p \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & p_x \\ r_{21} & r_{22} & r_{23} & p_y \\ r_{31} & r_{32} & r_{33} & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (18)$$

$$\begin{aligned} \vec{0}h = \vec{0}u \vec{0}a_{05} = {}^0R_5 \cdot \vec{0}h = {}^0R_5 \cdot d_5 \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} &= d_5 \cdot {}^0R_5 \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = d_5 \cdot \underbrace{\begin{bmatrix} r_{13} \\ r_{23} \\ r_{33} \end{bmatrix}}_{{}^0Z_5} \end{aligned} \quad (19)$$

$$\vec{0}r_m = {}^0r_5 - d_5 \begin{bmatrix} r_{13} \\ r_{23} \\ r_{33} \end{bmatrix} \quad (20)$$

### Ecuaciones Utilizadas en Cinemática inversa

Partiendo de la ecuación (16) y extrayendo  ${}^3R_4$ ,  ${}^4R_5$  y  ${}^0R_3$  de sus matrices correspondientes  ${}^3A_4$ ,  ${}^4A_5$  y  ${}^0A_3$  obtenemos  $q_4$  y  $q_5$  de la siguiente manera:

$${}^3A_4 = \begin{bmatrix} C_4 & 0 & -S_4 & 0 \\ S_4 & 0 & C_4 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (21)$$

$${}^3R_4 = \begin{bmatrix} C_4 & 0 & -S_4 \\ S_4 & 0 & C_4 \\ 0 & -1 & 0 \end{bmatrix} \quad (22)$$

$${}^4A_5 = \begin{bmatrix} -C_5 & S_5 & 0 & 0 \\ -S_5 & -S_5 & 0 & 0 \\ 0 & 0 & 1 & 145,71 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (23)$$

$${}^4R_5 = \begin{bmatrix} -C_5 & S_5 & 0 \\ -S_5 & -C_5 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (24)$$

$${}^0R_3 = \begin{bmatrix} C_1C_2C_3 - C_1S_2S_3 & -C_1C_2S_3 - C_1C_3S_2 & S_1 \\ C_2C_3S_1 - S_1S_2S_3 & -C_2S_1S_3 - C_3S_1S_2 & -C_1 \\ C_2S_3 + C_3S_2 & C_2C_3 - S_2S_3 & 0 \end{bmatrix} \quad (25)$$

$$({}^0R_3)^T = \begin{bmatrix} C_1C_2C_3 - C_1S_2S_3 & C_2C_3S_1 - S_1S_2S_3 & C_2S_3 + C_3S_2 \\ -C_1C_2S_3 - C_1C_3S_2 & -C_2S_1S_3 - C_3S_1S_2 & C_2C_3 - S_2S_3 \\ S_1 & -C_1 & 0 \end{bmatrix} \quad (26)$$

$${}^3R_4 \cdot {}^4R_5 = \begin{bmatrix} C_4 & 0 & -S_4 \\ S_4 & 0 & C_4 \\ 0 & -1 & 0 \end{bmatrix} \cdot \begin{bmatrix} -C_5 & S_5 & 0 \\ -S_5 & -C_5 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} -C_4C_5 & C_4S_5 & -S_4 \\ -C_5S_4 & S_4S_5 & C_4 \\ S_5 & C_5 & 0 \end{bmatrix} \quad (27)$$

$$\begin{aligned}
& ({}^0R_3)^T \cdot [noa] = \\
& \begin{bmatrix} C_1C_2C_3 - C_1S_2S_3 & C_2C_3S_1 - S_1S_2S_3 & C_2S_3 + C_3S_2 \\ -C_1C_2S_3 - C_1C_3S_2 & -C_2S_1S_3 - C_3S_1S_2 & C_2C_3 - S_2S_3 \\ S_1 & -C_1 & 0 \end{bmatrix} \cdot \begin{bmatrix} n_x & o_x & a_x \\ n_y & o_y & a_y \\ n_z & o_z & a_z \end{bmatrix} \quad (28)
\end{aligned}$$

$${}^3R_4 \cdot {}^4R_5 = ({}^0R_3)^T \cdot [noa] \quad (29)$$

$$\begin{aligned}
& \begin{bmatrix} -C_4C_5 & C_4S_5 & -S_4 \\ -C_5S_4 & S_4S_5 & C_4 \\ S_5 & C_5 & 0 \end{bmatrix} = \\
& \begin{bmatrix} C_1C_2C_3 - C_1S_2S_3 & C_2C_3S_1 - S_1S_2S_3 & C_2S_3 + C_3S_2 \\ -C_1C_2S_3 - C_1C_3S_2 & -C_2S_1S_3 - C_3S_1S_2 & C_2C_3 - S_2S_3 \\ S_1 & -C_1 & 0 \end{bmatrix} \cdot \begin{bmatrix} n_x & o_x & a_x \\ n_y & o_y & a_y \\ n_z & o_z & a_z \end{bmatrix} \quad (30)
\end{aligned}$$

Igualando términos de la ecuación (30):

$$-S_4 = (C_1C_2C_3 - C_1S_2S_3) \cdot a_x + (C_2C_3S_1 - S_1S_2S_3) \cdot a_y + (C_2S_3 + C_3S_2) \cdot a_z \quad (31)$$

$$-C_4 = (C_1C_2S_3 + C_1C_3S_2) \cdot a_x + (C_2S_1S_3 + C_3S_1S_2) \cdot a_y + (S_2S_3 - C_2C_3) \cdot a_z \quad (32)$$

Dividiendo (31) entre (32) se obtiene  $q_4$ .

$$q_4 = \text{atan} \left( \frac{(C_1C_2C_3 - C_1S_2S_3) \cdot a_x + (C_2C_3S_1 - S_1S_2S_3) \cdot a_y + (C_2S_3 + C_3S_2) \cdot a_z}{(C_1C_2S_3 + C_1C_3S_2) \cdot a_x + (C_2S_1S_3 + C_3S_1S_2) \cdot a_y + (S_2S_3 - C_2C_3) \cdot a_z} \right) \quad (33)$$

Igualando términos de la ecuación (30):

$$S_5 = S_1n_X - C_1n_Y \quad (34)$$

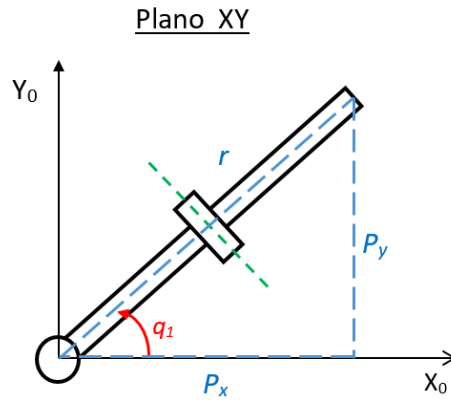
$$C_5 = S_1o_X - C_1o_Y \quad (35)$$

Dividiendo (34) entre (35) se obtiene  $q_5$ .

$$q_5 = \text{atan} \left( \frac{S_1n_X - C_1n_Y}{S_1o_X - C_1o_Y} \right) \quad (36)$$

Para el desarrollo de la cinemática inversa se utiliza el método geométrico de la siguiente manera:

**Figura 10** Plano XY del robot para cálculo de  $q_1$



Fuente: Elaboración propia

Utilizando la figura 10, se desarrollan las ecuaciones para el cálculo de  $q_1$ .

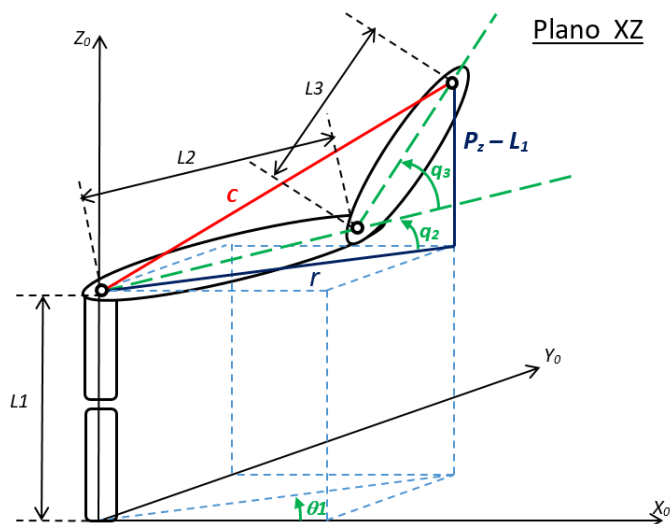
$$\tan q_1 = \frac{p_x}{p_y} \quad (37)$$

$$q_1 = \text{atan}\left(\frac{p_y}{p_x}\right) \quad (38)$$

$$r^2 = p_x^2 + p_y^2 \quad (39)$$

$$r = \pm \sqrt{p_x^2 + p_y^2} \quad (40)$$

**Figura 11** Plano XZ del robot para cálculo de  $q_3$



Fuente: Elaboración propia

Partiendo de la figura 11. Se desarrollan las ecuaciones para el cálculo de  $q_3$  de la siguiente manera:

Conociendo que:

$$L_1 = 272 \text{ mm}$$

$$L_2 = 253,52 \text{ mm}$$

$$L_3 = 253 \text{ mm}$$

$$\cos(\pi - \alpha) = -\cos(\alpha) \quad (41)$$

$$\sin^2 \alpha + \cos^2 \alpha = 1 \quad (42)$$

$$\text{sen}(\alpha) = \pm \sqrt{1 - \cos^2 \alpha} \quad (43)$$

Aplicando trigonometría se obtiene.

$$c^2 = L_2^2 + L_3^2 - 2L_2L_3\cos(\pi + q_3) \quad (44)$$

$$r^2 + (p_z - L_1)^2 = L_2^2 + L_3^2 + 2L_2L_3\cos(q_3) \quad (45)$$

$$\cos(q_3) = \frac{p_x^2 + p_y^2 + (p_z - L_1)^2 - L_2^2 - L_3^2}{2L_2L_3} \quad (46)$$

Teniendo en cuenta que:

$$\text{sen}(q_3) = \pm \sqrt{1 - \cos^2(q_3)} \quad (47)$$

Dividiendo toda la ecuación entre  $\cos(q_3)$ .

$$\tan(q_3) = \frac{\pm \sqrt{1 - \cos^2(q_3)}}{\cos(q_3)} \quad (48)$$

Por lo tanto:

$$q_3 = \text{atan} \left( \frac{\pm \sqrt{1 - \cos^2(q_3)}}{\cos(q_3)} \right) \quad (49)$$

Utilizando para éste caso el signo (-) en la ecuación (49).

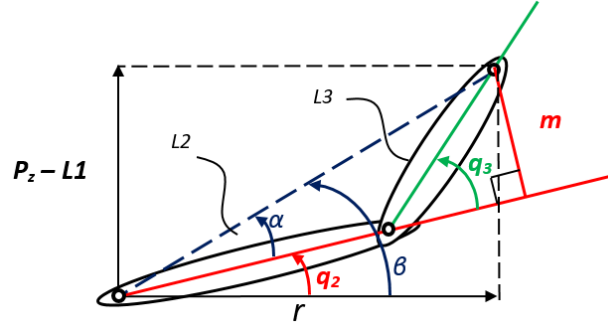
Y donde:

$$\cos(q_3) = \frac{p_x^2 + p_y^2 + (p_z - L_1)^2 - L_2^2 - L_3^2}{2L_2L_3} \quad (50)$$



Para el cálculo de  $q_2$  se utiliza el robot en la posición de la figura 11, el cual se encuentra dentro del plano generado por  $(r)$  y  $(P_z - L_1)$ , que son definidos en las ecuaciones anteriores y se observa en la figura 12.

**Figura 12** Posición del robot para el cálculo de  $q_2$



Fuente: Elaboración propia

$$\text{sen}(q_3) = \frac{m}{L_3} \quad (51)$$

$$m = L_3 \text{sen}(q_3) \quad (52)$$

$$\tan(\alpha) = \frac{m}{L_2 + L_3 \cos(q_3)} \quad (53)$$

$$\alpha = \text{atan}\left(\frac{L_3 \text{sen}(q_3)}{L_2 + L_3 \cos(q_3)}\right) \quad (54)$$

$$q_2 = \beta - \alpha \quad (55)$$

$$\tan(\beta) = \frac{p_z - L_1}{r} = \frac{p_z - L_1}{\pm\sqrt{p_x^2 + p_y^2}} \quad (56)$$

$$\beta = \text{atan}\left(\frac{p_z - L_1}{\pm\sqrt{p_x^2 + p_y^2}}\right) \quad (57)$$

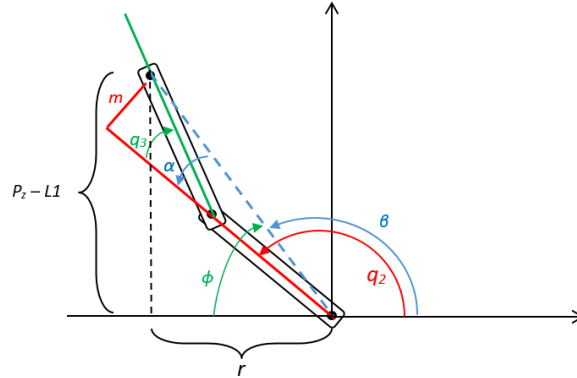
$$q_2 = \text{atan}\left(\frac{p_z - L_1}{\pm\sqrt{p_x^2 + p_y^2}}\right) - \text{atan}\left(\frac{L_3 \text{sen}(q_3)}{L_2 + L_3 \cos(q_3)}\right) \quad (58)$$

Para el caso de estudio el cálculo de  $q_2$  de la ecuación (58) se utiliza el signo (+) de la raíz.

Finalmente, la última ecuación desarrollada se realiza en base a una posición especial del robot, el cual adopta una forma particular, es decir para

cuando el ángulo  $q_2$  es mayor a cero. En la figura 13. Se observa la posición del robot para este caso.

**Figura 13** Posición especial del robot



*Fuente:* Elaboración propia

$$q_2 = \beta + \alpha \quad (59)$$

$$\varphi = 180^\circ - \beta \quad (60)$$

$$\tan(\varphi) = \frac{p_z - L_1}{r} \quad (61)$$

$$\tan(180^\circ - \beta) = \frac{p_z - L_1}{r} \quad (62)$$

$$-\tan(\beta) = \frac{p_z - L_1}{r} \quad (63)$$

$$\tan(\beta) = \frac{L_1 - p_z}{r} \quad (64)$$

$$\beta = \text{atan}\left(\frac{L_1 - p_z}{\pm\sqrt{p_x^2 + p_y^2}}\right) \quad (65)$$

$$\text{sen}(q_3) = \frac{m}{L_3} \quad (66)$$

Debido a que la rotación de  $q_3$  en el robot real es en sentido negativo, utilizamos únicamente para esta configuración del robot el signo negativo de  $q_3$  para que la distancia ( $m$ ) de un valor positivo.

$$m = L_3 \text{sen}(-q_3) \quad (67)$$

$$\tan(\alpha) = \frac{m}{L_2 + L_3 \cos(-q_3)} \quad (68)$$

$$\tan(\alpha) = \frac{L_3 \text{sen}(-q_3)}{L_2 + L_3 \cos(-q_3)} \quad (69)$$

$$\alpha = \text{atan}\left(\frac{L_3 \text{sen}(-q_3)}{L_2 + L_3 \text{cos}(-q_3)}\right) \quad (70)$$

$$q_2 = \text{atan}\left(\frac{L_1 - p_z}{\pm\sqrt{p_x^2 + p_y^2}}\right) + \text{atan}\left(\frac{L_3 \text{sen}(-q_3)}{L_2 + L_3 \text{cos}(-q_3)}\right) \quad (71)$$

Sabiendo que:

$$\text{sen}(-\delta) = -\text{sen}(\delta) \quad (72)$$

$$\text{cos}(-\delta) = \text{cos}(\delta) \quad (73)$$

Se cambia ( $\alpha$ ) a signo negativo en la ecuación quedando de la siguiente forma:

$$q_2 = \text{atan}\left(\frac{L_1 - p_z}{\pm\sqrt{p_x^2 + p_y^2}}\right) - \text{atan}\left(\frac{L_3 \text{sen}(q_3)}{L_2 + L_3 \text{cos}(q_3)}\right) \quad (74)$$

Asumiendo que dicha ecuación es únicamente para el caso dibujado, se utilizará el signo (+) de la raíz para la ecuación (74).

Finalmente, cada una de estas ecuaciones fue evaluada según los siguientes criterios:

- Signos de las raíces y la posición de codo arriba o codo abajo que trabaja el robot real.
- Modificaciones de las ecuaciones dependiendo del cuadrante donde se quiera llevar el robot. Para algunas posiciones, a las ecuaciones se les debe agregar el valor de  $+\pi$  a la ecuación y para otras posiciones se les debe agregar el valor de  $-\pi$ .

Por lo tanto, las ecuaciones utilizadas para  $q_1$ ,  $q_2$ ,  $q_3$ ,  $q_4$ , y  $q_5$  son las siguientes:

- Para el cálculo de  $q_1$  se tienen tres opciones de la misma ecuación, la original (38), la segunda sumando  $\pi$  y una tercera restando  $\pi$ .

$$q_1 = \text{atan}\left(\frac{p_y}{p_x}\right) \quad (38)$$

$$q_1 = \text{atan}\left(\frac{p_y}{p_x}\right) + \pi \quad (75)$$

$$q_1 = \text{atan}\left(\frac{p_y}{p_x}\right) - \pi \quad (76)$$

- Para el cálculo de  $q_3$  se tienen dos opciones de la misma ecuación, la original (49) utilizando el símbolo negativo de la raíz (-) y una segunda ecuación restando  $\pi$ .

$$q_3 = \operatorname{atan}\left(\frac{-\sqrt{1 - \cos^2(q_3)}}{\cos(q_3)}\right) \quad (49)$$

$$q_3 = \operatorname{atan}\left(\frac{-\sqrt{1 - \cos^2(q_3)}}{\cos(q_3)}\right) - \pi \quad (77)$$

- Para el cálculo de  $q_2$  se tienen dos ecuaciones, la original (58) con posición del robot con codo arriba utilizando el signo positivo (+) de la raíz y una segunda ecuación (74), calculada en posición de cobra con el signo positivo (+) de la raíz y el valor de  $\pi$  a la ecuación.

$$q_2 = \operatorname{atan}\left(\frac{p_z - L_1}{+\sqrt{p_x^2 + p_y^2}}\right) - \operatorname{atan}\left(\frac{L_3 \operatorname{sen}(q_3)}{L_2 + L_3 \cos(q_3)}\right) \quad (58)$$

$$q_2 = \operatorname{atan}\left(\frac{L_1 - p_z}{+\sqrt{p_x^2 + p_y^2}}\right) - \operatorname{atan}\left(\frac{L_3 \operatorname{sen}(q_3)}{L_2 + L_3 \cos(q_3)}\right) + \pi \quad (74)$$

- Para el cálculo de  $q_4$  se tienen dos opciones de la misma ecuación, la original (33) y una segunda ecuación restando  $\pi$ .

$$q_4 = \operatorname{atan}\left(\frac{(C_1 C_2 C_3 - C_1 S_2 S_3) \cdot a_x + (C_2 C_3 S_1 - S_1 S_2 S_3) \cdot a_y + (C_2 S_3 + C_3 S_2) \cdot a_z}{(C_1 C_2 S_3 + C_1 C_3 S_2) \cdot a_x + (C_2 S_1 S_3 + C_3 S_1 S_2) \cdot a_y + (S_2 S_3 - C_2 C_3) \cdot a_z}\right) \quad (33)$$

$$q_4 = \operatorname{atan}\left(\frac{(C_1 C_2 C_3 - C_1 S_2 S_3) \cdot a_x + (C_2 C_3 S_1 - S_1 S_2 S_3) \cdot a_y + (C_2 S_3 + C_3 S_2) \cdot a_z}{(C_1 C_2 S_3 + C_1 C_3 S_2) \cdot a_x + (C_2 S_1 S_3 + C_3 S_1 S_2) \cdot a_y + (S_2 S_3 - C_2 C_3) \cdot a_z}\right) - \pi \quad (78)$$

- Para el cálculo de  $q_5$  se tienen tres opciones de la misma ecuación, la original (36), una segunda ecuación sumando  $\pi$  y una tercera ecuación restando  $\pi$ .

$$q_5 = \operatorname{atan}\left(\frac{S_1 n_X - C_1 n_Y}{S_1 o_X - C_1 o_Y}\right) \quad (36)$$

$$q_5 = \operatorname{atan}\left(\frac{S_1 n_X - C_1 n_Y}{S_1 o_X - C_1 o_Y}\right) + \pi \quad (79)$$

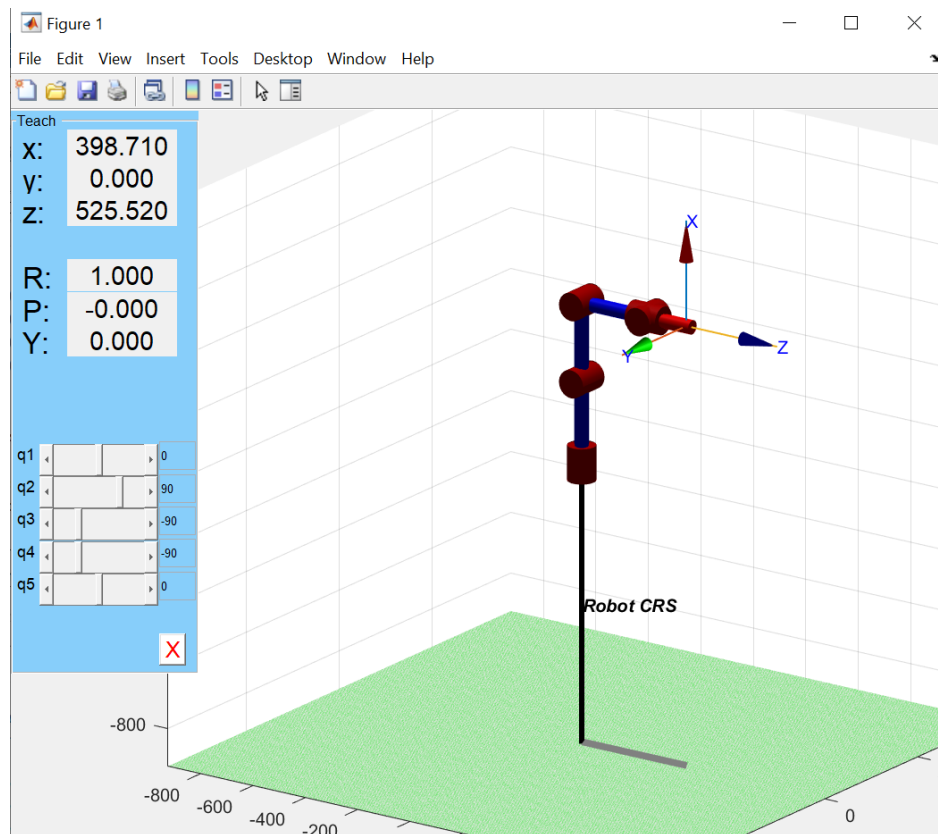
$$q_5 = \operatorname{atan}\left(\frac{S_1 n_X - C_1 n_Y}{S_1 o_X - C_1 o_Y}\right) - \pi \quad (80)$$

## 6. Resultados y discusión

### Validación en MATLAB de Cinemática directa

En la Figura 14 se puede observar el resultado de la simulación en MATLAB del problema cinemático directo, podemos notar como el manipulador está representando exactamente al robot original en su posición HOME, también se observa como los sistemas de coordenadas coinciden con los utilizados en el cálculo de la cinemática directa y al utilizar el panel “teach” del entorno 3D, el cual se puede modificar y jugar con los ángulos, la simulación hace los mismos movimientos que el robot original. De ésta forma queda validada la cinemática directa.

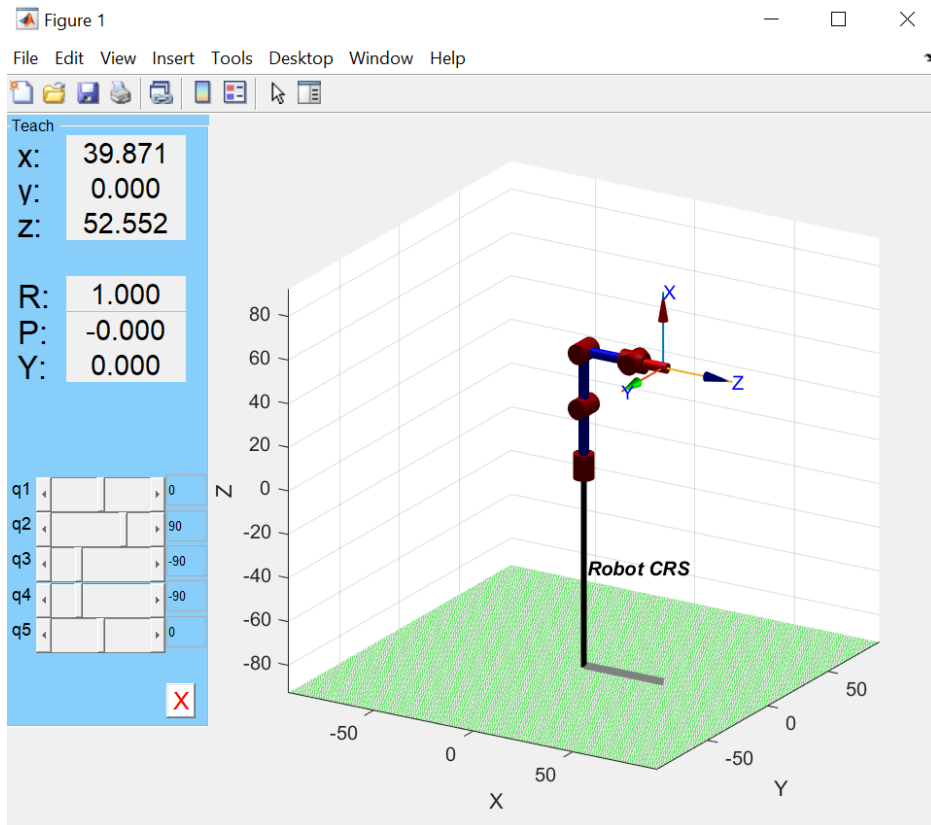
**Figura 14** Simulación de la cinemática directa



Luego de realizar la simulación de la cinemática directa con los valores de los parámetros en milímetros, como se desarrolló inicialmente, se pudo notar cierta lentitud en el desarrollo de programa, debido a que estaba procesando valores numéricos altos en los parámetros DH involucrados, por lo cual se ajustaron los parámetros únicamente para agilizar los cálculos por parte de la

librería robótica utilizada, los valores rotacionales  $\theta$  y  $\alpha$  no cambian, ya que representan ángulos, pero las longitudes del robot fueron convertidas de milímetros a centímetros. Por lo tanto, en la Figura 15, se puede observar los cambios de conversión realizados con respecto a las dimensiones longitudinales.

**Figura 15** Validación ajustada de la cinemática directa



Posibles combinaciones para cálculo de cinemática inversa.

Como se observó en el apartado del desarrollo de la cinemática inversa, se determinaron doce ecuaciones para realizar el cálculo de la cinemática inversa. Donde  $q_1$  tiene tres opciones,  $q_2$  tiene dos opciones,  $q_3$  tiene dos opciones,  $q_4$  tiene dos opciones y  $q_5$  tiene tres opciones, de esta manera se generan setenta y dos posibles combinaciones entre dichas ecuaciones para resolver el problema, el cual agrega al cálculo una dificultad mayor para la obtención de los ángulos  $q_1$ ,  $q_2$ ,  $q_3$ ,  $q_4$  y  $q_5$ .

La resolución del problema se solventó mediante programación de bucles, para ir recorriendo cada una de las combinaciones hasta conseguir la solución. Para generar los índices de dichos bucles se tomó en cuenta la Tabla 3, que muestra cada una de las combinaciones posibles para el cálculo de la cinemática inversa.

La tabla representa la opción de la ecuación que se debe utilizar cada combinación, por ejemplo, para la combinación 1, se utiliza la ecuación número 1 del cálculo de  $q_1$   $q_2$   $q_3$   $q_4$  y  $q_5$ . Otro ejemplo, para la combinación 6, se utiliza la opción 1 para el cálculo de  $q_1$ , la opción 1 para el cálculo de  $q_3$ , la opción 1 para el cálculo de  $q_2$ , la opción 3 para el cálculo de  $q_5$  y la opción 2 para el cálculo de  $q_4$ .

**Tabla 3** *Combinación de ecuaciones de cinemática inversa*

Combinación	Ecuaciones utilizadas para cada combinación				
	$q_1$	$q_3$	$q_2$	$q_5$	$q_4$
	3 opciones	2 opciones	2 opciones	3 opciones	2 opciones
1	1	1	1	1	1
2	1	1	1	1	2
3	1	1	1	2	1
4	1	1	1	2	2
5	1	1	1	3	1
6	1	1	1	3	2
7	1	1	2	1	1
8	1	1	2	1	2
9	1	1	2	2	1
10	1	1	2	2	2
11	1	1	2	3	1
12	1	1	2	3	2
13	1	2	1	1	1
14	1	2	1	1	2
15	1	2	1	2	1
16	1	2	1	2	2
17	1	2	1	3	1
18	1	2	1	3	2
19	1	2	2	1	1
20	1	2	2	1	2
21	1	2	2	2	1
22	1	2	2	2	2
23	1	2	2	3	1
24	1	2	2	3	2
25	2	1	1	1	1
26	2	1	1	1	2
27	2	1	1	2	1
28	2	1	1	2	2
29	2	1	1	3	1
30	2	1	1	3	2
31	2	1	2	1	1
32	2	1	2	1	2
33	2	1	2	2	1

34	2	1	2	2	2
35	2	1	2	3	1
36	2	1	2	3	2
37	2	2	1	1	1
38	2	2	1	1	2
39	2	2	1	2	1
40	2	2	1	2	2
41	2	2	1	3	1
42	2	2	1	3	2
43	2	2	2	1	1
44	2	2	2	1	2
45	2	2	2	2	1
46	2	2	2	2	2
47	2	2	2	3	1
48	2	2	2	3	2
49	3	1	1	1	1
50	3	1	1	1	2
51	3	1	1	2	1
52	3	1	1	2	2
53	3	1	1	3	1
54	3	1	1	3	2
55	3	1	2	1	1
56	3	1	2	1	2
57	3	1	2	2	1
58	3	1	2	2	2
59	3	1	2	3	1
60	3	1	2	3	2
61	3	2	1	1	1
62	3	2	1	1	2
63	3	2	1	2	1
64	3	2	1	2	2
65	3	2	1	3	1
66	3	2	1	3	2
67	3	2	2	1	1
68	3	2	2	1	2
69	3	2	2	2	1
70	3	2	2	2	2
71	3	2	2	3	1
72	3	2	2	3	2

Fuente: Elaboración propia

Para la combinación 40, por ejemplo, se utiliza la opción 2 para el cálculo de  $q_1$ , la opción 2 para el cálculo de  $q_3$ , la opción 1 para el cálculo de  $q_2$ , la opción 2 para el cálculo de  $q_5$  y la opción 2 para el cálculo de  $q_4$ .



### Validación en MATLAB de Cinemática Inversa a partir de matriz $T=[noap]$

Para hacer la validación de la cinemática inversa en MATLAB, lo primero que se hizo fue calcular una matriz  $T=[noap]$  de ejemplo con la cinemática directa, colocando valores aleatorios a los ángulos  $q_1, q_2, q_3, q_4$  y  $q_5$ , de manera de obtener la matriz de rotación y la matriz de posición del efector final del robot. A partir de los valores de matriz  $T$  de ejemplo, se va recorriendo cada una de las 72 combinaciones posibles para el cálculo de los ángulos. Al ser calculado todos los ángulos con la combinación ( $n$ ) de cinemática Inversa, se utiliza la cinemática directa para calcular de nuevo una matriz  $T$ , pero con los valores obtenidos en la cinemática inversa. Finalmente, al tener una matriz  $T$  de ejemplo y una matriz  $T$  calculada con los valores de ángulos devueltos por la cinemática inversa, se hace una comparación, si ambas matrices son iguales, el bucle se detiene indicando que la cinemática inversa está correctamente calculada, si ambos matrices no son iguales, el bucle se repite  $n$  veces hasta que la matriz  $T$  de ejemplo y la matriz  $T$  calculada por la cinemática inversa sean iguales. En el caso de que el bucle pase por las 72 posibles combinaciones del cálculo de la cinemática inversa y no se obtenga solución alguna, el bucle se detiene sin encontrar la solución para dicha configuración del robot.

Los ángulos escogidos aleatoriamente para  $q_1, q_2, q_3, q_4$  y  $q_5$ , para trabajar en el algoritmo desarrollado en MATLAB, son:

```
q1= -pi/2;           % desde -180 hasta +180
q2= 7*pi/18;        % desde -90 hasta +10
                    % la suma de q2 y q3 no puede mayor que 79
q3= -4*pi/9         % desde -35 hasta +90
q4= -pi/3;         % desde -110 hasta +110
q5= pi/3;          % desde -180 hasta 180
```

Inicia con el cálculo de la matriz de translación  $T$  de ejemplo usando los valores aleatorios anteriormente escogidos.

```
% Asignación de las Variables de las matrices de Transformación
Homogénea
% 0A1= A1, 1A2=A2, 2A3=A3, 3A4=A4, 4A5=A5,
syms A1 A2 A3 A4 A5;
A1=[cos(q1) 0 sin(q1) 0 ; sin(q1) 0 -cos(q1) 0 ; 0 1 0 272;0 0 0 1];
A2=[cos(q2) -sin(q2) 0 253.52*cos(q2); sin(q2) cos(q2) 0
253.52*sin(q2); 0 0 1 0; 0 0 0 1];
A3=[cos(q3) -sin(q3) 0 253*cos(q3);sin(q3) cos(q3) 0 253*sin(q3); 0 0
1 0;0 0 0 1];
A4=[cos(q4) 0 -sin(q4) 0;sin(q4) 0 cos(q4) 0; 0 -1 0 0; 0 0 0 1];
```

```
A5=[-cos(q5) sin(q5) 0 0;-sin(q5) -cos(q5) 0 0;0 0 1 145.71; 0 0 0 1];
```

```
%Cálculo de Matriz T
T5=A1*A2*A3*A4*A5
```

Posteriormente se realiza la verificación de la cinemática inversa como se lo indico anteriormente, realizando el cálculo y comparación respectivo de matrices.

```
syms Q1_1 Q1_2 Q1_3;
syms Q2_1 Q2_2;
syms Q3_1 Q3_2;
syms Q4_1 Q4_2;
syms Q5_1 Q5_2 Q5_3
syms QQ1 QQ2 QQ3;

syms Q1 Q2 Q3 Q4 Q5;
syms px py pz l1 l2 l3 ax ay az nx ny nz ox oy oz;
syms CosQ3;
syms x i1 i2 i3 i4 i5 i6 w m n o p r s t;
x=1; i1=1; i2=1; i3=1; i5=1; w=1; m=0; n=0; o=0; p=0; r=0; s=0; t=0;

nx=T5(1); ox=T5(5); ax=T5(9);
ny=T5(2); oy=T5(6); ay=T5(10);
nz=T5(3); oz=T5(7); az=T5(11);

r5=[T5(13) T5(14) T5(15)]; % estos serían Px Py y Pz a colocar para
calcular la inversa.
z5=[T5(9) T5(10) T5(11)]; % estos serían r13 r23 y r33 px py pz para
calcular la inversa.
rm=r5-145.71*z5; % ecuación derivada del desacoplo cinemático
px=rm(1); % esto es px del punto M es decir, del eje coord 3
py=rm(2); % esto es py del punto M es decir, del eje coord 3
pz=rm(3); % esto es pz del punto M es decir, del eje coord 3

l1=272; l2=253.52; l3=253;

% Ecuaciones para calcular Q1 se tienen 3 opciones
if T5(13)<0&&T5(14)>=0
    Q1_1=atan(py/px)+pi;
    Q1_2=atan(py/px)-pi;
    Q1_3=atan(py/px);
end
if T5(13)<0&&T5(14)<0
    Q1_1=atan(py/px)-pi;
    Q1_2=atan(py/px)+pi;
    Q1_3=atan(py/px);
end
if T5(13)>=0
    Q1_1=atan(py/px);
    Q1_2=atan(py/px)+pi;
    Q1_3=atan(py/px)-pi;
end
if T5(13)<0&&T5(5)<0
    Q1_1=atan(py/px);
    Q1_2=atan(py/px)+pi;
    Q1_3=atan(py/px)-pi;
```

```

end

%Ecuaciones para calcular Q3
CosQ3=((px^2)+(py^2)+((pz-11)^2)-(12^2)-(13^2))/(2*12*13);

Q3_1=atan((-sqrt(1-((CosQ3)^2)))/(CosQ3));
Q3_2=atan((-sqrt(1-((CosQ3)^2)))/(CosQ3))-pi;

while x == 1
    %índice para combinacion de q4
    if rem(w,2)==0
        i4=2;
    else
        i4=1;
    end
    % índice para combinacion de q5
    if w == 3 || w == m
        m = w + 2;
        i5=i5+1;
    end
    if (w == 7 || w == n)
        n = w + 6;
        i5 = 1;
    end
    % índice para combinacion de q2
    if (w == 7 || w == o)
        o = w + 12;
        i2 = 2;
    end
    if (w == 13 || w == p)
        p = w + 12;
        i2 = 1;
    end
    % índice para combinacion de q3
    if(w == 13 || w == r)
        r = w + 24;
        i3 = 2;
    end
    if (w == 25 || w == s)
        s = w + 24;
        i3 = 1;
    end
    % índice para combinacion de q1
    if (w == 25 || w == t)
        t = w + 24;
        i1=i1+1;
    end

    %Ecuaciones para calcular Q2
    if i3==1
        QQ3=Q3_1;
    end
    if i3==2
        QQ3=Q3_2;
    end

    Q2_1=atan((pz-11)/(sqrt((px^2)+(py^2))))-
    atan((13*sin(QQ3))/(12+(13*cos(QQ3))));
    Q2_2=pi+atan((11-pz)/(sqrt((px^2)+(py^2))))-
    atan((13*sin(QQ3))/(12+(13*cos(QQ3))));

```

```

%Ecuaciones Para calcular Q5
if i1==1
    QQ1=Q1_1;
end
if i1==2
    QQ1=Q1_2;
end
if i1==3
    QQ1=Q1_3;
end
Q5_1=atan(((sin(QQ1)*nx)-(cos(QQ1)*ny))/((sin(QQ1)*ox)-(cos(QQ1)*oy)));
Q5_2=atan(((sin(QQ1)*nx)-(cos(QQ1)*ny))/((sin(QQ1)*ox)-(cos(QQ1)*oy)))-pi;
Q5_3=atan(((sin(QQ1)*nx)-(cos(QQ1)*ny))/((sin(QQ1)*ox)-(cos(QQ1)*oy)))+pi;

%Ecuaciones Para Calcular Q4
if i2==1
    QQ2=Q2_1;
end
if i2==2
    QQ2=Q2_2;
end
v1=((cos(QQ1)*cos(QQ2)*cos(QQ3))-(cos(QQ1)*sin(QQ2)*sin(QQ3)))*ax;
v2=((cos(QQ2)*cos(QQ3)*sin(QQ1))-(sin(QQ1)*sin(QQ2)*sin(QQ3)))*ay;
v3=((cos(QQ2)*sin(QQ3))+(cos(QQ3)*sin(QQ2)))*az;

v4=((cos(QQ1)*cos(QQ2)*sin(QQ3))+(cos(QQ1)*cos(QQ3)*sin(QQ2)))*ax;
v5=((cos(QQ2)*sin(QQ1)*sin(QQ3))+(cos(QQ3)*sin(QQ1)*sin(QQ2)))*ay;
v6=((sin(QQ2)*sin(QQ3))-(cos(QQ2)*cos(QQ3)))*az;

Q4_1=atan((v1+v2+v3)/(v4+v5+v6));
Q4_2=atan((v1+v2+v3)/(v4+v5+v6))-pi;

% Valores de q1, q2, q3, q4 y q5 dependiendo de las combinaciones
y los índices i1, i2, i3, i4, i5
if i1 == 1
    Q1 = Q1_1;
end
if i1 == 2
    Q1 = Q1_2;
end
if i1 == 3
    Q1 = Q1_3;
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
if (i2 == 1)
    Q2 = Q2_1;
end
if (i2 == 2)
    Q2 = Q2_2;
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
if (i3 == 1)
    Q3 = Q3_1;
end
if (i3 == 2)
    Q3 = Q3_2;
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```

if (i4 == 1)
    Q4 = Q4_1;
end
if (i4 == 2)
    Q4 = Q4_2;
end
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
if (i5 == 1)
    Q5 = Q5_1;
end
if (i5 == 2)
    Q5 = Q5_2;
end
if (i5 == 3)
    Q5 = Q5_3;
end
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
w=w+1;

B1=[cos(Q1) 0 sin(Q1) 0 ; sin(Q1) 0 -cos(Q1) 0 ; 0 1 0 272;0 0 0 1];
B2=[cos(Q2) -sin(Q2) 0 253.52*cos(Q2); sin(Q2) cos(Q2) 0
253.52*sin(Q2); 0 0 1 0; 0 0 0 1];
B3=[cos(Q3) -sin(Q3) 0 253*cos(Q3);sin(Q3) cos(Q3) 0 253*sin(Q3); 0 0
1 0;0 0 0 1];
B4=[cos(Q4) 0 -sin(Q4) 0;sin(Q4) 0 cos(Q4) 0; 0 -1 0 0; 0 0 0 1];
B5=[-cos(Q5) sin(Q5) 0 0;-sin(Q5) -cos(Q5) 0 0;0 0 1 145.71; 0 0 0 1];
R5=B1*B2*B3*B4*B5;
syms limit1; limit1=Q2+Q3; % restricción: la suma de los ángulos Q2 y
Q3 no debe superar 1,38 rad
if (1.570<= Q2 && Q2 <= 1.571)
    limit1=0;
end
if (-1.570<= Q3 && Q3 <= -1.571 && limit1==1)
    limit1=0;
end
end

% estos serían Px, Py y Pz a colocar para calcular la inversa.
% estos serian r13, r23 y r33 px py pz para calcular la inversa.
if (R5(13)-0.001<=T5(13) && T5(13)<= R5(13)+0.001 &&...
R5(14)-0.001<=T5(14) && T5(14)<= R5(14)+0.001 &&...
R5(15)-0.001<=T5(15) && T5(15)<= R5(15)+0.001 &&...
R5(9) -0.001<=T5(9) && T5(9) <= R5(9) +0.001 &&...
R5(10)-0.001<=T5(10) && T5(10)<= R5(10)+0.001 &&...
R5(11)-0.001<=T5(11) && T5(11)<= R5(11)+0.001 &&...
R5(5) -0.001<=T5(5) && T5(5) <= R5(5) +0.001 &&...
R5(6) -0.001<=T5(6) && T5(6) <= R5(6)+0.001 &&...
R5(7) -0.001<=T5(7) && T5(7) <= R5(7)+0.001 &&...
Q1 >= -3.142 && Q1<=3.142&&... % Q1 >= -3.1416 &&
Q1<=3.1416&&...
Q2 >= 0 && Q2<=1.76&&... % Q2 >= 0 &&
Q2<=1.7453&&...
Q3 >= -2.182 && Q3<=0 &&...% Q3 >= -2.1817 && Q3<=0
&&...
Q4 >= -3.491 && Q4<=0.350&&... % Q4 >= -3.4907 &&
Q4<=0.3491&&...
Q5 >= -3.142 && Q5<=3.142&&... % Q5 >= -3.1416 &&
Q5<=3.1416&&...
limit1<=1.38); % limit1 <= 1.3789

fprintf('Inversa hallada correctamente, Iteraciones %d\n',w);

```

```

        x=0;
end
    if w==73;
        disp('Inversa No hallada');
        x=0;
    end
end

B1=[cos(Q1) 0 sin(Q1) 0 ; sin(Q1) 0 -cos(Q1) 0 ; 0 1 0 272;0 0 0 1];
B2=[cos(Q2) -sin(Q2) 0 253.52*cos(Q2); sin(Q2) cos(Q2) 0
253.52*sin(Q2); 0 0 1 0; 0 0 0 1];
B3=[cos(Q3) -sin(Q3) 0 253*cos(Q3);sin(Q3) cos(Q3) 0 253*sin(Q3); 0 0
1 0;0 0 0 1];
B4=[cos(Q4) 0 -sin(Q4) 0;sin(Q4) 0 cos(Q4) 0; 0 -1 0 0; 0 0 0 1];
B5=[-cos(Q5) sin(Q5) 0 0;-sin(Q5) -cos(Q5) 0 0;0 0 1 145.71; 0 0 0 1];

fprintf('Matriz R5 con ángulos calculados por Inversa ');
R5=B1*B2*B3*B4*B5

```

El código completo de la validación de la cinemática inversa desarrollada, se puede observar en el Anexos.

Los resultados obtenidos por el algoritmo planteado presentan la matriz de translación  $T$  de ejemplo y la matriz  $R$  calculados con los ángulos calculados por la cinemática inversa, ver Figura 16.

**Figura 16** Resultados de la Validación de la cinemática inversa.

```

Command Window
>> Verificacion_de_Inversa

T5 =

    -0.8660    -0.5000     0.0000     0.0000
     0.1710    -0.2962    -0.9397   -472.7879
     0.4698    -0.8138     0.3420    516.1336
         0         0         0         1.0000

Inversa hallada correctamente, Iteraciones 2
Matriz R5 con ángulos calculados por Inversa
R5 =

    -0.8660    -0.5000     0.0000     0.0000
     0.1710    -0.2962    -0.9397   -472.7879
     0.4698    -0.8138     0.3420    516.1336
         0         0         0         1.0000

```

## 7. Conclusiones y recomendaciones

### Conclusiones.

El modelamiento matemático de la cadena cinemática de un robot manipulador es un proceso muy tedioso, por lo tanto, para que el producto final sea lo más cercano a la realidad, se debe desarrollar de forma detallada y hacer uso de software especializado.

Se pudo obtener el modelo cinemático directo del manipulador de cinco grados de libertad, utilizando el método de Denavit – Hartenberg, y se validó el funcionamiento por medio de la simulación en MATLAB. Dentro de la simulación, se propuso realizar una conversión a centímetros de los parámetros DH con respecto a los valores de traslación, debido a que esto, agiliza los movimientos del robot cuando el código está operando en tiempo real. Hay que considerar que el programa si funciona con los parámetros originales que se encuentran en milímetros, pero la simulación se torna un poco lenta, debido a que los valores son de mayor magnitud y el software demora en procesar la información.

La simulación de la cinemática directa del robot, es presentada en una interfaz gráfica, donde se puede observar todos los movimientos que puede realizar el robot en un esquema tridimensional, y mostrar los indicadores de posicionamiento y orientación del mismo. Se puede realizar una operación manual del robot manipulador variando los ángulos de cada una de sus articulaciones  $q_1$ ,  $q_2$ ,  $q_3$ ,  $q_4$  y  $q_5$ .

Se determinó el modelo cinemático inverso por medio del método de desacoplamiento cinemático. Este método tiene una gran carga de cálculos matemáticos, en este caso se presentaron 72 posibles combinaciones para obtener el resultado y realizar la validación del modelo, debido a esto se hace uso de software especializado, para minimizar el tiempo de determinación de resultados.

Los resultados obtenidos por el algoritmo de validación de la cinemática inversa, logran verificar que las matrices de traslación son iguales, mediante la comparación de las mismas, como lo muestra en la Figura 16. Por tanto, el modelo está correctamente calculado.

Con el presente trabajo se logra obtener una herramienta de aprendizaje en el área de la robótica, ya que la programación de los algoritmos realizados, nos permiten simular los movimientos de un robot manipulador de cinco grados de libertad, donde el estudiante podrá aprender los conceptos básicos de la cinemática directa e inversa y la planificación de trayectorias. Por lo tanto, se podrá reforzar proceso de aprendizaje en este campo de la ingeniería.

### **Recomendaciones.**

Se puede plantear una contrastación de los modelos cinemáticos planteados, usando un software diferente, y de esta manera confirmar los resultados obtenidos. Se puede hacer uso del software de bajo costo RoboAnalyzer, que sirve para el diseño y modelado de robots.

En consecuencia, al desarrollo del presente trabajo, se pueden plantear posibles trabajos futuros, que aborden la simulación de trayectorias por posicionamiento, donde el usuario indique un punto en el espacio, en el que quiera que se mueva el robot. También se podría analizar el estudio de la Dinámica del robot, control cinemático y control dinámico, con la finalidad incluir carga de objetos y control de velocidades.

## **8. Referencias bibliográficas.**

- Barrientos, A., Penín, L., Balaguer, C., & Aracil, R. (2007). *Fundamentos de Robótica*. McGraw de España.
- Cakir, M., & Butun, E. (2007). An educational tool for 6-DOF industrial robots with quaternion algebra. *Computer Applications in Engineering Education*, 15(2), 143–154. <https://doi.org/10.1002/cae.20104>
- Cehovin, L., Rezelj, A., & Skocaj, D. (2017). Open-Source Robotic Manipulator and Sensory Platform. *Robotics in Education, Advances in Intelligent Systems and Computing*, 457. <https://doi.org/10.1007/978-3-319-42975-5>
- Corke, P. (2015). *Robotics Toolbox for Matlab release 9*.
- Craig, J. J. (2006). *Robótica* (3era Ed., Vol. 1, Issue 8020). Pearson Education, 2006.



- Crespo, R., Riestra, E., Gánem, R., & Cárdenas, D. (2013). Realidad Virtual como herramienta para aprendizaje inmersivo en ingeniería. *Revista Del Congreso Internacional de Innovación Educativa*.
- Denavit, J., & Hartenberg, R. S. (1955). A kinematic notation for lower pair mechanisms based on matrices. *Journal of Applied Mechanics*, 77, 215–221.
- Fu, K. S., González, R. C., & Lee, C. S. G. (1988). *ROBOTICA: Control, detección, visión e inteligencia*. McGraw-Hill.
- Gil, A., Reinoso, O., Marin, J. M., Paya, L., & Ruiz, J. (2015). Development and deployment of a new robotics toolbox for education. *Computer Applications in Engineering Education*, 23(3), 443–454. <https://doi.org/10.1002/cae.21615>
- González Alvarez, N. (2017). *Desarrollo de un entorno de simulación para un robot paralelo activado por ocho cables*.
- Hwang, M. J., Chung, S. Y., Lee, K., Song, I. J., & Son, H. II. (2016). Registration between robot and workpiece in virtual environment for off-line programming. *IECON Proceedings (Industrial Electronics Conference)*, 10060070, 779–784. <https://doi.org/10.1109/IECON.2016.7793685>
- Jara, C. A., Candelas-Herías, F. A., & Torres, F. (2007). Herramientas interactivas para la enseñanza de robótica. *Departamento de Física, Ingeniería de Sistemas y Teoría de La Señal, Universidad de Alicante*. <http://rua.ua.es/dspace/handle/10045/2228>
- Jha, A., Soni, M., & Suhaib, M. (2021). Simulation and kinematic analysis of KUKA KR5 Arc robot. *IOP Conference Series: Materials Science and Engineering*, 1149(1), 012005. <https://doi.org/10.1088/1757-899x/1149/1/012005>
- Jung, S. (2013). Experiences in developing an experimental robotics course program for undergraduate education. *IEEE Transactions on Education*, 56(1), 129–136. <https://doi.org/10.1109/TE.2012.2213601>
- Kang, S., & Chou, W. (2019). Kinematic Analysis, Simulation and Manipulating of a 5-DOF Robotic Manipulator for Service Robot. *Proceedings of 2019 IEEE International Conference on Mechatronics and Automation, ICMA 2019*, 643–649. <https://doi.org/10.1109/ICMA.2019.8816413>
- Lazaro-Arvizu, Y., Morales-Caporal, R., Ordoñez-Flores, R., Quintero-Flores, P., & Leal-Lopez, M. (2015). *Desarrollo de un software para la simulación y control de un robot industrial*. 2(5), 958–967.
- Liu, F., Gao, G., Shi, L., & Lv, Y. (2017). Kinematic analysis and simulation of a 3-DOF robotic manipulator. *3rd IEEE International Conference On*, 1–5. <https://doi.org/10.1109/CIACT.2017.7977291>
- López Cruz, E. (2018). *Control de Super-Torsión para un robot manipulador de n grados de libertad*. Universidad Autónoma del Carmen.
- Medina Cervantes, J., Villafuerte Díaz, R., & Mejía Sánchez, E. (2014).

*Simulador 3D para brazo robot de 4 grados de libertad.*

- Ndawula, I., & Assal, S. F. M. (2018). Conceptual design and kinematic analysis of a novel open field 3DoF multi-gripper pot seedlings transplanting robot. *Proceedings of 2018 IEEE International Conference on Mechatronics and Automation, ICMA 2018*, 1458–1463. <https://doi.org/10.1109/ICMA.2018.8484670>
- Ollero, A. (2001). *Robotica Manipuladores y robots moviles*. Alfaomega - Marcombo.
- Reyes Cortés, F. (2011). *Robótica control de robots manipuladores*. Alfaomega.
- Siciliano, B., Sciavicco, L., Villani, L., & Oriolo, G. (2009). *Robotics: Modeling, Planning, and Control*. Springer-Verlag London Limited. <https://doi.org/10.1109/MRA.2009.934833>
- Subir, K. (2008). *Introducción a la Robótica*. Mc-Graw.
- Thermo Corporation. (n.d.). *CRS CataLyst-5 Articulated Robot*.
- Torres, F., Candelas, F. A., Puente, S. T., Pomares, J., Gil, P., & Ortiz, F. G. (2006). *Experiences with Virtual Environment and Remote Laboratory for Teaching and Learning Robotics at the University of Alicante \**.
- Urrea, C., & Coltters, J. P. (2015). Design and implementation of a graphic 3D simulator for the study of control techniques applied to cooperative robots. *International Journal of Control, Automation and Systems*, 13(6), 1476–1485. <https://doi.org/10.1007/s12555-014-0278-y>
- Urrea, C., & Kern, J. (2012). Modeling, simulation and control of a redundant SCARA-type manipulator robot. *International Journal of Advanced Robotic Systems*, 9, 1–14. <https://doi.org/10.5772/51701>
- Velíšek, K., Holubek, R., Delgado Sobrino, D. R., Ruzarovský, R., & Vetríková, N. (2017). Design of a robotized workstation making use of the integration of CAD models and Robotic Simulation software as way of pairing and comparing real and virtual environments. *MATEC Web of Conferences*, 94. <https://doi.org/10.1051/matecconf/20179405008>
- Vithanage, R. K. W., Harrison, C. S., & Desilva, A. K. M. (2019). A Detailed Kinematic Analysis of a Commercial Grade Articulated Industrial Robot. *Proceedings - 2018 International Conference on Computing, Electronics and Communications Engineering, ICCECE 2018*, 1, 236–241. <https://doi.org/10.1109/iCCECOME.2018.8659139>

## 9. Anexos

### Algoritmo de validación de la cinemática directa

```
% Verificación de Cinemática Directa del Robot CRS Catalyst-5
% Números de Denavit Hartenberg

%   j   /   Theta   /   d   /   a   /   alpha
% -----
%   1   /   q1     /   272   /   0   /   90
%   2   /   q2     /   0     /   253.52 /   0
%   3   /   q3     /   0     /   253   /   0
%   4   /   q4     /   0     /   0     /   -90°
%   5   /   q5+180 /   145.71 /   0     /   0

%Asignación de Números de D-H a cada articulación mediante la función
Link

% Se debe ejecutar el fichero "startup_rvc.m" de la librería de Peter
Corke

% Link([Theta, d , a , alpha , sigma , offset ])
%en milímetros
% L1=Link([ 0 272 0 pi/2 0 0 ]);
% L2=Link([ 0 0 253.52 0 0 0 ]);
% L3=Link([ 0 0 253 0 0 0 ]);
% L4=Link([ 0 0 0 -pi/2 0 0 ]);
% L5=Link([ 0 145.71 0 0 0 pi ]);

%en centímetros
L1=Link([ 0 27.2 0 pi/2 0 0 ]);
L2=Link([ 0 0 25.352 0 0 0 ]);
L3=Link([ 0 0 25.3 0 0 0 ]);
L4=Link([ 0 0 0 -pi/2 0 0 ]);
L5=Link([ 0 14.571 0 0 0 pi ]);

bot=SerialLink([L1 L2 L3 L4 L5], 'name', 'Robot CRS');
bot.n
bot.fkine([0 pi/2 -pi/2 -pi/2 0])
bot.plot([0 pi/2 -pi/2 -pi/2 0]);
bot.teach([0 pi/2 -pi/2 -pi/2 0]);
```

### Algoritmo de validación de la cinemática inversa

```
% Verificación de cinemática inversa utilizando 72 combinaciones.
% Definir variables de los angulos q1 q2 q3 q4 q5
syms q1 q2 q3 q4 q5;
% % Asignación de Valores a q1 q2 q3 q4 q5
% q1= -90*pi/180; % desde -180 hasta +180 (límites -179
hasta 179)
% q2= -20*pi/180 +pi/2; % desde -90 hasta +10 la suma de q2 y q3
no puede superar de 79 grados
% q3= 10*pi/180 -pi/2; % desde -35 hasta +90
% q4= 30*pi/180 -pi/2; % desde -110 hasta +110
```

```

% q5= 60*pi/180;           % desde -180 hasta 180 (limites -179 hasta
179) 1.3789

q1= -pi/2;                 % desde -180 hasta +180 (límites -179
hasta 179)
q2= 7*pi/18;               % desde -90 hasta +10 la suma de q2 y q3
no puede superar de 79 grados
q3= -4*pi/9;               % desde -35 hasta +90
q4= -pi/3;                 % desde -110 hasta +110
q5= pi/3;                  % desde -180 hasta 180 (limites -179 hasta
179) 1.3789

% Asignación de las Variables de las matrices de Transformación
Homogénea
% 0A1= A1, 1A2=A2, 2A3=A3, 3A4=A4, 4A5=A5,
syms A1 A2 A3 A4 A5;
A1=[cos(q1) 0 sin(q1) 0 ; sin(q1) 0 -cos(q1) 0 ; 0 1 0 272;0 0 0 1];
A2=[cos(q2) -sin(q2) 0 253.52*cos(q2); sin(q2) cos(q2) 0
253.52*sin(q2); 0 0 1 0; 0 0 0 1];
A3=[cos(q3) -sin(q3) 0 253*cos(q3);sin(q3) cos(q3) 0 253*sin(q3); 0 0
1 0;0 0 0 1];
A4=[cos(q4) 0 -sin(q4) 0;sin(q4) 0 cos(q4) 0; 0 -1 0 0; 0 0 0 1];
A5=[-cos(q5) sin(q5) 0 0;-sin(q5) -cos(q5) 0 0;0 0 1 145.71; 0 0 0 1];

%Cálculo de Matriz T
T5=A1*A2*A3*A4*A5

%% Verificación de Cinemática Inversa

syms Q1_1 Q1_2 Q1_3;
syms Q2_1 Q2_2;
syms Q3_1 Q3_2;
syms Q4_1 Q4_2;
syms Q5_1 Q5_2 Q5_3
syms QQ1 QQ2 QQ3;

syms Q1 Q2 Q3 Q4 Q5;
syms px py pz l1 l2 l3 ax ay az nx ny nz ox oy oz;
syms CosQ3;
syms x i1 i2 i3 i4 i5 i6 w m n o p r s t;
x=1; i1=1; i2=1; i3=1; i5=1; w=1; m=0; n=0; o=0; p=0; r=0; s=0; t=0;

nx=T5(1); ox=T5(5); ax=T5(9);
ny=T5(2); oy=T5(6); ay=T5(10);
nz=T5(3); oz=T5(7); az=T5(11);

r5=[T5(13) T5(14) T5(15)]; % estos serían Px Py y Pz a colocar para
calcular la inversa.
z5=[T5(9) T5(10) T5(11)]; % estos serían r13 r23 y r33 px py pz para
calcular la inversa.
rm=r5-145.71*z5; % ecuación derivada del desacoplo cinemático
px=rm(1); % esto es px del punto M es decir, del eje coord 3
py=rm(2); % esto es py del punto M es decir, del eje coord 3
pz=rm(3); % esto es pz del punto M es decir, del eje coord 3

l1=272; l2=253.52; l3=253;

% Ecuaciones para calcular Q1 se tienen 3 opciones
if T5(13)<0&&T5(14)>=0
    Q1_1=atan(py/px)+pi;
    Q1_2=atan(py/px)-pi;

```

```

    Q1_3=atan (py/px) ;
end
if T5 (13)<0&&T5 (14)<0
    Q1_1=atan (py/px) -pi;
    Q1_2=atan (py/px) +pi;
    Q1_3=atan (py/px) ;
end
if T5 (13)>=0
    Q1_1=atan (py/px) ;
    Q1_2=atan (py/px) +pi;
    Q1_3=atan (py/px) -pi;
end
if T5 (13)<0&&T5 (5)<0
    Q1_1=atan (py/px) ;
    Q1_2=atan (py/px) +pi;
    Q1_3=atan (py/px) -pi;
end

%Ecuaciones para calcular Q3
CosQ3=(( px^2) + (py^2) + ((pz-11) ^2) - (12^2) - (13^2)) / (2*12*13) ;

Q3_1=atan((-sqrt(1-((CosQ3)^2)))/(CosQ3));
Q3_2=atan((-sqrt(1-((CosQ3)^2)))/(CosQ3))-pi;

while x == 1
    %índice para combinacion de q4
    if rem(w,2)==0
        i4=2;
    else
        i4=1;
    end
    % índice para combinacion de q5
    if w == 3 || w == m
        m = w + 2;
        i5=i5+1;
    end
    if (w == 7 || w == n)
        n = w + 6;
        i5 = 1;
    end
    % índice para combinacion de q2
    if (w == 7 || w == o)
        o = w + 12;
        i2 = 2;
    end
    if (w == 13 || w == p)
        p = w + 12;
        i2 = 1;
    end
    % índice para combinacion de q3
    if(w == 13 || w == r)
        r = w + 24;
        i3 = 2;
    end
    if (w == 25 || w == s)
        s = w + 24;
        i3 = 1;
    end
    % índice para combinacion de q1
    if (w == 25 || w == t)
        t = w + 24;
    end
end

```

```

        i1=i1+1;
end

%Ecuaciones para calcular Q2
if i3==1
    QQ3=Q3_1;
end
if i3==2
    QQ3=Q3_2;
end

Q2_1=atan((pz-l1)/(sqrt((px^2)+(py^2))))-
atan((l3*sin(QQ3))/(l2+(l3*cos(QQ3))));
Q2_2=pi+atan((l1-pz)/(sqrt((px^2)+(py^2))))-
atan((l3*sin(QQ3))/(l2+(l3*cos(QQ3))));

%Ecuaciones Para calcular Q5
if i1==1
    QQ1=Q1_1;
end
if i1==2
    QQ1=Q1_2;
end
if i1==3
    QQ1=Q1_3;
end
Q5_1=atan(((sin(QQ1)*nx)-(cos(QQ1)*ny))/((sin(QQ1)*ox)-
(cos(QQ1)*oy)));
Q5_2=atan(((sin(QQ1)*nx)-(cos(QQ1)*ny))/((sin(QQ1)*ox)-
(cos(QQ1)*oy)))-pi;
Q5_3=atan(((sin(QQ1)*nx)-(cos(QQ1)*ny))/((sin(QQ1)*ox)-
(cos(QQ1)*oy)))+pi;

%Ecuaciones Para Calcular Q4
if i2==1
    QQ2=Q2_1;
end
if i2==2
    QQ2=Q2_2;
end
v1=((cos(QQ1)*cos(QQ2)*cos(QQ3))-(cos(QQ1)*sin(QQ2)*sin(QQ3)))*ax;
v2=((cos(QQ2)*cos(QQ3)*sin(QQ1))-(sin(QQ1)*sin(QQ2)*sin(QQ3)))*ay;
v3=((cos(QQ2)*sin(QQ3))+(cos(QQ3)*sin(QQ2)))*az;

v4=((cos(QQ1)*cos(QQ2)*sin(QQ3))+(cos(QQ1)*cos(QQ3)*sin(QQ2)))*ax;
v5=((cos(QQ2)*sin(QQ1)*sin(QQ3))+(cos(QQ3)*sin(QQ1)*sin(QQ2)))*ay;
v6=((sin(QQ2)*sin(QQ3))-(cos(QQ2)*cos(QQ3)))*az;

Q4_1=atan((v1+v2+v3)/(v4+v5+v6));
Q4_2=atan((v1+v2+v3)/(v4+v5+v6))-pi;

% Valores de q1, q2, q3, q4 y q5 dependiendo de las combinaciones
y los índices i1, i2, i3, i4, i5
if i1 == 1
    Q1 = Q1_1;
end
if i1 == 2
    Q1 = Q1_2;
end
if i1 == 3
    Q1 = Q1_3;
end

```

```

end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
if (i2 == 1)
    Q2 = Q2_1;
end
if (i2 == 2)
    Q2 = Q2_2;
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
if (i3 == 1)
    Q3 = Q3_1;
end
if (i3 == 2)
    Q3 = Q3_2;
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
if (i4 == 1)
    Q4 = Q4_1;
end
if (i4 == 2)
    Q4 = Q4_2;
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
if (i5 == 1)
    Q5 = Q5_1;
end
if (i5 == 2)
    Q5 = Q5_2;
end
if (i5 == 3)
    Q5 = Q5_3;
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
w=w+1;

B1=[cos(Q1) 0 sin(Q1) 0 ; sin(Q1) 0 -cos(Q1) 0 ; 0 1 0 272;0 0 0 1];
B2=[cos(Q2) -sin(Q2) 0 253.52*cos(Q2); sin(Q2) cos(Q2) 0
253.52*sin(Q2); 0 0 1 0; 0 0 0 1];
B3=[cos(Q3) -sin(Q3) 0 253*cos(Q3);sin(Q3) cos(Q3) 0 253*sin(Q3); 0 0
1 0;0 0 0 1];
B4=[cos(Q4) 0 -sin(Q4) 0;sin(Q4) 0 cos(Q4) 0; 0 -1 0 0; 0 0 0 1];
B5=[-cos(Q5) sin(Q5) 0 0;-sin(Q5) -cos(Q5) 0 0;0 0 1 145.71; 0 0 0 1];
R5=B1*B2*B3*B4*B5;
syms limit1; limit1=Q2+Q3; % restricción: la suma de los ángulos Q2 y
Q3 no debe superar 1,38 rad
if (1.570<= Q2 && Q2 <= 1.571)
    limit1=0;
end
if (-1.570<= Q3 && Q3 <= -1.571 && limit1==1)
    limit1=0;
end

% estos serían Px, Py y Pz a colocar para calcular la inversa.
% estos serían r13, r23 y r33 px py pz para calcular la inversa.
if (R5(13)-0.001<=T5(13) && T5(13)<= R5(13)+0.001 &&...
R5(14)-0.001<=T5(14) && T5(14)<= R5(14)+0.001 &&...
R5(15)-0.001<=T5(15) && T5(15)<= R5(15)+0.001 &&...
R5(9) -0.001<=T5(9) && T5(9) <= R5(9) +0.001 &&...
R5(10)-0.001<=T5(10) && T5(10)<= R5(10)+0.001 &&...
R5(11)-0.001<=T5(11) && T5(11)<= R5(11)+0.001 &&...
R5(5) -0.001<=T5(5) && T5(5) <= R5(5) +0.001 &&...

```

```

R5(6) -0.001<=T5(6)  && T5(6) <= R5(6)+0.001 &&...
R5(7) -0.001<=T5(7)  && T5(7) <= R5(7)+0.001 &&...
Q1 >= -3.142 && Q1<=3.142&&... %      Q1 >= -3.1416 &&
Q1<=3.1416&&...
Q2 >= 0          && Q2<=1.76&&... %      Q2 >= 0          &&
Q2<=1.7453&&...
Q3 >= -2.182 && Q3<=0          &&...%      Q3 >= -2.1817 && Q3<=0
&&...
Q4 >= -3.491 && Q4<=0.350&&... %      Q4 >= -3.4907 &&
Q4<=0.3491&&...
Q5 >= -3.142 && Q5<=3.142&&... %      Q5 >= -3.1416 &&
Q5<=3.1416&&...
limit1<=1.38);          %      limit1 <= 1.3789

fprintf('Inversa hallada correctamente, Iteraciones %d\n',w);

x=0;
end
if w==73;
    disp('Inversa No hallada');
    x=0;
end
end

B1=[cos(Q1) 0 sin(Q1) 0 ; sin(Q1) 0 -cos(Q1) 0 ; 0 1 0 272;0 0 0 1];
B2=[cos(Q2) -sin(Q2) 0 253.52*cos(Q2); sin(Q2) cos(Q2) 0
253.52*sin(Q2); 0 0 1 0; 0 0 0 1];
B3=[cos(Q3) -sin(Q3) 0 253*cos(Q3);sin(Q3) cos(Q3) 0 253*sin(Q3); 0 0
1 0;0 0 0 1];
B4=[cos(Q4) 0 -sin(Q4) 0;sin(Q4) 0 cos(Q4) 0; 0 -1 0 0; 0 0 0 1];
B5=[-cos(Q5) sin(Q5) 0 0;-sin(Q5) -cos(Q5) 0 0;0 0 1 145.71; 0 0 0 1];

fprintf('Matriz R5 con ángulos calculados por Inversa ');
R5=B1*B2*B3*B4*B5

```