



UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería Informática



Trabajo Final del Grado en Ingeniería
Informática

Trabajo de Final de Grado
ROBÓTICA: Modelado de Cinemática
Directa e Inversa basado en el Algoritmo de
Denavit-Hartenberg – Desacoplamiento de
Subproblemas

Memoria



Presentado por Jaime Sáiz de la Peña
en Universidad de Burgos
17 de enero de 2023

Tutores:

José Manuel Sáiz Diez
Raúl Marticorena Sánchez





UNIVERSIDAD DE BURGOS

ESCUELA POLITÉCNICA SUPERIOR

Grado en Ingeniería Informática



D. José Manuel Sáiz Diez, profesor del departamento de Ingeniería Informática, área de Lenguajes y Sistemas Informáticos

Expone:

Que el alumno D. Jaime Sáiz de la Peña, con DNI 71311023Y, ha realizado el Trabajo final del Grado en Ingeniería Informática, titulado: ROBÓTICA: Modelado de Cinemática Directa e Inversa basado en el Algoritmo de Denavit-Hartenberg/Desacoplamiento de Subproblemas.

y que dicho trabajo ha sido realizado por el alumno bajo la dirección del que suscribe, en virtud de lo cual, Se autoriza su presentación y defensa.

En Burgos, a 17 de enero de 2023

SAIZ DIEZ JOSE
MANUEL -
14590420W

Firmado digitalmente por SAIZ DIEZ JOSE
MANUEL - 14590420W
Nombre de reconocimiento (DN): c=ES,
serialNumber=IDCES-14590420W,
givenName=JOSE MANUEL, sn=SAIZ DIEZ,
cn=SAIZ DIEZ JOSE MANUEL - 14590420W
Fecha: 2023.01.14 00:26:38 +01'00'

José Manuel Sáiz Diez

“This work is licensed under the Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-nc-sa/4.0/> or send a letter to Creative Commons, PO Box 1866, Mountain View, CA 94042, USA.”

Resumen

Hoy en día, los robots son esenciales para mantener el estado de bienestar en el que la sociedad se ha acostumbrado a vivir. Desde la revolución industrial, han ayudado al hombre en diferentes tareas, centradas especialmente en la industria, donde se les ha sacado hasta ahora el mayor provecho. No obstante, con la evolución que a su vez han sufrido los procesadores y su capacidad de cálculo, cada vez tienen más utilidades en el día a día y ofrecen una mayor gama de posibilidades.

Actualmente, se pueden encontrar robots en una gran cantidad de sectores distintos como pueden ser la medicina, la logística, la industria o la educación entre otros. Se observa como los robots ya no tienen como único objetivo la mejora de la producción en la industria, sino que ahora ofrecen servicios y mejoran el nivel de vida del día a día. Y al ser las utilidades tan variadas, también lo son las formas en las que se presentan dichos robots. Por ejemplo, no tiene nada que ver una “aspiradora robotizada” con un robot de tipo COBOT.

Este TFG tiene como objetivo, facilitar la creación de las ecuaciones necesarias para calcular los movimientos de los brazos robóticos, abriendo las puertas a un desarrollo de robots más rápido y sencillo al simplificar el desarrollo de la parte matemática de los modelos robóticos planteados.

Este trabajo se compone por un conjunto de programas que serán usados en función de las necesidades del usuario. Los resultados obtenidos, serán una herramienta de ayuda para el desarrollo del modelado de cada problema planteado, a través de un Desarrollo Analítico. Estos resultados obtenidos sirven, tanto para resolver los problemas con Cinemática Directa, como con Cinemática Inversa, partiendo del Algoritmo de Denavit-Hartenberg (basado en Matrices de Transformación Homogéneas) y añadiendo además una técnica de Desacoplamiento en Subproblemas (Grupos de Articulaciones), más sencillos de resolver. Además se determinará la ventaja de definir Planos de Ataque y Ángulos de Ataque para acercarse al punto final con un ángulo determinado.

Descriptores

Brazo robótico, Robot Colaborativo (COBOT), Cinemática Directa, Cinemática Inversa, Desarrollo Analítico, Matrices de Transformación Homogéneas, Denavit-Hartenberg, Desacoplamiento, Grupos de Articulaciones, Plano de Ataque, Ángulo de Ataque.

Abstract

Nowadays, robots are essential to maintain the welfare state in which society has become accustomed to living. Since the industrial revolution, they have helped humans in different tasks, especially focused on industry, where up to now the greatest benefit has been obtained. However, with the evolution that processors and their calculation capacity have undergone, they have more and more day-to-day uses and offer a greater range of possibilities.

Currently, robots can be found in a large number of different sectors such as medicine, logistics, industry or education, among others. Robots no longer have the only objective of increasing productivity in the industry, but now offer services and improve human life on a day-to-day basis. Since the uses are so varied, so are the ways in which these robots are built. For example, a "robotic vacuum" has nothing to do with a COBOT-type robot.

The objective of this TFG is to improve the creation of the necessary equations to calculate the movements of the model robotic arms, opening the doors to a faster and easier development of robots by simplifying the development of the mathematical part of the proposed robotics.

This work is made up of a set of programs that will be used depending on the user's needs. The results obtained will be a help tool for the development of the modeling of each problem raised, through an Analytical Development. These obtained results serve to solve problems with Forward Kinematics as well as with Inverse Kinematics, starting from the Denavit-Hartenberg Algorithm (based on Homogeneous Transformation Matrix) and also adding a technique of Decoupling in Subproblems (Groups of Joints), more simple to solve. In addition, the advantage of defining Attack Planes and Attack Angles to approach the end point with a certain angle will be determined.

Keywords

Robotic Arm, Collaborative Robots (COBOT), Forward Kinematics, Inverse Kinematics, Analytical Development, Homogeneous Transformation Matrix, Denavit-Hartenberg, Decoupling, Groups of Joints, Attack Planes, Attack Angles.

Índice de contenido

1	Introducción.....	15
1.1	Estructura de la memoria.....	16
1.2	Materiales entregados.....	17
2	Objetivos del proyecto.....	18
2.1	Objetivos generales.....	18
2.2	Objetivos en investigación.....	18
2.3	Objetivos personales.....	18
3	Conceptos teóricos.....	19
3.1	Sistemas de coordenadas y la regla de la mano derecha.....	20
3.2	Razones trigonométricas.....	22
3.3	Teoremas de cos y sin.....	24
3.4	Ecuaciones trascendentes.....	25
3.4.1	Un caso de ecuación trascendente: $P_x \sin(\theta) + P_y \cos(\theta) = CTE$	25
3.5	Tipos de articulaciones.....	26
3.6	Cinemática	28
3.6.1	Cinemática Directa.....	28
3.6.2	Cinemática Inversa.....	29
3.7	Denavit-Hartenberg.....	30
3.7.1	Cinemática Directa a partir de DH.....	33
3.7.2	Cinemática Inversa a partir de DH	34
3.8	Desacoplo en subproblemas.....	39
4	Técnicas y herramientas.....	40
4.1	Repositorio	40
4.2	Git.....	40
4.3	GitHub vs GitLab vs AWS CodeCommit.....	41
4.4	Metodología Scrum.....	42
4.5	Lenguaje de programación: C.....	43
4.6	MEX	43
4.7	Documentación	44
4.8	Arduino.....	45
4.9	Librerías	47
4.9.1	Robotics System Toolbox.....	47
4.9.2	Symbolic Math Toolbox.....	47
4.9.3	Peter Corke Robotics Toolbox	48
5	Aspectos relevantes del desarrollo del proyecto.....	49
5.1	Estrategias de aproximación al extremo del brazo para el cálculo en Cinemática Inversa – Planteamiento inicial.....	50
5.2	Buscando soluciones lógicas a la caracterización de los brazos robóticos (Desacoplamiento y Plano y Ángulo de Ataque).....	53
5.3	Metodología para el cálculo de datos desde la aplicación y su uso en la caracterización de brazos robóticos – Resolución final	55
5.4	Casos planteados y resueltos.....	61
6	Trabajos relacionados.....	62
6.1	Relacionados con la funcionalidad.....	62
6.1.1	DOBL.....	62
6.1.2	Humanoide 17DOF.....	62
6.1.3	MarchProject.....	63
6.1.4	Hexápodo Phoenix de Lynxmotion.....	64
6.2	Relacionados con la implementación.....	65

6.2.1	Modelado cinemático directo de robot esférico.....	65
6.2.2	Modelación Dinámica y Simulación de Robots.....	65
6.2.3	Forward Kinematics:The Denavit-Hartenberg convention.....	66
6.2.4	Comparación trabajos relacionados	67
6.3	Alternativas al algoritmo de Denavit-Hartenberg para resolución de problemas de Cinemática Inversa.....	67
7	Conclusiones y líneas de trabajo futuras.....	69
7.1	Conclusiones relacionadas con el proyecto.....	69
7.2	Líneas de trabajo futuras.....	69
8	Referencias Bibliográficas.....	71

Índice de ilustraciones

Ilustración 1: Regla de la mano derecha [WWWjsvanegascDoc1].....	20
Ilustración 2: Levógiro y dextrógiro [WWWjsvanegascDoc1].....	20
Ilustración 3: Cuadrante sistema de coordenadas [WWWwikipediaDoc0].....	21
Ilustración 4: Ángulos Suplementarios.....	22
Ilustración 5: Ángulos Complementarios	22
Ilustración 6: Ángulos que difieren en 180°	22
Ilustración 7: Ángulos opuestos	23
Ilustración 8: Coseno de la suma:	23
Ilustración 9: Teoremas del seno y del coseno.....	24
Ilustración 10: Brazo con dos motores perpendiculares.....	25
Ilustración 11: Tipos de robots básicos (Centro: [WWWdocplayerDoc0] - Derecha: [WWWslideshareDoc2]).....	27
Ilustración 12: Tipos de articulaciones [WWWdocplayerDoc1].....	28
Ilustración 13: Ejemplo cinemática directa.....	29
Ilustración 14: Ejemplo cinemática inversa.....	29
Ilustración 15: Rotaciones sobre los ejes de un sistema de coordenadas.....	32
Ilustración 16: Regla de la mano derecha.....	32
Ilustración 17: Giro sobre el eje x.....	32
Ilustración 18: Ejemplo de dualidad de los ejes de dos motores.....	33
Ilustración 19: Ejemplo doble solución CI I.....	35
Ilustración 20: Ejemplo doble solución CI II.....	35
Ilustración 21: "Cinemática Inversa del Robot" [WWWkramirez.netDoc0].....	39
Ilustración 22: Logo de Git [WWWGitDoc1].....	40
Ilustración 23: Repositorio GitHub [WWWgithubDoc1].....	41
Ilustración 24: Repositorio GitLab [WWWgitlabDoc1].....	41
Ilustración 25: Repositorio AWS CodeCommit [WWWSourceForgeDoc1].....	41
Ilustración 26: Gestión de las ramas en GitHub y GitLab[WWWkinstaDoc].....	42
Ilustración 27: Metodología Scrum [WWWganttproDoc1].....	43
Ilustración 28: Mex setup.....	44
Ilustración 29: Ejemplo OpenOffice Draw.....	45
Ilustración 30: Ejemplos placas Arduino [WWWxatakaDoc0].....	46
Ilustración 31: Ejemplo librería "Robotics System Toolbox" [WWWmathworksDoc0].....	47
Ilustración 32: Ejemplo uso de variables simbólicas.	47
Ilustración 33: Ejemplo operación variables simbólicas.....	48
Ilustración 34: Ejemplo uso de variables simbólicas.	48
Ilustración 35: Brazo de prueba para presentación de TFG.....	50
Ilustración 36: Estrategias de alcanzabilidad del Brazo de prueba para presentación de TFG.....	52
Ilustración 37: DOBI Humanoid Intelligent Programmable RC Robot [WWWauselectronicsdirectDoc0].....	62
Ilustración 38: Robot educacional de 17DOF Biped [WWWaliexpressProd148].....	63
Ilustración 39: Humanoid Robotic Hand [WWWaliexpressProd148].....	63
Ilustración 40: Exoesqueleto "MarchProject" [WWWprojectmarchDoc1].....	64
Ilustración 41: Hexápodo Phoenix de Lynxmotion [WWWaskixDoc0]	64
Ilustración 42: Modelado cinemático directo de robot esférico con 3 grados de libertad [WWWocw.ehu.eusDoc1].....	65
Ilustración 43: Algoritmo de Denavit-Hartenberg modificado (John J. Craig) [WWWmtorrest.sitios.ing.uc.clDoc0]	65
Ilustración 44: Forward Kinematics:The Denavit-Hartenberg convention [WWWusers.cs.duke.eduDoc0]	66

Índice de tablas

Tabla 1: Ejemplo parámetros de Denavit-Hartenberg.....	32
Tabla 2: Estudio de una cadena cinemática utilizando los algoritmos de Denavit-Hartenberg y Craig modificado [WWWdocplayerDoc7].....	33
Tabla 3: Diferencias trabajos relacionados.....	67
Tabla 4: Propiedades de varios Algoritmos de Cinemática Inversa [WWWplatea.pntic.mecDoc0]	68

1 Introducción

La robótica ya está en la mente de todos. Las noticias que aparecen a diario nos indican que representa el presente y el futuro en todos los campos, y sobre todo en la **Industria**.

De hecho, los robots industriales que **repiten los mismos movimientos** de forma constante, al estar orientados a la producción de forma masiva, están siendo aplicados desde hace décadas. Y los **Robots Colaborativos (COBOT)** que comparten el mismo campo de trabajo con el personal humano, de manera fiable, son ya el presente, siendo un buen ejemplo el entorno médico.

La gran mayoría de los sistemas robóticos están basados en sistemas ubicados en un emplazamiento fijo, y sin embargo, el problema planteado para sistemas sin una ubicación fija son similares en cuanto a su desarrollo. De hecho, están basados en **brazos, ubicados sobre una plataforma móvil**, que se desplazan en un espacio cercano, para acceder a un punto predeterminado, para cubrir una cierta necesidad mediante algún tipo de herramienta, dependiente de cada caso en particular.

En cuanto a las herramientas, dependen de cada caso, por lo que no se pretende resolver ese problema en el actual Trabajo Fin de Grado (TFG). Sin embargo, el posicionamiento de la base y el posicionamiento de la herramienta en el punto exacto desde donde va a ser usada, sí puede ser un problema cuya generalización puede ser objeto de este trabajo.

Y en cuanto al posicionamiento de la herramienta en un punto, se pueden definir dos caminos. Por un lado el movimiento de la base (sistema rodado sobre un plano regular, o a través de un sistema que sea capaz de caminar sobre un plano irregular, o sobre peldaños). Y por otro, una vez alcanzada una cierta ubicación por parte de la base, se debe alcanzar un cierto punto en el espacio cercano donde colocar la herramienta con una cierta orientación.

En ambos casos, tanto si la ubicación de la base se realiza mediante algún tipo de “pata”, como si la herramienta se coloca en un punto determinado en el espacio cercano a través de un “brazo”, se basará en los mismos elementos, **brazos robóticos** con una serie de articulaciones.

Por último, estos brazos robóticos representan un problema conocido, como es el modelado de cada brazo robótico, dependiente de su funcionalidad (“patas” del robot para caminar, que deben alcanzar un punto de apoyo sobre los que ir avanzando, o “brazo” con el que alcanzar un punto en el espacio). Por otro lado el desarrollo de las ecuaciones que caractericen cada brazo robótico y que cumplan con las necesidades y restricciones del modelo, y la codificación de la aplicación que controle cada brazo robótico y el movimiento de todos ellos de una forma sincronizada haciendo uso de las ecuaciones anteriormente creadas.

Además, hay dos formas de trabajar con brazos robóticos, mediante **Cinemática Directa** (definiendo los ángulos de todos los motores incluidos en las diferentes articulaciones para los motores rotacionales, o la traslación sobre un único eje para los motores prismáticos, y calculando el punto alcanzado) y mediante **Cinemática Inversa** (definiendo el punto que se quiere alcanzar, y calculando cuáles son los ángulos y traslaciones lineales necesarios para ello). [WWWdlscribDoc0]

En caso de querer determinar los ángulos y traslaciones, y después calcular cuál es el punto alcanzado (Cinemática Directa), el problema se puede resolver de varias formas, por ejemplo, mediante un modelado geométrico, o a través del Algoritmo de **Denavit-Hartenberg**, entre otros.

Sin embargo, si se quiere alcanzar un punto predeterminado, y para ello se deben calcular los ángulos de los motores o las traslaciones lineales (variables articulares), resulta más complicado, ya que será necesario encontrar las ecuaciones que caractericen el modelo para calcular esas variables

articulares.

El desarrollo de esas ecuaciones también se puede hacer de varias formas. A través de un desarrollo geométrico (realizando un análisis trigonométrico del modelo), mediante un **desarrollo analítico** (usando diferentes recursos matemáticos como Matrices de Transformación Homogéneas, Matrices de Desplazamiento a través de la Fórmula de Rodrigues), y mediante Sistemas Numéricos (mecanismos matemáticos iterativos como Fabrik). [WWWUVADoc1]

El desarrollo geométrico resulta muy complejo cuando el número de articulaciones es, incluso bajo, por ejemplo, más de tres articulaciones.

Los desarrollos numéricos se basan en cierto grado de recursividad, lo que hace que el sistema sea relativamente lento o que no garantice llegar a un resultado aceptable en un tiempo razonable.

Y los sistemas analíticos también resulta relativamente complejos cuando el número de articulaciones crece. Sin embargo, son exactos y notablemente más sencillo cuando el número de articulaciones es alto.

En este contexto, el presente TFG, “*Modelado de Cinemática Directa e Inversa mediante el Algoritmo de Denavit-Hartenberg*”, facilitará el proceso de creación de un modelo de brazo robótico, su análisis, el desarrollo de las ecuaciones que lo caractericen y el cálculo de los datos que determinen cada movimiento en tiempo real.

Para ello se propondrá una alternativa al desarrollo individual de la Cinemática Directa e Inversa de los brazos robóticos, proceso costoso en cuanto a tiempo y recursos. En definitiva, se evitará tener que resolver este tipo de problemas de forma trigonométrica y computacional para cada modelo en particular.

Esto no significa que el usuario no deba tener unos conocimientos básicos sobre robótica y matemáticas, ya que es importante introducir los datos de forma correcta, así como determinar las condiciones y restricciones de cada modelo.

Además el desarrollo realizado será probado en diferentes **robots reales**, no solo para probar que los **resultados obtenidos** son **aplicables**, y los **datos obtenidos** en las comprobaciones desarrolladas, son **exactos**, sino para comprobar los problemas que puedan presentarse durante su implantación por las **limitaciones** de las **arquitecturas** de los propios modelos y por las necesarias limitaciones de las piezas de acoplamiento utilizadas en su **montaje**.

1.1 Estructura de la memoria

La memoria cuenta de varios capítulos cuyo resumen se presenta a continuación.

- **Introducción:** en este apartado se dibuja de forma simple la idea general del TFG, así como su organización y composición.
- **Objetivos del proyecto:** este apartado recoge cuales son los objetivos de este TFG, así como las variaciones producidas en función de los problemas y nuevas necesidades que han surgido. Los objetivos se separan en tres tipos: generales, técnicos y personales.
- **Conceptos teóricos:** se expone la base matemática teórica sobre la que se sustenta el TFG. También se explica la metodología seguida para determinar las ecuaciones que caracterizan los diferentes modelos robóticos, fundamentalmente para resolver problemas de cinemática

inversa.

- **Técnicas y herramientas:** explicación, partiendo de la base teórica anteriormente mencionada, de las técnicas, herramientas y soluciones utilizadas para resolver los distintos problemas.
- **Trabajos relacionados:** en este apartado se exponen trabajos similares tanto desde un punto de vista funcional, como desde un punto de vista de algoritmos e implementación.
- **Conclusiones y líneas de trabajo futuras:** se presentan las conclusiones obtenidas tras la realización del TFG. Estas conclusiones están tanto en el plano teórico como práctico. Respecto a las líneas de trabajo futuras y posibles mejoras se separan en cuanto a la usabilidad y las mejoras funcionales.

1.2 Materiales entregados

Junto con esta memoria se entrega un dispositivo USB con el siguiente material.

- La memoria del TFG en formato PDF.
- Los anexos en formato PDF.
 - Anexo: Documentación Técnica.
 - Anexo: Casos básicos y aplicación en otras arquitecturas.
 - Anexo: Caso: 1.0 Brazo Robótico.
 - Anexo: Caso: 7.0 Humanoide.
 - Anexo: Caso: 10 – Motor RMD-x-8 – Biblioteca.
- El código fuente de los distintos programas que forman el TFG.
- Las librerías usadas para Arduino.
- La librería “no oficial” de *MATLAB* utilizada.
- Una serie de vídeos sobre el TFG y la funcionalidad del mismo (aplicando los resultados a distintos brazos robóticos, tanto independientes, como formando parte de otros robots más complejos).
- Repositorio de *GitHub*: <https://github.com/JaimeSaiz/Robotic-Proyects->
- Copia de la máquina virtual en la que se han realizado todas las pruebas (excepto las relacionadas con Arduino) con todo configurado, incluyendo plataformas de edición y ejecutables.

2 Objetivos del proyecto

2.1 Objetivos generales

El objetivo principal del TFG es facilitar el cálculo de la cinemática directa e inversa de brazos robóticos mediante el uso de una aplicación informática. Este objetivo se consigue a partir del cálculo de los parámetros de Denavit-Hartenberg (DH).

Los parámetros de DH obtenidos servirán para definir los modelos planteados. Y las matrices obtenidas desde estos parámetros (Matrices de Transformación Homogénea), cuyos elementos estarán formados por una serie de términos trigonométricos que determinan la posición y orientación relativas de los diferentes Sistemas de Ejes, servirán para probar que las soluciones calculadas para la cinemática inversa son correctas.

Además, se hará uso de los elementos trigonométricos que conforman las matrices para realizar el análisis de cada modelo planteado, con el fin de determinar las ecuaciones que permitan calcular las variables articulares (ángulos de los motores rotacionales). De esa forma, tanto si se ha hecho mediante modelos analíticos (cinemática inversa basada en los parámetros de Denavit-Hartenberg), como numéricos (Fabrik) o geométricos, se podrá probar que las ecuaciones obtenidas dan los mismos resultados.

Una vez obtenidas las ecuaciones que caracterizan los modelos planteados, se probarán en robots reales en los que se calcularán las variables articulares para alcanzar una serie de puntos predeterminados. Por tanto, se trata de comprobar a través de las matrices de transformación homogéneas obtenidas desde el Algoritmo de DH, que realmente se alcanzan esos puntos predeterminados, con lo que se determinará que las ecuaciones calculadas serán correctas.

2.2 Objetivos en investigación

En este TFG se pretende encontrar, analizar y mejorar un método analítico genérico aplicable a todos los brazos robóticos. Este método debe ser implementado en *C* y *MATLAB*.

Para ello se usará una serie de metodologías, ya existentes pero, hasta ahora, no usadas de forma conjunta (Aplicaciones de ayuda al análisis de caracterización de modelos robóticos, Algoritmo de Denavit-Hartenberg, Desacoplamiento en subproblemas de alcanzabilidad y no solo de posicionamiento de herramientas finales, resolución de variables articulares interdependientes, determinación de Planos y Ángulos de Ataque).

Otro objetivo es demostrar el funcionamiento de los programas y metodologías creadas, aplicándolos a distintos robots.

2.3 Objetivos personales

Este TFG debe servir como base fundamental a la robótica actual. El aprendizaje a nivel personal es importante. Se espera que este TFG me abra las puertas en las empresas que implementen brazos robóticas en su día a día.

Además, en un futuro inmediato, pretendo asentar los conocimientos necesarios para poder crear un prototipo de piernas robóticas, orientado a personas con un cierto grado severo de minusvalía física. Por otro lado, me gustaría profundizar en mi conocimiento de lenguajes de programación como *C* y *C++* con Arduino y los distintos programas a desarrollar, y así mismo en *MATLAB*.

3 Conceptos teóricos

Se pretende ir construyendo el conocimiento necesario para entender la problemática a la hora de modelar cualquier tipo de elementos robóticos, y por tanto, se comenzará por los elementos más básicos.

Así, se iniciará por explicar los Sistemas de Ejes de Coordenadas y la Regla de la mano Derecha para caracterizar el espacio en el que desarrollar la teoría a aplicar y los posteriores cálculos.

Una vez entendido el espacio, se incorporarán las Razones Trigonométricas, las Teorías del Seno y del Coseno y las Ecuaciones Trascendentes (basadas en elementos logarítmicos, exponenciales y trigonométricos). Y estos recursos matemáticos servirán para operar con las matrices e igualdades encontradas en su tratamiento para determinar las ecuaciones que caracterizarán los modelos robóticos.

Se presentarán los diferentes tipos de articulaciones, como consecuencia los tipos de robots de los que se va a tratar en el resto del TFG.

A continuación se explicarán dos conceptos fundamentales, la cinemática Directa y la Cinemática Inversa, fundamentales en robótica para entender la problemática a resolver.

Relacionado con la Cinemática Directa, se introducirá el Algoritmo de Denavit-Hartenberg (basado en las matrices de Transformaciones Homogéneas), por el que se podrá modelar cualquier brazo robótico a través de una serie de parámetros que determinan giros y traslaciones en el espacio entre los diferentes Sistemas de Ejes de Coordenadas de cada Articulación. Así, se podrá caracterizar un brazo robótico a través de una serie de ecuaciones para calcular el punto alcanzado con su extremo, dados los ángulos de sus articulaciones.

Se introducirá cómo se puede relacionar este modelado de la Cinemática Directa, basado en el Algoritmo de Denavit-Hartenberg, con el modelado de la Cinemática Inversa, a través de una metodología, por la que se predetermina un punto a alcanzar y se utiliza una serie de recursos matemáticos (producto de matrices, inversa de matrices, sumas de cuadrados de sus elementos, producto vectorial de vectores unitarios, etc.), para determinar las ecuaciones por las que se calculan los ángulos de las articulaciones del brazo robótico (valores articulares).

Un aspecto importante que se va a presentar es el Desacoplamiento. Y en realidad es un recurso ya utilizado cuando se define un modelo robótico con varias articulaciones de las que, las tres primeras son para acercarse al punto deseado, y las siguientes para ser usadas como herramientas, por ejemplo una cierta rotación para orientar una pinza de forma correcta y que pueda ser cerrada sobre un objeto.

Pero en este caso, se usará el Desacoplamiento para dividir un problema de un número indeterminado de articulaciones, y que puede llegar a ser elevado, en subproblemas más fáciles de resolver, de tal forma que el extremo de un primer grupo de articulaciones sea, precisamente, la base del segundo grupo de articulaciones. Y así, sucesivamente, hasta alcanzar el punto final. Lo interesante de este TFG es que, en este caso, solo se habrá tratado de resolver el acercamiento al punto deseado, para la que se habrá llegado a una solución mediante el uso de una metodología sistematizada.

Por último, se generará una metodología propia con la que caracterizar cualquier tipo de brazo robótico.

3.1 Sistemas de coordenadas y la regla de la mano derecha

Un sistema de coordenadas es un sistema de referencia. Habitualmente se trabaja en sistemas de dos o tres dimensiones ya que es lo más visual y habitual en la naturaleza. Un sistema de coordenadas está compuesto por ejes. Para saber como se sitúan dichos ejes, se aplica la regla de la mano derecha. [WWWTeklaDoc0] Los distintos ejes no basta con que se sitúen a 90° entre sí, sino que tienen que estar en una posición concreta. Esta posición la podemos determinar situando la mano de la siguiente manera (dextrógiro):



Ilustración 1: Regla de la mano derecha
[WWWjsvanegascDoc1]

Se debe tener cuidado ya que si se realiza con la mano izquierda, el giro será levógiro en lugar de dextrógiro. A continuación se puede ver la diferencia de hacerlo con la mano izquierda (sistema de coordenadas 1), y hacerlo correctamente con la derecha (sistema de coordenadas 2).

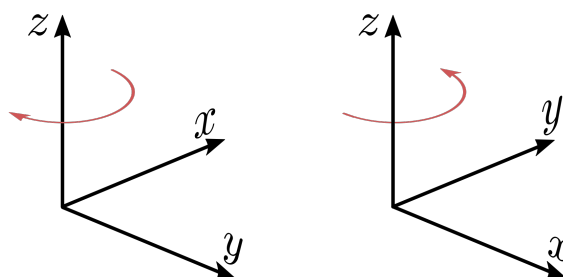


Ilustración 2: Levógiro y dextrógiro [WWWjsvanegascDoc1]

Todos los puntos del sistema vendrán determinados por tantos valores, como dimensiones haya. Dependiendo de dichos valores, el punto se encontrará en un cuadrante, o en otro. En el caso de que haya tres dimensiones (espacio tridimensional), habrá 8 cuadrantes. Las coordenadas que indican un punto pueden ser tanto positivas como negativas, dependiendo de los cuadrantes en los que se encuentre el punto.

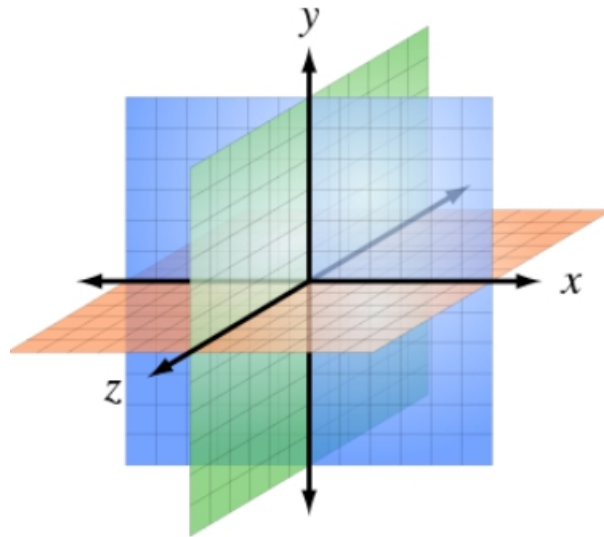


Ilustración 3: Cuadrante sistema de coordenadas
[WWWwikipediaDoc0]

3.2 Razones trigonométricas

A lo largo de este TFG se utilizan de forma recurrente distintas razones trigonométricas. [WWWwebs.ucmDoc0][WWWsuperprofDoc0][WWWlaplace.usDoc0][WWWwikipediaDoc18]

- $\cos^2(\alpha) + \sin^2(\beta) = 1$
- Ángulos Suplementarios:
 $\beta = 180 - \alpha \Leftrightarrow \sin(\alpha) = \sin(\beta) ; \cos(\alpha) = -\cos(\beta) ; \operatorname{tg}(\alpha) = -\operatorname{tg}(\beta) \dots$
 $\dots ; \operatorname{tg}(\beta) = -\operatorname{cotg}(\alpha)$

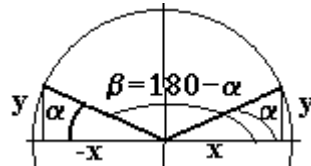


Ilustración 4: Ángulos Suplementarios ($\beta = 180 - \alpha$)

- Ángulos Complementarios: $\beta = 90 - \alpha \Leftrightarrow \operatorname{tg}(\beta) = \operatorname{cotg}(\alpha)$

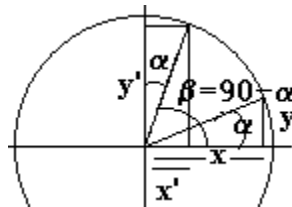


Ilustración 5: Ángulos Complementarios ($\beta = 90 - \alpha$)

- Ángulos que difieren en 180° : $\beta = 180 + \alpha \Leftrightarrow \operatorname{tg}(\beta) = \operatorname{tg}(\alpha)$

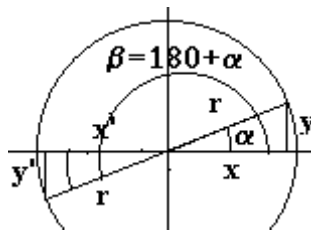


Ilustración 6: Ángulos que difieren en 180° ($\beta = 180 + \alpha$)

- Ángulos opuestos:
 $\beta = -\alpha \Leftrightarrow \operatorname{tg}(\beta) = -\operatorname{tg}(\alpha) ; \cos(-\alpha) = \cos(\alpha) ; \sin(-\alpha) = -\sin(\alpha) \dots$
 $\dots ; \sin(-\alpha) = -\sin(\alpha)$

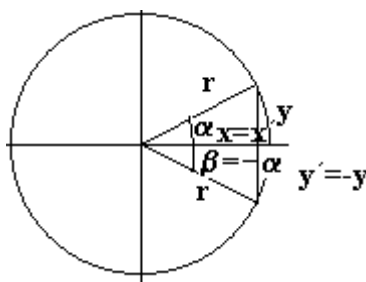


Ilustración 7: Ángulos opuestos
($\beta = -\alpha$)

- Coseno de la suma: $\cos(\alpha + \beta) = \cos(\alpha)\cos(\beta) - \sin(\alpha)\sin(\beta)$

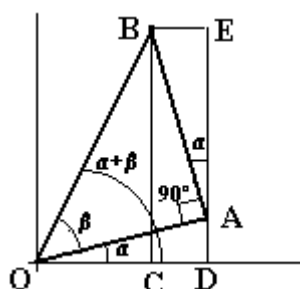


Ilustración 8: Coseno de la suma: $(\cos(\alpha + \beta))$

- Coseno de la diferencia: $\cos(\alpha - \beta) = \cos(\alpha)\cos(\beta) + \sin(\alpha)\sin(\beta)$
- Seno de la suma: $\sin(\alpha + \beta) = \sin(\alpha)\cos(\beta) + \cos(\alpha)\sin(\beta)$
- Seno de la diferencia: $\sin(\alpha - \beta) = \sin(\alpha)\cos(\beta) - \cos(\alpha)\sin(\beta)$
- Tangente de la suma: $\operatorname{tg}(\alpha + \beta) = (\operatorname{tg}(\alpha) + \operatorname{tg}(\beta)) / (1 - \operatorname{tg}(\alpha)\operatorname{tg}(\beta))$
- Tangente de la diferencia: $\operatorname{tg}(\alpha - \beta) = (\operatorname{tg}(\alpha) - \operatorname{tg}(\beta)) / (1 + \operatorname{tg}(\alpha)\operatorname{tg}(\beta))$
- Coseno del ángulo doble ($\alpha = \beta$ en el coseno de la suma):
 $\alpha = \beta \Leftrightarrow \cos(\alpha + \beta) = \cos(2\alpha) = \cos^2(\alpha) - \sin^2(\alpha)$
- Coseno del ángulo triple ($\alpha = \beta = C$ en el coseno de la suma):
 $\alpha = \beta = C \Leftrightarrow \cos(\alpha + \beta + C) = \cos(3\alpha) = 4\cos^3(\alpha) - 3\cos(\alpha)$
- Seno del ángulo doble ($\alpha = \beta$ en el seno de la suma):
 $\alpha = \beta \Leftrightarrow \sin(\alpha + \beta) = \sin(2\alpha) = 2\sin(\alpha)\cos(\alpha)$
- Seno del ángulo triple ($\alpha = \beta = C$ en el seno de la suma):
 $\alpha = \beta = C \Leftrightarrow \sin(\alpha + \beta + C) = \sin(3\alpha) = 3\sin(\alpha) - 4\sin^3(\alpha)$
- Tangente del ángulo doble ($\alpha = \beta$ en la tg de la suma):
 $\alpha = \beta \Leftrightarrow \operatorname{tg}(\alpha + \beta) = (\operatorname{tg}(\alpha) + \operatorname{tg}(\beta)) / (1 - \operatorname{tg}(\alpha)\operatorname{tg}(\beta)) = 2\operatorname{tg}(\alpha) / (1 - \operatorname{tg}^2(\alpha))$
- Tangente del ángulo triple ($\alpha = \beta = C$ en la tg de la suma):
 $\alpha = \beta = C \Leftrightarrow \operatorname{tg}(\alpha + \beta + C) = (3\operatorname{tg}(\alpha) - \operatorname{tg}^3(\alpha)) / (1 - 3\operatorname{tg}^2(\alpha))$
- Coseno del ángulo mitad ($\alpha = \alpha/2$ en el coseno del ángulo doble):
 $\alpha = \alpha/2 \Leftrightarrow \cos(\alpha/2) = \sqrt{[(1 + \cos(\alpha))/2]}$
- Seno del ángulo mitad ($\alpha = \alpha/2$ en el seno del ángulo doble):
 $\alpha = \alpha/2 \Leftrightarrow \sin(\alpha/2) = \sqrt{[(1 - \cos(\alpha))/2]}$
- Tangente del ángulo mitad ($\alpha = \alpha/2$ en el coseno y seno del ángulo doble):
 $\alpha = \alpha/2 \Leftrightarrow \operatorname{tg}(\alpha/2) = \sin(\alpha/2) / \cos(\alpha/2) = \sqrt{[(1 - \cos(\alpha)) / (1 + \cos(\alpha))]}$
- Suma de cosenos: $\cos(\alpha) + \cos(\beta) = 2\cos[(\alpha + \beta)/2]\cos[(\alpha - \beta)/2]$

- Diferencia de cosenos: $\cos(\alpha) - \cos(\beta) = -2 \sin[(\alpha + \beta)/2] \sin[(\alpha - \beta)/2]$
- Suma de senos: $\sin(\alpha) + \sin(\beta) = 2 \sin[(\alpha + \beta)/2] \cos[(\alpha - \beta)/2]$
- Diferencia de senos: $\sin(\alpha) - \sin(\beta) = 2 \cos[(\alpha + \beta)/2] \sin[(\alpha - \beta)/2]$
- $\cos(\alpha) \cos(\beta) = (\cos(\alpha - \beta) + \cos(\alpha + \beta))/2$
- $\sin(\alpha) \sin(\beta) = (\cos(\alpha - \beta) - \cos(\alpha + \beta))/2$
- $\sin(\alpha) \cos(\beta) = (\sin(\alpha + \beta) + \sin(\alpha - \beta))/2$
- $\cos(\alpha) \sin(\beta) = (\sin(\alpha + \beta) - \sin(\alpha - \beta))/2$
- $\tan(\alpha) \tan(\beta) = (\cos(\alpha - \beta) - \cos(\alpha + \beta)) / (\cos(\alpha - \beta) + \cos(\alpha + \beta))$
- $\sin(\alpha) \pm \sin(\beta) = 2 (\sin((\alpha \pm \beta)/2) \cos((\alpha \mp \beta)/2))$
- $\cos(\alpha) + \cos(\beta) = 2 (\cos((\alpha + \beta)/2) \cos((\alpha - \beta)/2))$
- $\cos(\alpha) - \cos(\beta) = -2 (\sin((\alpha + \beta)/2) \sin((\alpha - \beta)/2))$
- $\sin(\alpha) \pm \cos(\beta) = 2 (\sin((\alpha \pm \beta)/2) \cos((\alpha \mp \beta)/2))$
- $\sin(\arcsin(\alpha)) = \alpha$
- $\cos(\arcsin(\alpha)) = \sqrt{1 - \alpha^2}$
- $\tan(\arcsin(\alpha)) = \alpha / \sqrt{1 - \alpha^2}$
- $\sin(\arccos(\alpha)) = \sqrt{1 - \alpha^2}$
- $\cos(\arccos(\alpha)) = \alpha$
- $\tan(\arccos(\alpha)) = \sqrt{1 - \alpha^2} / \alpha$
- $\sin(\arctan(\alpha)) = \alpha / \sqrt{1 + \alpha^2}$
- $\cos(\arctan(\alpha)) = 1 / \sqrt{1 + \alpha^2}$
- $\tan(\arctan(\alpha)) = \alpha$
- $\sin(\alpha + \beta) + \sin(\alpha - \beta) = 2 \sin(\alpha) \cos(\beta)$
- $\sin(\alpha + \beta) - \sin(\alpha - \beta) = 2 \cos(\alpha) \sin(\beta)$
- $1 + \cos(\alpha) = (2 \cos(\alpha/2))^2$
- $1 - \cos(\alpha) = (2 \sin(\alpha/2))^2$

No hay que olvidar la posibilidad de tratar estas ecuaciones como ecuaciones polinómicas del tipo:

$$ax^2 + bx = c$$

Donde la solución sería:

$$x = (-b \pm \sqrt{b^2 - 4ac}) / 2a$$

3.3 Teoremas de cos y sin

Los teoremas de los senos y cosenos son:

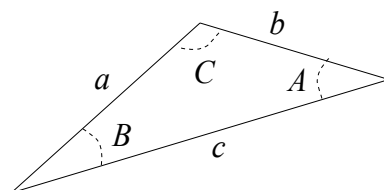


Ilustración 9: Teoremas del seno y del coseno

- $a^2 = b^2 + c^2 - 2 b c \cos(A)$ Teorema del coseno
- $b^2 = a^2 + c^2 - 2 a c \cos(B)$
- $c^2 = a^2 + b^2 - 2 a b \cos(C)$
- $a/\sin(A) = b/\sin(B) = c/\sin(C) = 2 \text{ Radio}$ Teorema del seno
- $(a+b)/(a-b) = \operatorname{tg}((A+B)/2)/\operatorname{tg}((A-B)/2)$
- $(a+c)/(a-c) = \operatorname{tg}((A+C)/2)/\operatorname{tg}((A-C)/2)$
- $(b+c)/(b-c) = \operatorname{tg}((B+C)/2)/\operatorname{tg}((B-C)/2)$

3.4 Ecuaciones trascendentes

Las ecuaciones trascendentes son un tipo de ecuaciones que no pueden reducirse a una ecuación polinómica, de la forma $f(x)=0$, para resolver mediante operaciones algebraicas y que, por tanto, no pueden solucionarse fácilmente con sumas, restas, multiplicaciones o divisiones.

Ecuaciones trascendentes a aplicar como soluciones de las variables articulares tras usar diferentes razones trigonométricas sobre las ecuaciones angulares representativas de los brazos robóticos [WWWyoutubeDoc84]:

- $\sin \theta = a \Leftrightarrow \theta = \operatorname{atan2}(a, \pm \sqrt{1-a^2})$ (2 soluciones)
- $\cos \theta = b \Leftrightarrow \theta = \pm \operatorname{atan2}(\sqrt{1-b^2}, b)$ (2 soluciones)
- $\sin \theta = a$ y $\cos \theta = b \Leftrightarrow \theta = \operatorname{atan2}(a, b)$ (1 solución)
- $a \cos \theta + b \sin \theta = 0 \Leftrightarrow \theta = \operatorname{atan2}(a, -b)$ y $\theta = \operatorname{atan2}(-a, b)$ (2 soluciones)
- $a \cos \theta + b \sin \theta = c \Leftrightarrow \theta = \operatorname{atan2}(b, a) \pm \operatorname{atan2}(\sqrt{a^2 + b^2 - c^2}, c)$ (2 soluciones)
- $a \cos \theta - b \sin \theta = c$ y $a \sin \theta + b \cos \theta = d \Leftrightarrow \theta = \operatorname{atan2}(d, c) - \operatorname{atan2}(b, a)$ (1 solución)
- $a \cos \theta + b \sin \theta = c$ y $d \sin \theta + e \cos \theta = f \Leftrightarrow \theta = \operatorname{atan2}(af - ce, cd - bf)$ (1 solución)

3.4.1 Un caso de ecuación trascendente: $P_x \sin(\theta) + P_y \cos(\theta) = \text{CTE}$

Como ejemplo, en este apartado se puede ver la resolución matemática de una de las ecuaciones trascendentes haciendo uso, también, de la interpretación geométrica del problema en el esquema planteado.

Por ejemplo, para el caso: $(-S\theta_1)P_x + C\theta_1 P_y = L_2$ (cuya solución vendrá dada por la expresión: $a \cos \theta + b \sin \theta = c \Leftrightarrow \theta = \operatorname{atan2}(b, a) \pm \operatorname{atan2}(\sqrt{a^2 + b^2 - c^2}, c)$)

Se hacen las siguientes sustituciones trigonométricas:

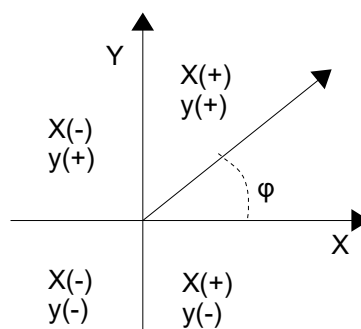


Ilustración 10: Brazo con dos motores perpendiculares

$$\begin{aligned}
P_x &= r \cos(\varphi) \\
P_y &= r \sin(\varphi) \\
r &= \pm (P_x^2 + P_y^2)^{1/2} \\
\varphi &= \text{atan}(P_y/P_x) = \text{atan}(P_x, P_y) = \text{atan2}(P_y, P_x)
\end{aligned}$$

$$\begin{aligned}
(-S\theta_1)P_x + C\theta_1 P_y &= d_2 = L_2 \\
(-S\theta_1)r \cos(\varphi) + C\theta_1 r \sin(\varphi) &= d_2 \\
C\theta_1 r \sin(\varphi) - S\theta_1 r \cos(\varphi) &= d_2 \\
\sin(\varphi) \cos(\theta_1) - \cos(\varphi) \sin(\theta_1) &= d_2/r \quad (0 < d_2/r \leq 1) \\
\sin(\varphi - \theta_1) &= d_2/r \quad (0 < d_2/r \leq 1)
\end{aligned}$$

$$\begin{aligned}
\cos^2(a) \sin^2(a) &= 1 \\
\cos(\varphi - \theta_1) &= \pm (1 - (d_2/r)^2)^{1/2} \\
\sin(\varphi - \theta_1)/\cos(\varphi - \theta_1) &= \tan(\varphi - \theta_1) = (d_2/r)/(\pm (1 - (d_2/r)^2)^{1/2}) \quad (0 < d_2/r \leq 1) \\
\varphi - \theta_1 &= \text{atan2}(\pm (1 - (d_2/r)^2)^{1/2}, d_2/r) \\
\theta_1 &= \varphi - \text{atan2}(\pm (1 - (d_2/r)^2)^{1/2}, d_2/r)
\end{aligned}$$

Dos soluciones:

$$\begin{aligned}
\theta_1 &= \text{atan}(P_x, P_y) - \text{atan}((d_2/r), \pm (1 - (d_2/r)^2)^{1/2}) \\
\theta_1 &= \text{atan2}(P_y, P_x) - \text{atan2}((\pm (1 - (d_2/r)^2)^{1/2}, d_2/r))
\end{aligned}$$

3.5 Tipos de articulaciones

Existen varios tipos de brazos robóticos dependiendo de sus articulaciones. Cada uno con algunas características concretas que se ven reflejadas a la hora de aplicar el algoritmo de Denavit-Hartenberg. Los tipos de robot principales son robot cartesiano, robot cilíndrico, robot esférico o polar, robot angular o antropomórfico y robot *SCARA*.

Dependiendo de los robots y los posibles desplazamientos, cada uno tiene una serie de grados de libertad como se indica a continuación:

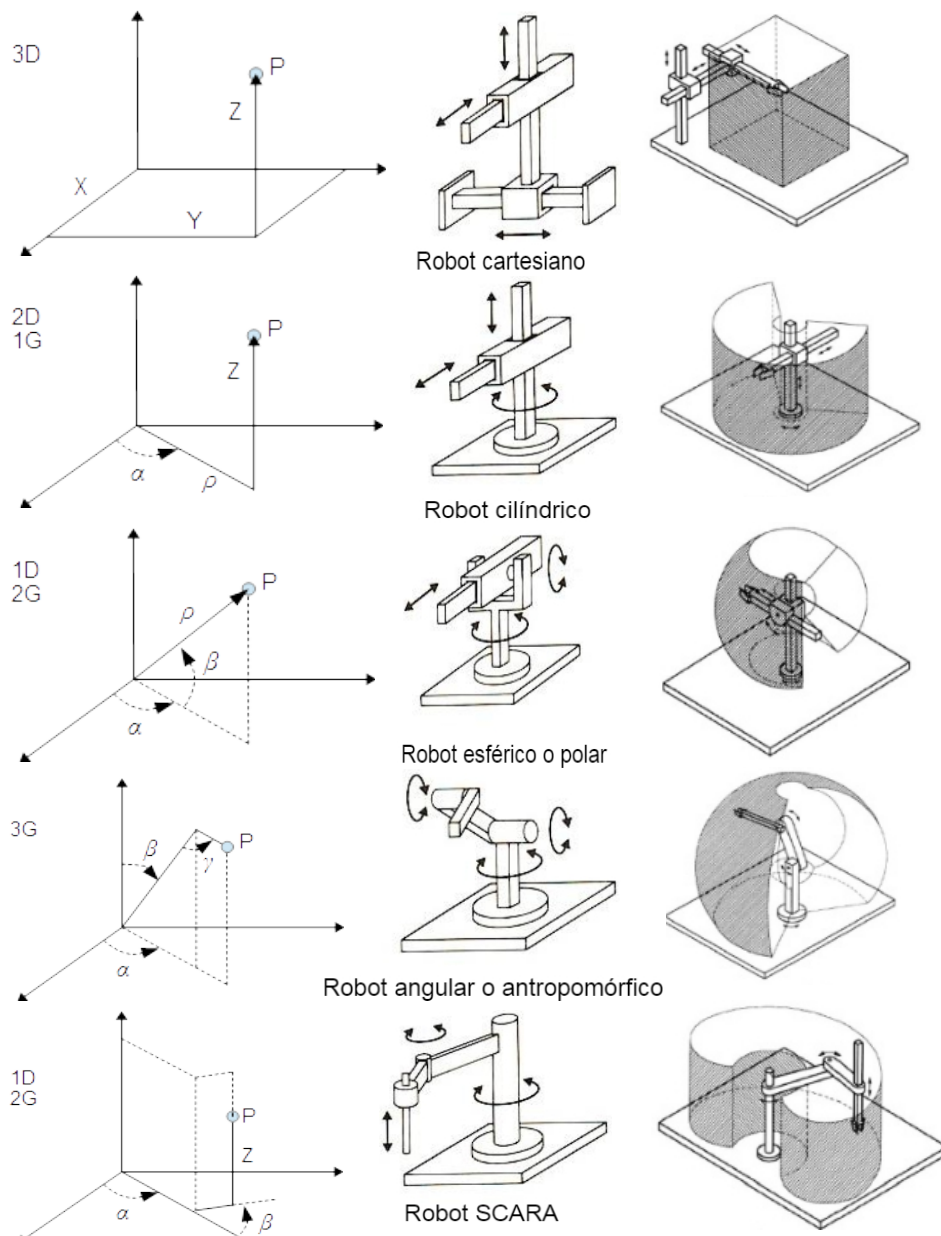


Ilustración 11: Tipos de robots básicos (Centro: [WWWdocplayerDoc0] - Derecha: [WWWslideshareDoc2])

Dependiendo de la necesidad y la funcionalidad de cada brazo en concreto, se deseará utilizar un tipo u otro.

Existen a su vez, varios tipos de articulaciones para los motores dependiendo de las necesidades. A continuación se muestran algunos tipos.

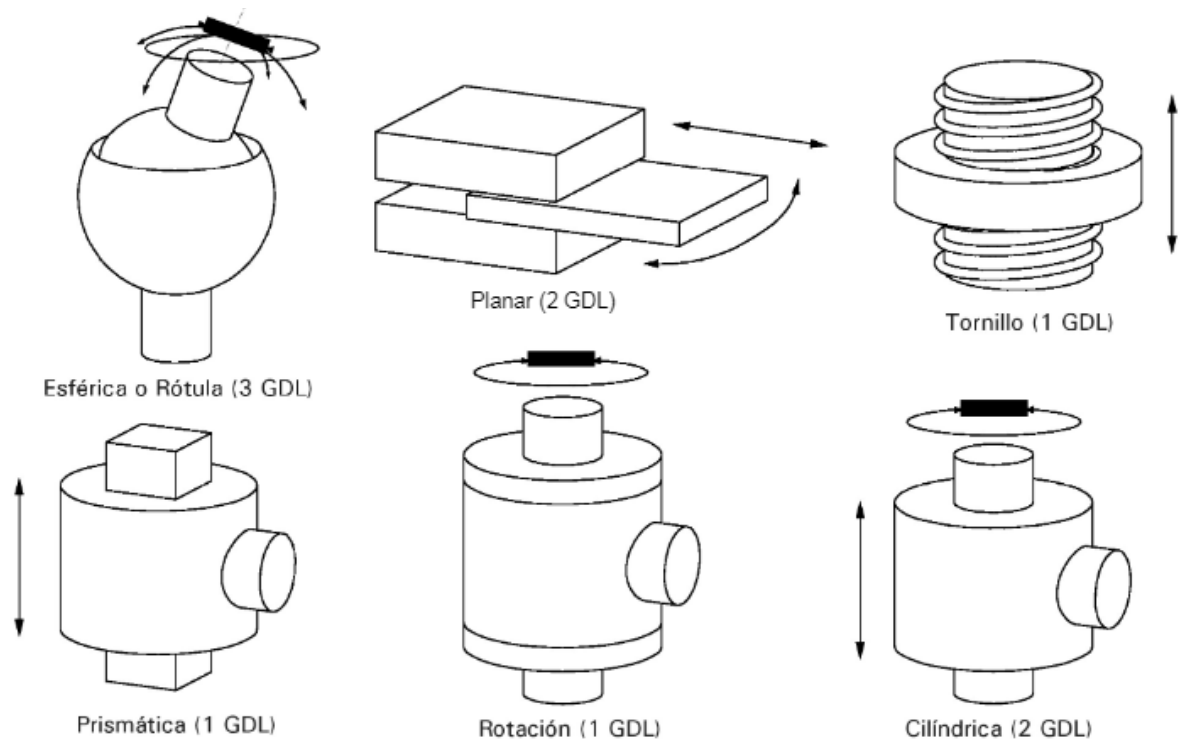


Ilustración 12: Tipos de articulaciones [WWWdocplayerDoc1]

Destacar por ejemplo que aquellos motores cuyo movimiento es rectilíneo y no circular, no tendrán una θ variable, sino que será su componente r o d .

3.6 Cinemática

La cinemática es la ciencia que estudia el movimiento, sin tener en cuenta las fuerzas que puedan ser aplicadas y que generan dicho movimiento. En el momento de análisis, y por tanto, de cálculo, solo importa la posición entre los distintos motores en posiciones de reposo. Por tanto, si un brazo robótico está en movimiento, se considerará tan solo un instante asumiendo que el brazo no tiene ningún movimiento y que no hay ninguna fuerza sobre él.

3.6.1 Cinemática Directa

La cinemática directa consiste en calcular el punto final que alcanza un brazo robótico conociendo los ángulos que forma cada una de las articulaciones.

A continuación se muestra un ejemplo simple de cinemática directa.

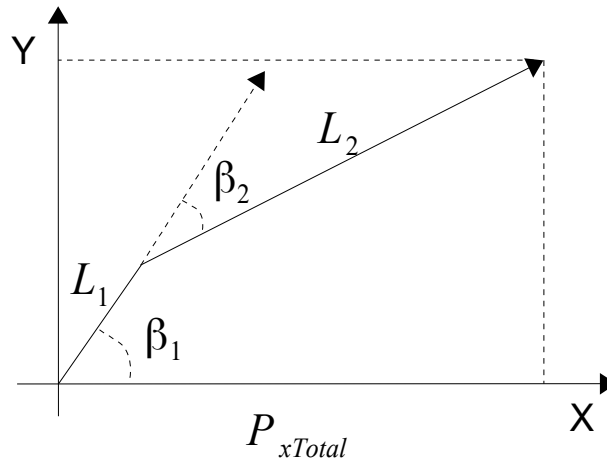


Ilustración 13: Ejemplo cinemática directa

Conociendo los ángulos que forman β_1 y β_2 , mediante cinemática directa se debe sacar el punto $P(x, y)$ exacto sobre el que se encuentra el extremo del brazo robótico.

3.6.2 Cinemática Inversa

La cinemática inversa consiste en calcular los ángulos correspondientes a los motores de un brazo robótico para alcanzar un punto predeterminado. Este cálculo es más complejo que la cinemática directa ya que es común que haya casos con múltiples incógnitas y soluciones, mientras que en la cinemática directa sólo hay una solución posible (el punto que se alcanza).

A continuación se muestra un ejemplo simple de cinemática inversa.

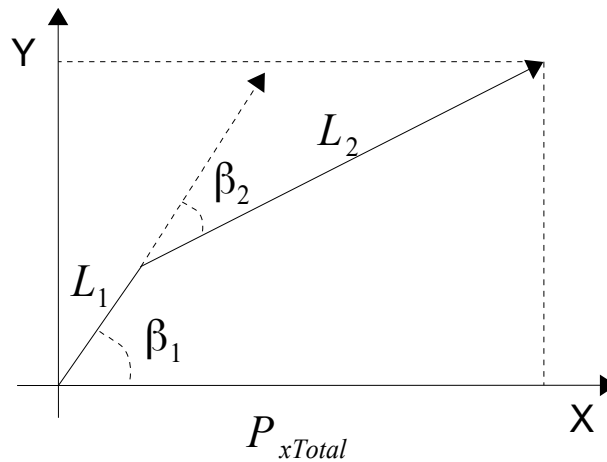


Ilustración 14: Ejemplo cinemática inversa

Conociendo el punto final $P(x, y)$ y siendo β_1 y β_2 dos incógnitas, a través de la cinemática inversa se sacan los valores de β_1 y β_2 (variables articulares).

3.7 Denavit-Hartenberg

Los brazos robóticos simples, pueden ser calculados trigonométricamente de forma sencilla. A medida que va aumentando el número de articulaciones (dependiendo también de cómo se posicionen entre sí), el problema se complica enormemente. Por tanto, con un brazo de tan sólo 4 motores ya tendría una complejidad notablemente más elevada que uno de 2, e incluso de 3 motores. El algoritmo de Denavit-Hartenberg proporciona una metodología que ayuda a resolver, tanto los problemas de cinemática directa, como de cinemática inversa.

Según el Algoritmo de Matrices de Transformación Homogénea de *Denavit-Hartenberg* se crean 4 matrices por cada articulación, dos de rotación y dos de traslación. Dichas matrices representan los 4 movimientos que se necesitan para llegar de un Sistema de Coordenadas (Sistema Cartesiano) a otro. Esos 4 movimientos, como se puede ver a continuación, constan de una rotación sobre el eje Z actual (para hacer coincidir el eje X actual con el siguiente), una traslación sobre el eje Z actual (para que ambas Z estén en el mismo plano), una traslación sobre X (para que ambas X estén en el mismo plano) y una rotación sobre el eje X actual, que ya coincide con el eje X siguiente (para que coincidan los ejes Z).

$$R_{z, \theta_i} Trans_{z, d_i} Trans_{x, a_i} R_{x, \alpha_i} = {}^{i-1}A_i =$$

$$\begin{pmatrix} C\theta_i & -S\theta_i & 0 & 0 \\ S\theta_i & C\theta_i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & C\alpha_i & -S\alpha_i & 0 \\ 0 & S\alpha_i & C\alpha_i & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} =$$

$$\begin{pmatrix} C\theta_i & -C\alpha_i S\theta_i & S\alpha_i S\theta_i & a_i C\theta_i \\ S\theta_i & C\alpha_i C\theta_i & -S\alpha_i C\theta_i & a_i S\theta_i \\ 0 & S\alpha_i & C\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

La matriz resultante (4x4) del producto de estas cuatro matrices define cada una de las articulaciones de un brazo robótico.

No obstante, conociendo los valores de algunos parámetros, se debe sustituir para facilitar el cálculo. En muchos casos α tiene valores de 0° o de 90° que simplifican de manera considerable las operaciones.

Una vez obtenidas las matrices de cada articulación, se multiplican empezando por la base hacia la punta del brazo robótico las matrices resultantes de cada una de las articulaciones. En un ejemplo de dos motores la matriz final sería: $T = {}^0A_1 {}^1A_2 = {}^0A_2$.

Una vez conocida la matriz que regirá el movimiento del brazo robótico con el que se esté trabajando, es necesario calcular los parámetros de DH. Estos parámetros relacionan las distintas articulaciones del problema, indicando cuál es la disposición entre sí. Para calcular estos parámetros el algoritmo define una serie de pasos:

- D-H 1. Numerar los eslabones comenzando con 1 (primer eslabón móvil de la cadena (E_1)) y acabando con n (último eslabón móvil). Se numerará como eslabón 0 (E_0) a la base fija del

robot.

- D-H 2. Numerar cada articulación comenzando por 1 (la correspondiente al primer grado de libertad (A_1)) y acabando en n (A_n).
- D-H 3. Localizar el eje de cada articulación. Si esta 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.
- D-H 4. Para i de 0 a $n-1$, situar el eje z_i sobre el eje de la articulación $i+1$.
- D-H 5. Situar el origen del sistema de la base $\{S_0\}$ (x_0, y_0, z_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 .
- D-H 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 $i+1$ (cualquier posición en el eje z_{i+1}).
- D-H 7. Situar x_i en la línea normal común a z_{i-1} y z_i .
- D-H 8. Situar y_i de modo que forme un sistema dextrógiro x_i y z_i .
- D-H 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 .
- D-H 10. Obtener θ_i como el ángulo que hay que girar en torno a z_{i-1} para que x_{i-1} y x_i queden paralelos.
- D-H 11. Obtener d_i como la distancia, medida a lo largo del eje z_{i-1} , que habría que desplazar $\{S_{i-1}\}$ para que x_i y x_{i-1} quedasen alineados.
- D-H 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 $\{S_i\}$.
- D-H 13. Obtener α_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 $\{S_i\}$.
- D-H 14. Obtener las matrices de transformación ${}^{i-1}A_i$.
- D-H 15. Obtener la matriz de transformación que relaciona el sistema de la base con el extremo del robot $T = {}^0A_1 {}^1A_2 \dots {}^{n-1}A_n$.
- D-H 16. La matriz T define la orientación (submatriz de rotación) y posición (submatriz de traslación) del extremo referido a la base en función de las n coordenadas articulares.

i	θ_i	d_i	a_i	α_i
1	$\theta_1 - \pi/2$	d_1	0	0
2	θ_2	0	0	$\pi/2$
3	θ_3	d_3	0	0

Tabla 1: Ejemplo parámetros de Denavit-Hartenberg

Cuando se habla de giro sobre un eje, se refiere a lo siguiente:

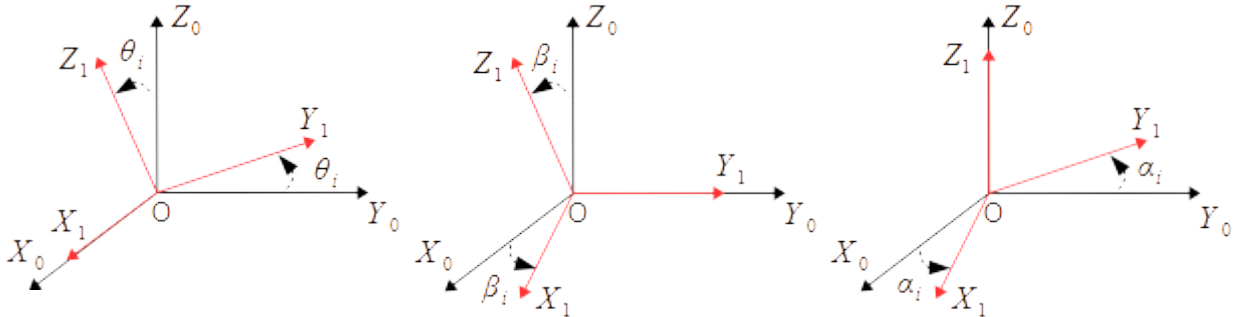


Ilustración 15: Rotaciones sobre los ejes de un sistema de coordenadas

Para saber si el giro es positivo o negativo, se deberá utilizar la regla de la mano derecha. Esta regla determina que si se agarra un eje de coordenadas situando el pulgar en el mismo sentido que el eje, si el giro va en la misma dirección que todos los dedos menos el pulgar, el giro será positivo. En caso contrario el giro será negativo.



Ilustración 16: Regla de la mano derecha

En el siguiente caso el giro sobre X es positivo ya que el sentido del movimiento es igual que el de los dedos.

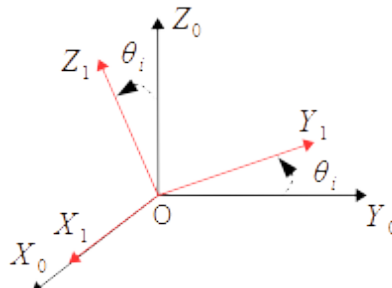


Ilustración 17: Giro sobre el eje x

Como ya se ha mencionado anteriormente, la matriz de Denavit-Hartenberg muestra la relación entre las distintas articulaciones. Es importante destacar que, pudiendo existir varios sistemas de referencia válidos y distintos, pueden existir varias Matrices de Transformación Homogéneas

(parámetros de DH) distintas pero correctas. Por ejemplo, en un mismo motor y colocando el primer eje indicado por los pasos de Denavit-Hartenberg, ya se encuentran dos posibilidades distintas.

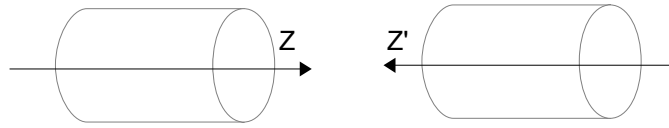


Ilustración 18: Ejemplo de dualidad de los ejes de dos motores

Otra forma de calcular los parámetros de DH es mediante el método modificado de John J. Craig. Este método es igualmente correcto, solo que alguna de las premisas de las que parte son distintas.

PARÁMETRO D-H	MÉTODO CLÁSICO	MÉTODO MODIFICADO (John J. Craig)
Ejes de las articulaciones	Z_{i-1} es para cada articulación i	Z_i es para cada articulación i
Longitud del eslabón (a_i)	Distancia entre O_i a la intersección de Z_{i-1} y el eje X_i a lo largo de X_i	Distancia de Z_i a Z_{i+1} medida a lo largo de X_i
Ángulo de torsión (α_i)	Ángulo de Z_{i-1} a Z_i sobre el eje X_i	Ángulo entre Z_i a Z_{i+1} medido alrededor de X_i
Desplazamiento (d_i)	Distancia del origen del sistema $i-1$ a la intersección del eje Z_{i-1} con el eje X_i a lo largo del eje Z_{i-1}	Distancia de X_{i-1} a X_i a lo largo de Z_i
Ángulo (θ_i)	Ángulo entre el eje X_{i-1} y X_i alrededor del eje Z_{i-1}	Ángulo entre X_{i-1} y X_i medido alrededor de Z_i

Tabla 2: Estudio de una cadena cinemática utilizando los algoritmos de Denavit-Hartenberg y Craig modificado [WWWdocplayerDoc7]

3.7.1 Cinemática Directa a partir de DH

Una vez calculados los parámetros de DH, y teniendo la matriz de DH compuesta por todas las articulaciones, es sencillo encontrar el punto en el que se encuentra el extremo del brazo robótico.

Como se puede ver en la siguiente fórmula, la matriz de Denavit-Hartenberg nos aporta una gran cantidad de información, y no sólo el punto final.

$$P^0 = P_{(P_x^0, P_y^0, P_z^0)} = \begin{pmatrix} P_x^0 \\ P_y^0 \\ P_z^0 \\ 1 \end{pmatrix} = T \begin{pmatrix} P_u^1 \\ P_v^1 \\ P_w^1 \\ 1 \end{pmatrix} = \begin{pmatrix} n_x & o_x & a_x & P_{O'x} \\ n_y & o_y & a_y & P_{O'y} \\ n_z & o_z & a_z & P_{O'z} \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} P_u^1 \\ P_v^1 \\ P_w^1 \\ 1 \end{pmatrix}$$

A continuación se desglosa la información aportada.

La submatriz $M_{3 \times 3}$ indica la orientación con la que se ataca el punto alcanzado.

$$\begin{pmatrix} n_x & o_x & a_x \\ n_y & o_y & a_y \\ n_z & o_z & a_z \end{pmatrix}$$

Por otro lado, la submatriz $M_{3 \times 1}$ indica los puntos finales alcanzados respecto a el sistema de coordenadas inicial.

$$M_{3 \times 1} = \begin{pmatrix} P_{o'x} \\ P_{o'y} \\ P_{o'z} \end{pmatrix}$$

$P_{o'x}$ representa el valor de x , sobre el que se encuentra el punto, del sistema de coordenadas 0. $P_{o'y}$ y $P_{o'z}$ son los valores respectivamente de y y z .

Por tanto se puede afirmar, que el resultado tendrá la siguiente forma:

$$\begin{pmatrix} [R_3^0]_{3 \times 3} & [O_3^0]_{3 \times 1} \\ [0]_{1 \times 3} & [1]_{1 \times 1} \end{pmatrix}_{4 \times 4} = \begin{pmatrix} R_3^0 & O_3^0 \\ 0 & 1 \end{pmatrix}_{4 \times 4}$$

3.7.2 Cinemática Inversa a partir de DH

Así como es posible abordar el problema cinemático directo de una manera sistemática a partir de la utilización de matrices de transformación homogéneas, e independientemente de la configuración del robot, no ocurre lo mismo con el problema cinemático inverso, siendo el procedimiento de obtención de las ecuaciones fuertemente dependiente de la configuración del robot. Por tanto, la forma más fácil de resolver la cinemática inversa será determinar una relación matemática explícita (que podría ser calculada, por ejemplo, mediante *MATLAB*) como:

$q_k = f_k(x, y, z, \alpha, \beta, \theta, \dots)$ (Hasta n variables geométricas, o articulares)

$k = 1, \dots, n$ (Grados de Libertad – GdL, o DoF)

Esta solución permitirá que el problema cinemático inverso pueda resolverse en tiempo real, y además, en aplicación de otros modelos de cálculo actualmente utilizados, como puede ser una solución de tipo iterativo, no garantizaría tener la solución en el momento adecuado.

Por otra parte, la solución al problema cinemático inverso no tiene por qué ser única, existiendo diferentes duplas que posicionen y orienten el extremo del mismo modo. En estos casos, una solución de este tipo permite incluir determinadas reglas o restricciones (codo hacia arriba, etc.) que aseguren que la solución obtenida sea la más adecuada.

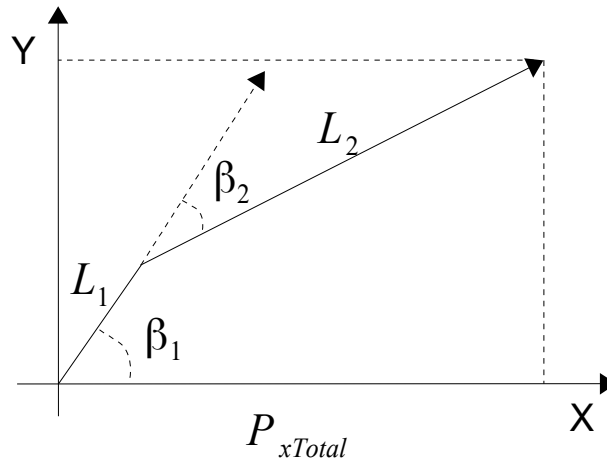


Ilustración 19: Ejemplo doble solución CI I

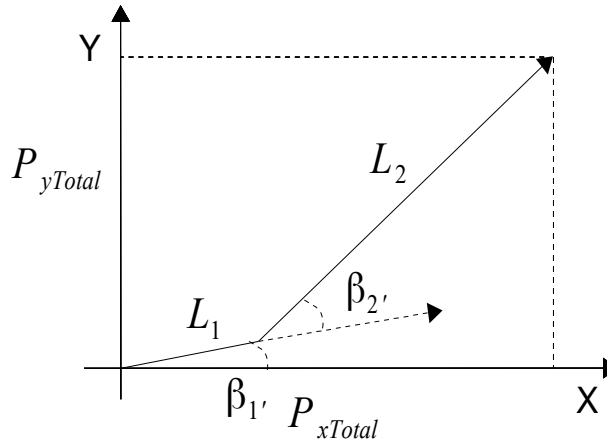


Ilustración 20: Ejemplo doble solución CI II

En este ejemplo si $|\beta_1| = |\beta_2|$ y $|\beta_2| = |\beta_1|$ el punto $p(x, y)$ será el mismo en ambos casos.

Así, se van a calcular las ecuaciones correspondientes al modelo cinemático inverso a partir de las matrices de Transformación Homogénea de Denavit-Hartenberg y sus parámetros, y utilizando sus matrices inversas.

Para ello, se partirá de la ecuación:

$$T = {}^0A_6 = {}^0A_1 {}^1A_2 {}^2A_3 {}^3A_4 {}^4A_5 {}^5A_6$$

En esta ecuación se realizarán varias fases. En cada una de ellas se invertirá un elemento y se pasará al otro lado de la igualdad, y a continuación, se realizará el producto de las matrices. Una vez hecho este cálculo, se analizarán todos los elementos de las matrices resultantes, haciendo uso de las diferentes razones trigonométricas para su simplificación, y finalmente, se escogerá aquel elemento de ambos lados de la ecuación que permita despejar una única variable [WWWdiag.uniroma1Doc16][WWWocw.ehu.eusDoc1]:

$$({}^0A_1)^{-1} T = {}^1A_2 {}^2A_3 {}^3A_4 {}^4A_5 {}^5A_6 = {}^1A_6 \text{ Se despeja } q_1.$$

$$({}^1A_2)^{-1} ({}^0A_1)^{-1} T = {}^2A_3 {}^3A_4 {}^4A_5 {}^5A_6 = {}^2A_6 \text{ Se despeja } q_2.$$

$$\begin{aligned}
({}^2A_3)^{-1}({}^1A_2)^{-1}({}^0A_1)^{-1}T &= {}^3A_4 {}^4A_5 {}^5A_6 = {}^3A_6 \quad \text{Se despeja } q_3. \\
({}^3A_4)^{-1}({}^2A_3)^{-1}({}^1A_2)^{-1}({}^0A_1)^{-1}T &= {}^4A_5 {}^5A_6 = {}^4A_6 \quad \text{Se despeja } q_4. \\
({}^4A_5)^{-1}({}^3A_4)^{-1}({}^2A_3)^{-1}({}^1A_2)^{-1}({}^0A_1)^{-1}T &= {}^5A_6 \quad \text{Se despeja } q_5 \text{ y } q_6.
\end{aligned}$$

Para ello, se usará la inversa, tal y como se desarrolla a continuación. [BookBarrientosDoc0]

Para ello, se partirá de:

$$A = \begin{pmatrix} [n]_{3 \times 1} & [o]_{3 \times 1} & [a]_{3 \times 1} & [p]_{3 \times 1} \\ [0]_{1 \times 1} & [0]_{1 \times 1} & [0]_{1 \times 1} & [1]_{1 \times 1} \end{pmatrix}_{4 \times 4} = \begin{pmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} [R]_{3 \times 3} & [p]_{3 \times 1} \\ [0]_{1 \times 3} & [1]_{1 \times 1} \end{pmatrix}_{4 \times 4}$$

Y se realizarán los siguientes cálculos:

$$\begin{aligned}
P^0 &= AP^1 = \begin{pmatrix} P^0 \\ 1 \end{pmatrix} = \begin{pmatrix} R_1^0 & O_1 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} P^1 \\ 1 \end{pmatrix} \\
P^0 &= R_1^0 P^1 + O_1 \\
(R_1^0)^{-1} P^0 &= (R_1^0)^{-1} R_1^0 P^1 + (R_1^0)^{-1} O_1
\end{aligned}$$

Aplicando el producto de matrices inversas entre sí $R^{-1}R=I$:

$$\begin{aligned}
((R_1^0)^{-1} R_1^0 = I) &\rightarrow (R_1^0)^{-1} P^0 = I P^1 + (R_1^0)^{-1} O_1 = P^1 + (R_1^0)^{-1} O_1 \\
P^1 &= (R_1^0)^{-1} P^0 - (R_1^0)^{-1} O_1
\end{aligned}$$

Aplicando ahora la propiedad de las matrices de rotación por la que $R^{-1}=R^T$ para matrices ortonormales ($A^{-1}=A^T$) de 3×3 [WWWyoutubeDoc76][WWWocw.ehu.eusDoc0] [BookBarrientosDoc0]:

$$\begin{aligned}
((R_1^0)^{-1} = (R_1^0)^T) &\rightarrow P^1 = (R_1^0)^T P^0 - (R_1^0)^T O_1 \\
\begin{pmatrix} P^1 \\ 1 \end{pmatrix} &= \begin{pmatrix} (R_1^0)^T & -(R_1^0)^T O_1 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} P^0 \\ 1 \end{pmatrix} = P^1 = A^{-1} P^0 \\
A^{-1} &= \begin{pmatrix} n_x & n_y & n_z & -n^T p \\ o_x & o_y & o_z & -o^T p \\ a_x & a_y & a_z & -a^T p \\ 0 & 0 & 0 & 1 \end{pmatrix}
\end{aligned}$$

Hay que considerar que la matriz A no es ortonormal, como sí lo es la matriz $R(3 \times 3)$. Por tanto, hay que considerar que la cuarta columna, sus elementos son el producto escalar del vector \vec{n} por el vector \vec{p} , representado por el producto matricial $-n^T p$, y con el signo cambiado, el producto escalar del vector \vec{o} por el vector \vec{p} , representado por el producto matricial $-o^T p$, y con el signo cambiado y el producto escalar del vector \vec{a} por el vector \vec{p} , representado por el producto matricial $-a^T p$, y con el signo cambiado:

$$\begin{aligned}
-n^T p &= \begin{pmatrix} -n_x & -n_y & -n_z \end{pmatrix} \begin{pmatrix} p_x \\ p_y \\ p_z \end{pmatrix} \\
-o^T p &= \begin{pmatrix} -o_x & -o_y & -o_z \end{pmatrix} \begin{pmatrix} p_x \\ p_y \\ p_z \end{pmatrix} \\
-a^T p &= \begin{pmatrix} -a_x & -a_y & -a_z \end{pmatrix} \begin{pmatrix} p_x \\ p_y \\ p_z \end{pmatrix}
\end{aligned}$$

También se aplicarán las siguientes razones trigonométricas ya comentadas con anterioridad [WWWwebs.ucmDoc0]:

$$\begin{aligned}
C\theta_i C\theta_i + S\theta_i S\theta_i &= C^2\theta_i + S^2\theta_i = 1 \\
S(a-b) &= S(a)C(-b) + C(a)S(-b) = S(\theta_i)C(-\theta_i) + C(\theta_i)S(-\theta_i) = \\
S(\theta_i)C(\theta_i) - C(\theta_i)S(\theta_i) &= S(\theta_i - \theta_i) = 0
\end{aligned}$$

De esta forma, se calcula, y se simplifica el cálculo de la inversa para cada articulación [WWWkramirez.netDoc0]:

$$\begin{aligned}
{}^{i-1}A_i &= \begin{pmatrix} C\theta_i & -S\theta_i C\alpha_i & S\theta_i S\alpha_i & a_i C\theta_i \\ S\theta_i & C\theta_i C\alpha_i & -C\theta_i S\alpha_i & a_i S\theta_i \\ 0 & S\alpha_i & C\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{pmatrix} \\
({}^{i-1}A_i)^{-1} &= {}^iA_{i-1} = \begin{pmatrix} C\theta_i & S\theta_i & 0 & -a_i C\theta_i C\theta_i - a_i S\theta_i S\theta_i \\ -S\theta_i C\alpha_i & C\theta_i C\alpha_i & S\alpha_i & a_i S\theta_i C\theta_i C\alpha_i - a_i C\theta_i S\theta_i C\alpha_i - d_i S\alpha_i \\ S\theta_i S\alpha_i & -C\theta_i S\alpha_i & C\alpha_i & -a_i S\theta_i C\theta_i S\alpha_i + a_i C\theta_i S\theta_i S\alpha_i - d_i C\alpha_i \\ 0 & 0 & 0 & 1 \end{pmatrix} = \\
&= \begin{pmatrix} C\theta_i & S\theta_i & 0 & -a_i(C\theta_i C\theta_i + S\theta_i S\theta_i) \\ -S\theta_i C\alpha_i & C\theta_i C\alpha_i & S\alpha_i & a_i C\alpha_i(S\theta_i C\theta_i - C\theta_i S\theta_i) - d_i S\alpha_i \\ S\theta_i S\alpha_i & -C\theta_i S\alpha_i & C\alpha_i & -a_i S\alpha_i(S\theta_i C\theta_i - C\theta_i S\theta_i) - d_i C\alpha_i \\ 0 & 0 & 0 & 1 \end{pmatrix} = \\
&= \begin{pmatrix} C\theta_i & S\theta_i & 0 & -a_i \\ -C\alpha_i S\theta_i & C\alpha_i C\theta_i & S\alpha_i & -S\alpha_i d_i \\ S\alpha_i S\theta_i & -S\alpha_i C\theta_i & C\alpha_i & -C\alpha_i d_i \\ 0 & 0 & 0 & 1 \end{pmatrix}
\end{aligned}$$

Y una vez calculadas las inversas correspondientes a las matrices planteadas se podrán sustituir los datos conocidos que simplifiquen estas matrices, los parámetros D-H (d_i, a_i, \dots). Así, se llegaría a un conjunto de matrices desde las que se podrían obtener una serie de ecuaciones que permitieran calcular los datos de las coordenadas articulares para alcanzar el punto final con la orientación deseada ($\theta_i, \alpha_i, \dots$). Estas ecuaciones serían el resultado de revisar los elementos del lado derecho en

$({}^4A_5)^{-1}({}^3A_4)^{-1}({}^2A_3)^{-1}({}^1A_2)^{-1}({}^0A_1)^{-1}T = {}^5A_6$, encontrar aquellos que sean cero o constantes, e igualarlos con los elementos respectivos del lado izquierdo [WWWdlscibDoc0][WWWwag.caltech.eduDoc0][WWWocw.ehu.eusDoc1].

Por otra parte, puesto que el problema cinemático directo, resuelto a través de ecuaciones homogéneas, inicialmente contendría 12 ecuaciones, y se buscan solo 4 relaciones que incluyan las variables articulares (una por cada grado de libertad), existirán, necesariamente ciertas interdependencias entre las 12 expresiones. Además, la submatriz de rotación tiene 9 elementos, y por tanto 9 ecuaciones posibles. Sin embargo, habrá 3 ecuaciones menos por la necesaria relación de ortonormalidad de sus elementos, y otras 3 menos por la existencia de tres vectores unitarios con los que se opera para calcular las matrices homogéneas [WWWyoutubeDoc89]. Por tanto, quedan 3 variables independientes en esa submatriz que pueden ser evaluadas. Pero este resultado no será cierto en todos los casos porque dependerá de cada modelo en particular, por lo que, podría darse el caso de que existieran menos ecuaciones que variables a calcular, por ejemplo en el caso de tener tres motores con sus ejes de rotación paralelos (sin variable y por estar en línea).

En estas situaciones de interdependencia se deben encontrar maneras de ir asignando valores a algunas variables en base a criterios lógicos, para acabar teniendo ecuaciones con de resultados finitos, e incluso, con una solución única.

Para más detalles sobre la aplicación de este procedimiento se pueden ver diferentes ejemplos en la documentación técnica donde se han resuelto gran variedad de problemas.

3.8 Desacoplo en subproblemas

Cuando se van a resolver modelos de brazo robótico, resulta habitual encontrar modelos con muy bajo número de articulaciones. Y resulta razonable cuando se trata de posicionar el extremo del robot en un punto determinado porque basta con tres articulaciones para alcanzar cualquier punto en el espacio, siempre que la distancia lo permita y que la propia estructura del brazo no lo impida por el montaje del chasis.

Por otra parte, a veces se debe resolver un segundo problema añadido. En este caso, el trabajo con una herramienta en el extremo del brazo.

Si la herramienta es relativamente completa, podría contar con tres grados de libertad que permitan una gran maniobrabilidad. Eso supone tres nuevas articulaciones.

Sin embargo, a la hora de resolver el problema, el número de datos podría ser el mismo, las tres componentes del punto a alcanzar, por lo que, a priori, si no se cuenta con alguna otra condición previa, resulta imposible llegar a resolver el problema puesto que tiene que existir una interdependencia de tres variables articulares respecto a las otras tres. [WWWkramirez.netDoc0] [BookBarrientosDoc0][WWWyoutubeDoc77]

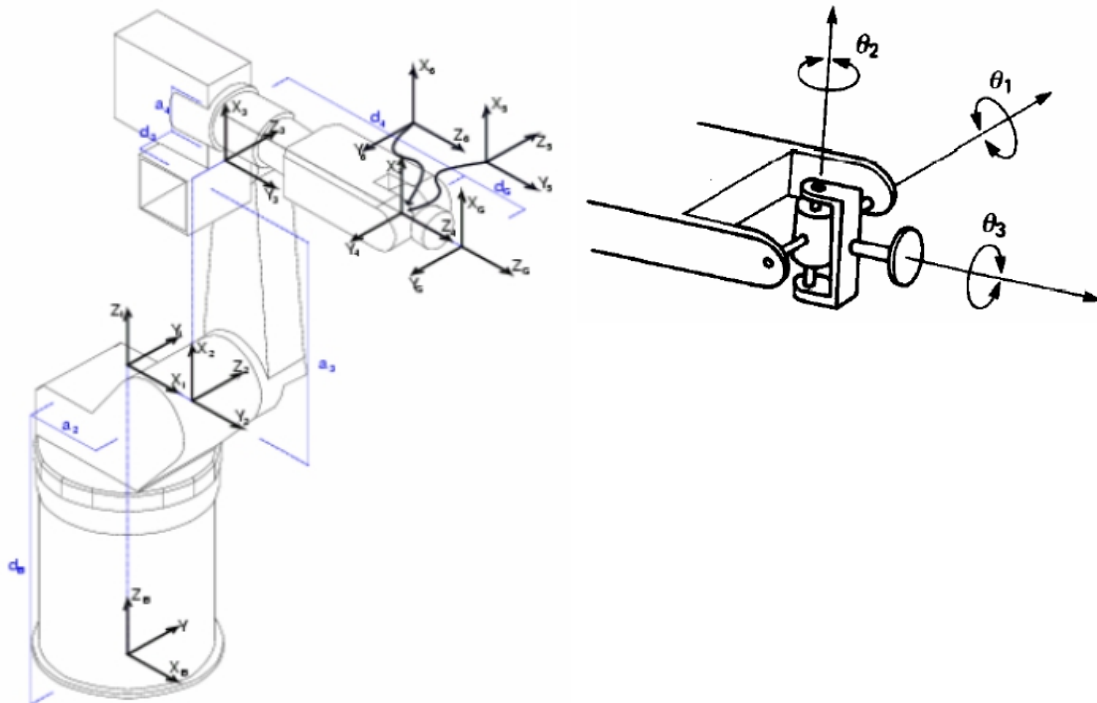


Ilustración 21: "Cinemática Inversa del Robot" [WWWkramirez.netDoc0]

Para resolver este problema, algunos autores han desarrollado una estrategia que permite evitar esa dependencia. Se trata de contar con tres motores para alcanzar el punto en el espacio, y otros tres motores para desarrollar la función de la herramienta, como se indica en la imagen.

Pero no todos los modelos permiten realizar las funciones deseadas con esta estrategia. Por ejemplo, si se quiere introducir la herramienta en un espacio cerrado y con cierta inclinación, los tres motores definidos no lo permitirán.

Para poder realizar esa función, se necesitan más articulaciones. Y para resolver ese problema, se vuelve a encontrar una interdependencia entre diferentes variables articulares.

4 Técnicas y herramientas

4.1 Repositorio

Hoy en día todo proyecto relacionado con el desarrollo de software debe tener un sistema de control de versiones. Estos sistemas de control de versiones son especialmente importantes si este software se desarrolla de forma colaborativa. La gestión de las versiones, permite aceptar, rechazar y hacer un seguimiento a los cambios mientras se desarrolla un mismo proyecto de forma simultánea, sin que los trabajadores se entorpezcan entre si. Además, trabajar sobre un repositorio facilita la tarea de asignar tareas, y ver desde una perspectiva global la evolución de un proyecto. También se facilita realizar un seguimiento a la metodología decidida para el desarrollo, como por ejemplo, *SCRUM*.

4.2 Git

Git es una herramienta de control de versiones distribuida. Es la herramienta de su tipo más utilizada con mucha diferencia. Al llevar mucho tiempo desarrollándose, es ya estable y eficaz. Es de código abierto y fue originalmente creado por Linus Torvalds. Es tan popular que se ha convertido en estándar de facto. [WWWatlassianDoc0]



Ilustración 22: Logo de Git [WWGitDoc1]

A diferencia de otros “*softwares*” de control de sistemas, *Git* no almacena simplemente los documentos por su nombre, sino que también almacena información sobre su contenido. Esto es interesante porque muchas veces los archivos se modifican, se cambian de nombre, se reordenan, pero el contenido no cambia completamente. Partiendo de estos datos muestra mucha información sobre modificaciones y la evolución de los propios documentos y archivos.

Git permite el desarrollo de un proyecto de forma no lineal, es decir, con varios flujos de trabajo.

Algunos de los conceptos básicos de esta herramienta son:

- Rama: una rama es una línea (espejo en el momento de su creación) independiente de desarrollo de un proyecto.
- Fork: es una copia personal de un repositorio.
- Pull: es una petición de integración de código sobre la rama en la que se esté trabajando.
- Push: es una petición de integración de los cambios de la rama local sobre la rama remota.
- *Commit*: hace lo mismo que el push pero en el repositorio local.

4.3 GitHub vs GitLab vs AWS CodeCommit

GitHub, *GitLab* y *AWS CodeCommit* tres de los repositorios más utilizados. Están basados en *Git* y son consideradas plataformas de desarrollo colaborativo en la nube.

GitHub es el repositorio líder en cuanto a almacenamiento de proyectos colaborativos de programación. También es el líder en cuanto a usuarios, ya que *GitHub* tiene más de 40 millones de usuarios con una gran cantidad de repositorios. Es un sistema gratuito, aunque tiene opciones de pago. Se desarrolla de forma colaborativa y descentralizada, lo que suele traducirse en proyectos más duraderos.

GitHub

Ilustración 23: Repositorio GitHub [WWWgithubDoc1]

GitLab es muy parecido a GitHub aunque sea un producto de una empresa completamente independiente. Pese a ser casi idéntico, tiene algunas opciones de pago que en GitHub son gratuitas como la asignación de múltiples responsables a una *Issue*, testing en el buscador o dependencias entre las *Issues*.



Ilustración 24: Repositorio GitLab [WWWgitlabDoc1]

AWS CodeCommit es muy parecido a los dos anteriores, pero está pensado para repositorios *Git* privados. No tiene versión gratuita.



Ilustración 25: Repositorio AWS CodeCommit [WWWSourceForgeDoc1]

Una de las diferencias principales entre *GitHub* y *GitLab* (las dos opciones son “open source”) es el flujo de trabajo. *GitHub* está más enfocado en ofrecer una mayor velocidad mientras que *GitLab* da más importancia a la fiabilidad.

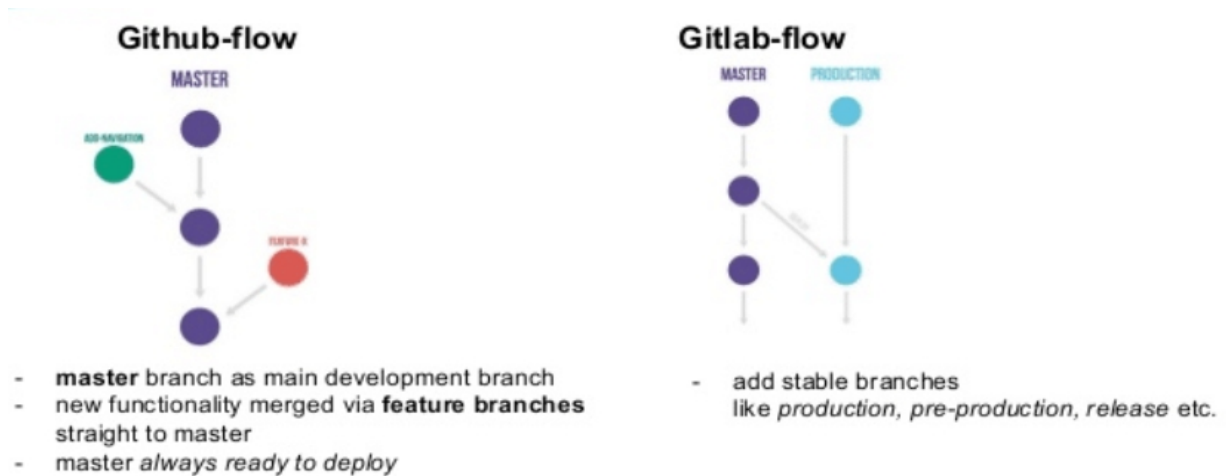


Ilustración 26: Gestión de las ramas en GitHub y GitLab[WWWkinstaDoc]

Se ha optado usar *GitHub* por ser gratuito y ofrecer el mayor número de posibilidades en su versión gratuita. Además, *GitHub* por sus características, está recomendado para equipos de trabajo pequeños y ágiles.

4.4 Metodología Scrum

La metodología *Scrum* es una de las metodologías más populares para el desarrollo de proyectos. Esta metodología es especialmente útil en la realización de proyectos largos en los que hay que realizar un trabajo constante y colaborativo. [WWWMetodologíaScrumDoc0]

En *Scrum* se pueden distinguir tres roles diferenciados; el “*scrum master*”, el “*product owner*” y el equipo de trabajo.

- *Scrum master*: es el responsable de hacer cumplir los acuerdos y las reglas establecidas, es el líder del equipo. También debe asegurarse de que el equipo está en las mejores condiciones posibles. Que salga adelante el “*Product Backlog*”, el “*Sprint Backlog*” y los “*Sprints*” son también parte de su responsabilidad.
- *Product owner*: es la representación del cliente, y por tanto, el dueño del producto. Su función es ir dictaminando los cambios que se deben hacer en el producto así como las expectativas. También es el encargado de priorizar las tareas que le interesen dando distintas importancias a las mismas.
- Equipo de trabajo: son los trabajadores que tienen un mayor conocimiento sobre las técnicas utilizadas ya que son quienes llevan a cabo el trabajo y cumplen con lo determinado por los dos roles anteriores. Es el propio equipo el que gestiona las tareas según sus necesidades para cumplir con los objetivos.

El trabajo realizado con esta metodología es un trabajo incremental. Cada iteración o reunión, generalmente cada dos o cuatro semanas, avanza en las partes determinadas como más urgentes. Estas tareas urgentes son añadidas al *sprint* que da comienzo en la iteración en la que se esté. Es esa

iterabilidad frecuente la que permite adaptarse a proyectos cambiantes.

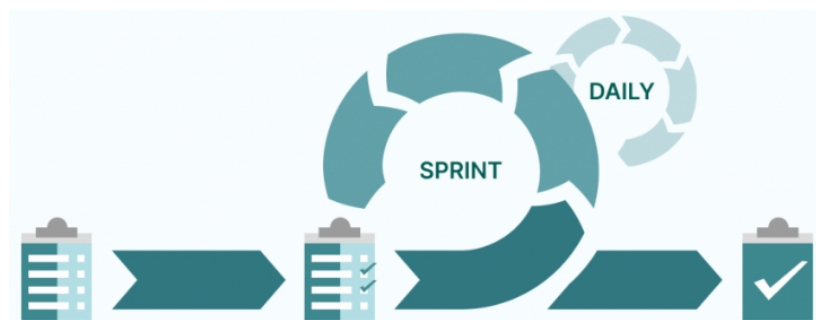


Ilustración 27: Metodología Scrum [WWWganttproDoc1]

Como se ve en la imagen anterior, también hay reuniones diarias. Estas reuniones tienen como finalidad, el actualizar y sincronizar las tareas del equipo. Tras esta reunión se actualiza el backlog. Es común entre las empresas que utilizan esta metodología que la “daily” se realice a primera hora de la mañana.

4.5 Lenguaje de programación: C

C es un lenguaje de programación. Es un lenguaje muy utilizado debido a su alta eficiencia y optimización. Se puede considerar uno de los lenguajes más importantes a día de hoy. Permite que el programador tenga una gran flexibilidad en cuanto a la gestión de los datos y por tanto de libertad. Por otro lado, tiene una baja comprobación de incorrecciones por lo que se delega parte de la responsabilidad del buen funcionamiento de los programas en el programador. Otra de sus mayores ventajas es su capacidad de migrar a distintos sistemas operativos debido a que no depende del *hardware* de los mismos. Está considerado como un lenguaje de nivel intermedio.

Un entorno de desarrollo (IDE) que se puede utilizar para desarrollar programas de C en *Linux* es *Geany*. *Geany* es un editor de texto sencillo compatible con una gran cantidad de lenguajes de programación. Realiza funciones básicas como el resaltado de sintaxis o completado de código. [WWWsoftonicDoc0]

4.6 MEX

MEX (“*MATLAB Executable*”) es un interfaz que provee la configuración a distintos compiladores para que los programas se puedan comunicar con *MATLAB*. Gracias a este interfaz, se pueden llamar a funciones de *MATLAB* desde un programa en C, C++ o Fortran. También funciona en el sentido contrario, es decir, *MEX* puede ser usado para llamar funciones programadas en C desde *MATLAB*. Este interfaz puede ser configurado con distintos compiladores. En el caso de los programas en C, el compilador que debe ser configurado en *MEX* es *gcc*. [WWWatmosDoc0]

```
>> mex -setup
com.jogamp.opengl.GLEException: X11GLXDrawableFactory - Could not
    at jogamp.opengl.x11.glx.X11GLXDrawableFactory$SharedRes
    at jogamp.opengl.SharedResourceRunner.run(SharedResource
    at java.lang.Thread.run(Thread.java:748)
Caused by: com.jogamp.opengl.GLEException: Startup Class Loader-S
    at jogamp.opengl.x11.glx.X11GLXContext.createImpl(X11GLX
    at jogamp.opengl.GLContextImpl.makeCurrentWithinLock(GLC
    at jogamp.opengl.GLContextImpl.makeCurrent(GLContextImpl
    at jogamp.opengl.GLContextImpl.makeCurrent(GLContextImpl
    at jogamp.opengl.x11.glx.X11GLXDrawableFactory$SharedRes
    ... 2 more
MEX configured to use 'gcc' for C language compilation.
To choose a different language, select one from the following:
mex -setup C++
mex -setup FORTRAN
```

Ilustración 28: Mex setup

Mex es también un tipo de archivo usado en *MATLAB* pero en este caso no debe ser confundido con este interfaz.

4.7 Documentación

En cuanto a la documentación se han utilizado varias herramientas del paquete *Apache OpenOffice*. Para la redacción de los documentos se ha utilizado *OpenOffice Writer*, para la creación de algunas imágenes, *OpenOffice Draw*, para la gestión de algunos datos *OpenOffice Calc* y para la creación de las fórmulas *OpenOffice Math*. Todas estas herramientas son sencillas de usar y de código abierto y es por ello que se ha decidido usarlas.

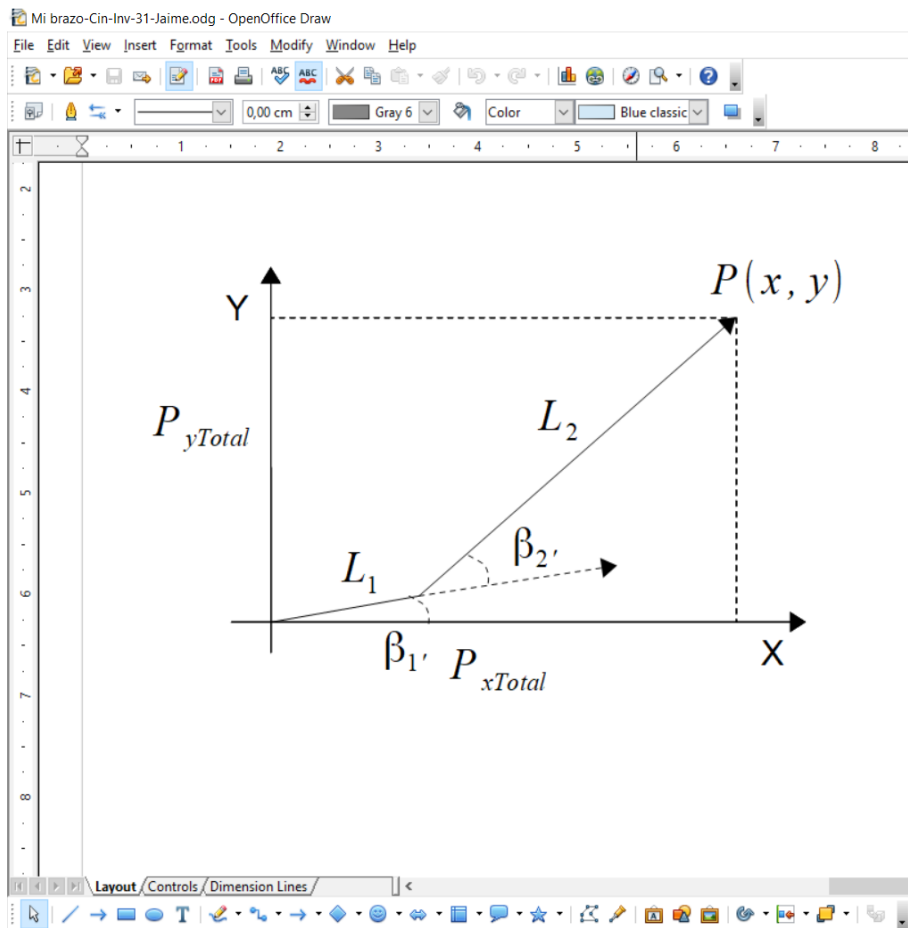


Ilustración 29: Ejemplo OpenOffice Draw

4.8 Arduino

Arduino es una plataforma de código cuya finalidad es la construcción de proyectos electrónicos. Arduino está compuesto por micro-controladores y un entorno de desarrollo integrado. El IDE se ejecuta en el ordenador del usuario y es el que nos permite desarrollar los programas y subirlos a las placas físicas. El código que se usa se programa en una versión simplificada de C++. [WWWxatakaDoc0]

Arduino ofrece una gran cantidad de placas distintas, cada una con cualidades que las hacen útiles para distintas funciones.

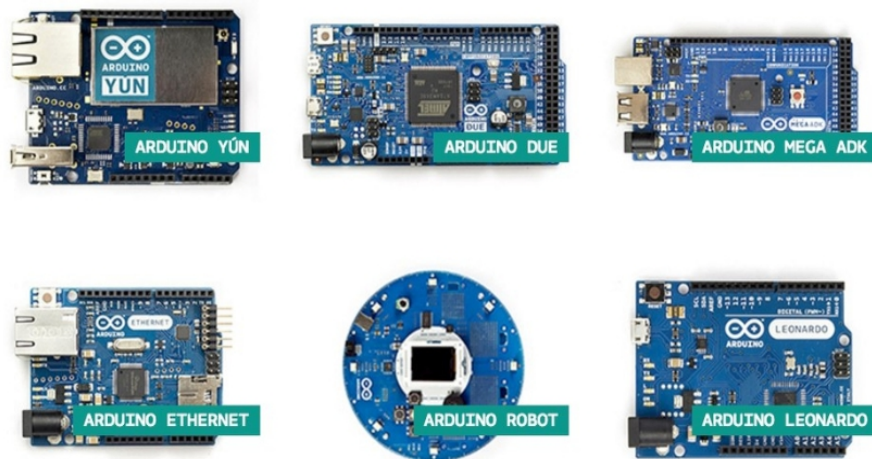


Ilustración 30: Ejemplos placas Arduino [WWWxatakaDoc0]

Los Arduino tienen tanto interfaces de entrada como interfaces de salida por lo que pueden tanto procesar información entrante como realizar acciones a través de los puertos periféricos (encender luces, producir sonido, mover motores, etc).

Se pueden realizar un sinnúmero de proyectos distintos con ellos, desde programar un despertador hasta realizar una báscula o un sistema de acceso por medio de huella dactilar. [WWWyoutube0]

4.9 Librerías

4.9.1 Robotics System Toolbox

Esta librería tiene como objetivo, ayudar en el diseño y simulación de aplicaciones relacionadas con la robótica. Es una librería de *MATLAB*. Esta librería incluye algunas herramientas y algoritmos que ayudan en el desarrollo de robots de distintos tipos, como por ejemplo brazos robóticos, coches autónomos etc. También son interesantes los algoritmos implementados con distintas funcionalidades, como por ejemplo, para optimizar desplazamientos. [WWWmathworksDoc0]

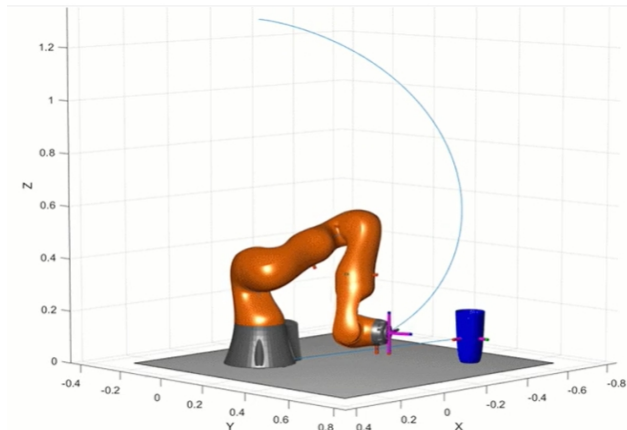


Ilustración 31: Ejemplo librería “Robotics System Toolbox” [WWWmathworksDoc0]

4.9.2 Symbolic Math Toolbox

Esta librería permite trabajar en *MATLAB* con variables simbólicas, sin la necesidad de tener que trabajar con variables que tengan asignado un valor. En el siguiente ejemplo se puede ver como la biblioteca permite que se usen variables simbólicas incluso estando mezcladas con variables numéricas. [WWWmathworksDoc1]

```
I1=[cos(b1) sin(b1) 0 0;  
    0 0 -1 (d1);  
    -sin(b1) cos(b1) 0 0;  
    0 0 0 1]
```

Ilustración 32: Ejemplo uso de variables simbólicas.

Las operaciones entre estas variables simbólicas y las numéricas o entre variables simbólicas están permitidas.

Por ejemplo, en la siguiente operación, se multiplican y suman números con variables simbólicas y no da ningún error, además de simplificar todo lo posible, eliminando las multiplicaciones que dan 0.

```
syms a1 a2 b1 b2 d1 d2 r1 r2

N1=[cos(b1) -cos(a1)*sin(b1) sin(a1)*sin(b1) r1*cos(b1);
    sin(b1) cos(a1)*cos(b1) -sin(a1)*cos(b1) r1*sin(b1);
    0 sin(a1) cos(a1) d1;
    0 0 0 1]
N2=[cos(b2) -cos(a2)*sin(b2) sin(a2)*sin(b2) r2*cos(b2);
    sin(b2) cos(a2)*cos(b2) -sin(a2)*cos(b2) r2*sin(b2);
    0 sin(a2) cos(a2) d2;
    0 0 0 1]

T=N1*N2
```

Ilustración 33: Ejemplo operación variables simbólicas

4.9.3 Peter Corke Robotics Toolbox

Esta librería es similar a la expuesta inicialmente, pero permite representar brazos robóticos de una forma mucho más simple. No obstante, es una librería que lleva desarrollándose 20 años de forma independiente a *MATLAB*. Algunas de las ventajas que tiene sobre la anterior es su código depurado, una programación clara y sencilla que permite el entendimiento y el acceso al código para analizarlo, estudiarlo e incluso modificarlo. [WWWRobotics-ToolboxDoc0]

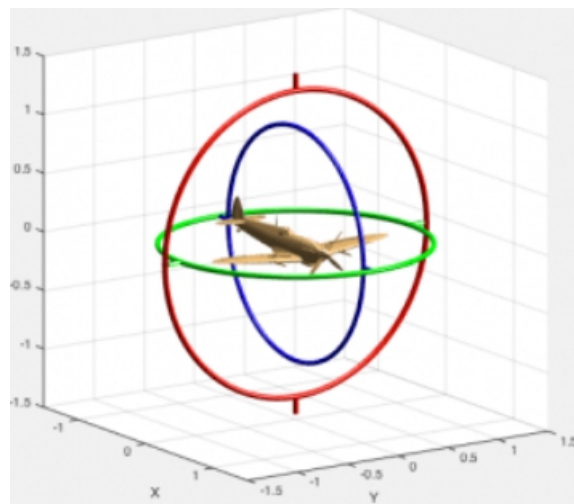


Ilustración 34: Ejemplo uso de variables simbólicas.

Una diferencia respecto a *Symbolic Math Toolbox* es el soporte. Esta librería y sus funcionalidades cuentan con un grupo en el que se comentan cosas relacionadas con la misma, pero no hay ninguna vinculación que obligue a que los fallos sean solucionados. La librería *Symbolic Math Toolbox*, al depender de una compañía privada si tiene esa responsabilidad con sus clientes. Por tanto, si bien es cierto que para un uso académico es indiferente cuál usar, para un uso comercial o empresarial, es mejor usar la librería *Symbolic Math Toolbox*. [WWWRobotics-ToolboxDoc1]

5 Aspectos relevantes del desarrollo del proyecto

El desarrollo de este proyecto ha evolucionado en todo momento, en paralelo al conocimiento del problema que se quería resolver. Se partía de la comprensión del problema desde una perspectiva trigonométrica, muy diferente a la solución analítica buscada.

El no entender desde un inicio completamente qué es lo que se buscaba a nivel matemático, ha hecho que se haya tenido que invertir un valioso tiempo en investigación. Además, el análisis encontrado sobre cinemática inversa en la red era “incompleto” o “erróneo” en muchos casos. Esto, unido a la falta de ejemplos ha acabado haciendo que uno de los puntos fuertes de este TFG sea la investigación que lo acompaña.

Así, se van a presentar tres aspectos importantes que marcan claramente el desarrollo del proyecto:

- Un primer apartado con un **planteamiento inicial** de las estrategias definidas para la resolución del problema a la hora de generar las ecuaciones que caracterizarán cualquier brazo robótico.
- En un segundo apartado, se presenta un momento de la evolución en el que ya se habían encontrado varios **mecanismos con los que reducir el número de variables articulares**, y con ello proponer una posible solución a cualquier modelo planteado.
- Por último una **Metodología** que ha resultado ser muy interesante, tanto porque puede ayudar en la resolución del problema, como porque será la base para la codificación de una aplicación que automatice la resolución del problema. Como cuarto apartado, un resumen de la gran cantidad de problemas resueltos con la Metodología explicada.

Además, habría que señalar que los diferentes robots que se pueden considerar como “muy conocidos” y “muy utilizados”, hasta ser conocidos por los nombres propios con quienes se relacionan, como el robot Stanford, SCARA, entre otros, también han sido resueltos con nuestra Metodología.

5.1 Estrategias de aproximación al extremo del brazo para el cálculo en Cinemática Inversa – Planteamiento inicial

Para la realización del TFG se ha planteado un problema genérico con motores montados sobre articulaciones sin un plan previo, ni suposición alguna de movimientos, salvo cierto grado de alcanzabilidad y complejidad.

Así la arquitectura del brazo podía ser el mostrado a continuación.

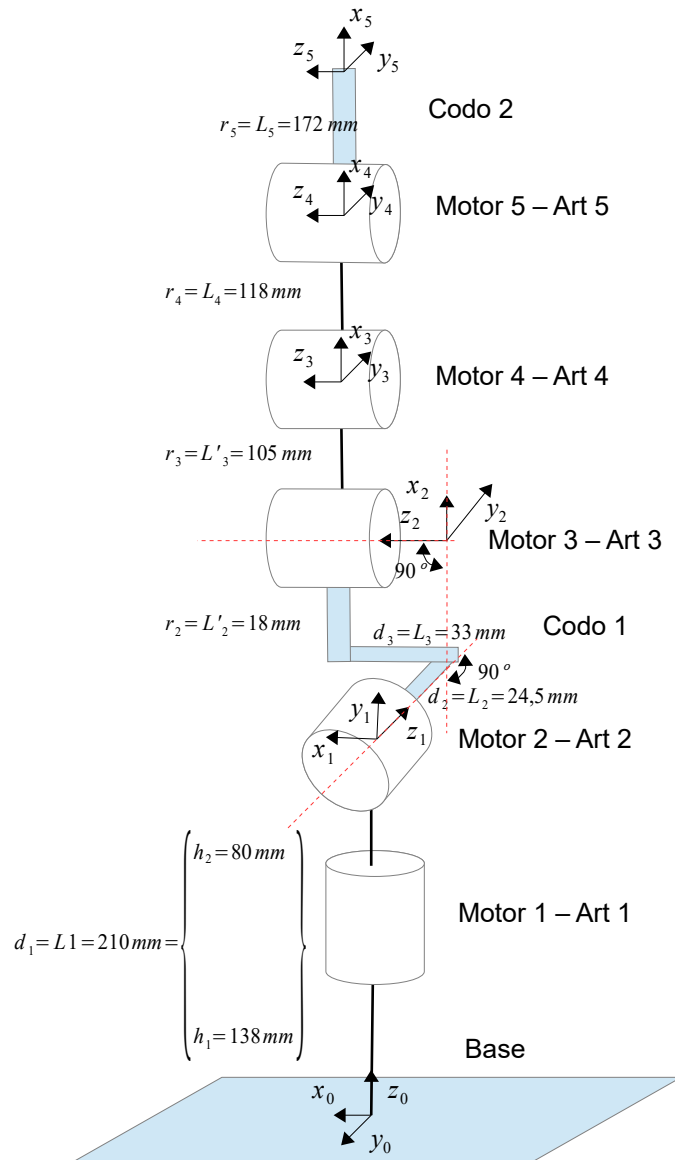


Ilustración 35: Brazo de prueba para presentación de TFG

De cara a solucionar el problema, y por la dificultad de resolver el problema de forma geométrica, se desarrollaría mediante el modelo de Denavit-Hartenberg.

Sin embargo, el montaje de los motores define elementos que previsiblemente trabajarían de forma interdependiente, y cuyos giros podrían ser corregidos por el giro de otros motores, lo que lleva a tener infinitas posibles soluciones (motores 3, 4 y 5).

Además, la existencia de desplazamientos laterales que modifican la dirección de avance del extremo del brazo en movimiento hacia el punto a alcanzar supone un problema añadido (conjunto de motores 2 y 3). Este problema se podría resolver haciendo que los diferentes motores orienten el

brazo hacia el objetivo, determinando los giros necesarios para alcanzar el plano de ataque hacia ese objetivo. Y el plano de ataque sería aquel en el que los movimientos de los siguientes motores mantengan la dirección adecuada de desplazamiento, por tanto, sin desplazamientos laterales.

A la hora de enfocar las soluciones, a veces múltiples, e incluso infinitas, de los posibles ángulos de los motores, se podría determinar la tipología de las articulaciones, más allá de que puedan ser esféricas o prismáticas. En este caso podrían definirse:

- Articulaciones de Orientación: aquellas articulaciones que direccionan el brazo hacia el punto a alcanzar y sin cambio de plano por no tener desplazamientos perpendiculares al plano de ataque. Por ejemplo, las que habitualmente sirven de base, girando el brazo para encarar el brazo hacia el punto a alcanzar (Motor 1).
- Articulaciones de Ajuste al Plano de Ataque: aquellas articulaciones que direccionan el brazo hacia el punto a alcanzar con cambio de plano por tener desplazamientos perpendiculares fuera del plano de ataque. Por ejemplo, aquellas que introducen al brazo en el plano de ataque, mismo plano donde se encuentra el punto a alcanzar (Motor 2) para que puedan ser alcanzados por otras articulaciones.
- Articulaciones de Aproximación: aquellas que acercan al extremo del brazo hacia el punto a alcanzar. Por ejemplo, las de tipo prismático, cuando acercan una herramienta hacia el objetivo.
- Articulaciones de Corrección y Aproximación: aquellas que, podrían no trabajar en solitario, y acercan al extremo del brazo hacia el punto a alcanzar una vez se encuentra en el mismo plano de ataque. Por ejemplo, motores montados con ejes de giro en paralelo (Motores 3, 4 y 5) como podrían ser las que forman parte de un modelo SCARA.
- Articulaciones de Montaje de Herramientas: aquellas que sirven para hacer una corrección final de orientación y sobre las que se podrían montar diferentes tipos de herramientas (tenazas, taladros, soldadores,...), cuyo extremo final debe orientarse convenientemente al elemento sobre el que trabaja.

En este trabajo no se tienen en cuenta las Articulaciones de Montaje de Herramientas dado que representan otro tipo de problema y dependen del tipo de herramienta y de sus propias necesidades de orientación.

Tampoco se tratan las Articulaciones Prismáticas ya que el problema fundamenta a resolver es la orientación y alcanzabilidad del punto.

Por tanto, solo quedan los tres primeros tipos.

De esta forma el "Motor 1" sería una Articulaciones de Orientación, y a priori podría calcularse por aproximación, y en la misma dirección que el propio punto a alcanzar.

El "Motor 2" buscaría la orientación del plano de ataque, importante cuando el ángulo de entrada al objetivo es significativo. Por tanto, también sería una Articulaciones de Orientación. Y por ello, a priori, podría ser calculado por aproximación y solo con el dato de la necesidad de un determinado plano de ataque, si fuera necesario.

El "Motor 3", sin embargo, está desplazado lateralmente al tener una componente perpendicular a la dirección hacia el objetivo, y que debe ser corregido por anteriores o posteriores motores por lo que, sería una Articulaciones de Ajuste al Plano de Ataque. Y en este caso, los siguientes motores no tienen esa capacidad de corrección, ya que solo trabajan sobre un plano de ataque, una vez alcanzado este. Por tanto, la corrección debe hacerse con los motores anteriores, en este caso, con el "Motor 1".

Como consecuencia, la dirección en el eje Y debe ser tal que a lo largo del brazo se avance precisamente el valor de esa componente y solo con los motores que afecten a esa dirección (Caso-a)"Motor 1" para alcanzar el plano de ataque más los motores 3, 4 y 5 para alcanzar el punto, o

Caso-b) "Motor 1" y "Motor 2" para alcanzar el plano y el resto para alcanzar el punto en el punto). Mientras que, las componentes X y Z serán alcanzadas con el resto de motores. Y como podría darse el caso de que el punto no pudiera ser alcanzado si sólo se moviera el "Motor 1", parece razonable pensar que el uso de los motores "Motor 1" y "Motor 2" permitirían alcanzar el plano en cualquier caso. Además, parece previsible pensar que aumentaría la alcanzabilidad del brazo. Por ello parece razonable pensar en que pueda ser predeterminado el giro del primer motor y calcular el giro del segundo para poder ser colocado el resto de motores sobre el plano de ataque, y en cualquier caso, si fuera necesario corregir ese primer giro, si no se alcanzara ese plano de ataque. Por otra parte, los últimos tres motores podrían tomar infinitas soluciones puesto que hay más motores que problemas a resolver una vez alcanzado el plano de ataque (tres motores y dos componentes en el plano), lo que permitiría predeterminar uno de los ángulos para calcular los otros dos, y comprobar la idoneidad del ángulo predeterminado para su posterior corrección.

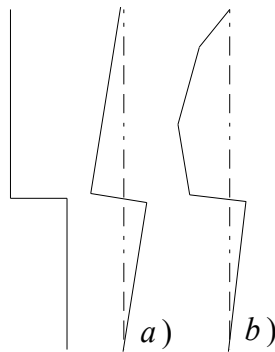


Ilustración 36: Estrategias de alcanzabilidad del Brazo de prueba para presentación de TFG

Cualquiera que sea la solución implementada, la herramienta con la que se verificará el correcto cálculo de los ángulos será el propio modelo de Denavit-Hartenberg, aplicando Cinemática Directa, ya que este modelo permite generar, de forma rápida y precisa, las ecuaciones necesarias para calcular el punto final, una vez aplicados los correspondientes ángulos de giro de cada motor del brazo.

5.2 Buscando soluciones lógicas a la caracterización de los brazos robóticos (Desacoplamiento y Plano y Ángulo de Ataque)

Como se comentaba en el apartado de conceptos teóricos, parecía razonable pensar que de Desacoplamiento de Grupos de Articulaciones seguía siendo una solución aceptable aún encontrando muchos problemas a la hora de definir las ecuaciones que caracterizaran los brazos robóticos.

Además, el problema general de localización de ángulos para alcanzar una determinada posición con el extremo del brazo robótico, experimentalmente se ha dividido en otros tres aspectos de discusión derivados, a resolver:

- La existencia de una solución (definición del espacio de trabajo – Localización/Orientación)
- Unicidad/Multiplicidad/Infinitas soluciones
- Métodos para la búsqueda de soluciones

Parecía claro que había soluciones infinitas para las diferentes variables articulares de los problemas planteados, aún cuando tenían un número bajo de articulaciones, pero tras el conocimiento adquirido a lo largo del TFG se ha llegado a comprender cómo pueden ser reducidas con diferentes estrategias.

Por un lado con la elección de un Plano de Ataque con los que buscar el plano con el que acercarse al punto de destino para poder entrar en el espacio de trabajo por reducido e inalcanzable que parezca, y por otro con la elección de un Ángulo de Ataque con el que acercar el extremo del brazo al punto de destino para utilizar la herramienta con la orientación adecuada. Luego, ya se cuenta con alguna condición que permitirá la elección de cierto número de ángulos.

Aún así, podría haber más articulaciones que resolver. Y otra posibilidad es la búsqueda de ciertos motores montados de forma consecutiva con sus ejes de giro paralelos. Esto representa un problema dado que, tres motores con esa condición darían como resultado infinitas soluciones por estar en un mismo plano y por tanto poderse definir solo con dos variables. La forma de resolverlo es proponiendo un ángulo por algún criterio, por ejemplo, un ángulo de ataque determinado o cierta proporcionalidad con el tamaño del segmento al que está asociado, para una de las tres articulaciones, de tal forma que solo resten dos variables articulares para dos motores, problema que puede ser resuelto.

Luego, parece razonable pensar que, cualquiera que sea el número de articulaciones, se pueda intentar localizar los motores consecutivos con ejes paralelos, grupos de tres motores de ejes no paralelos, otras condiciones como planos de ataque o ángulos de ataque, o cualquier otra condición que pueda ser aplicada al problema. Y una vez hecho este análisis, se podría desacoplar los grupos de articulaciones para poder ser resueltos por separado.

De hecho, si se hiciera un primer análisis de una primera agrupación de articulaciones (Véase el “Ejemplo 1-0” de la Documentación Técnica) se vería que podría calcularse el punto en el que se encontraría el extremo del segmento asociado, aplicando las ecuaciones que provienen de las Matrices de Transformación Homogénea utilizadas en el Algoritmo de Denavit-Hartenberg. Pues la consecuencia inmediata es que ese punto será la base del siguiente grupo de articulaciones.

Si ahora se resolviera el segundo grupo, se alcanzaría el extremo de ese segundo grupo, y por tanto al acoplar los resultados, se encontraría la posición real de ese punto respecto al Sistema de Coordenadas de la Base, base del primer grupo de articulaciones.

Si se repitiera este algoritmo de forma repetida, se alcanzaría a resolver el problema completo para un brazo robótico complejo.

Evidentemente, a priori no se sabe qué problemas deberían resolverse, como puede ser que se añadan dificultades como que las articulaciones puedan tener ángulos diferentes de 0° o 90° . Pero tras la realización del TFG, se ha demostrado que el método parece funcionar correctamente.

5.3 Metodología para el cálculo de datos desde la aplicación y su uso en la caracterización de brazos robóticos – Resolución final

En este apartado se dará una relación de pasos a seguir a la hora de resolver el análisis de un brazo robótico utilizando la experiencia y conocimientos adquiridos a lo largo del TFG. Con ellos se obtendrán las ecuaciones que caractericen cualquier brazo robótico.

Dado un caso de estudio de 5 articulaciones como:

$$\begin{aligned}
 T &= {}^0A_1 {}^1A_2 {}^2A_3 {}^3A_4 {}^4A_5 = {}^0A_5 \\
 ({}^0A_1)^{-1} T &= {}^1A_2 {}^2A_3 {}^3A_4 {}^4A_5 = {}^1A_5 \\
 ({}^1A_2)^{-1} ({}^0A_1)^{-1} T &= {}^2A_3 {}^3A_4 {}^4A_5 = {}^2A_5 \\
 ({}^2A_3)^{-1} ({}^1A_2)^{-1} ({}^0A_1)^{-1} T &= {}^3A_4 {}^4A_5 {}^5A_6 = {}^3A_5 \\
 ({}^3A_4)^{-1} ({}^2A_3)^{-1} ({}^1A_2)^{-1} ({}^0A_1)^{-1} T &= {}^4A_5 {}^5A_6 = {}^4A_5
 \end{aligned}$$

Y dadas las matrices correspondientes a las anteriores igualdades:

$$\begin{aligned}
 ({}^0A_1)^{-1} T &= \begin{pmatrix} f_{11}(n) & f_{11}(o) & f_{11}(a) & f_{11}(p) \\ f_{12}(n) & f_{12}(o) & f_{12}(a) & f_{12}(p) \\ f_{13}(n) & f_{13}(o) & f_{13}(a) & f_{13}(p) \\ 0 & 0 & 0 & 1 \end{pmatrix} = {}^1A_5 \\
 ({}^1A_2)^{-1} ({}^0A_1)^{-1} T &= \begin{pmatrix} f_{21}(n) & f_{21}(o) & f_{21}(a) & f_{21}(p) \\ f_{22}(n) & f_{22}(o) & f_{22}(a) & f_{22}(p) \\ f_{23}(n) & f_{23}(o) & f_{23}(a) & f_{23}(p) \\ 0 & 0 & 0 & 1 \end{pmatrix} = {}^2A_5 \\
 ({}^2A_3)^{-1} ({}^1A_2)^{-1} ({}^0A_1)^{-1} T &= \begin{pmatrix} f_{31}(n) & f_{31}(o) & f_{31}(a) & f_{31}(p) \\ f_{32}(n) & f_{32}(o) & f_{32}(a) & f_{32}(p) \\ f_{33}(n) & f_{33}(o) & f_{33}(a) & f_{33}(p) \\ 0 & 0 & 0 & 1 \end{pmatrix} = {}^3A_5 \\
 ({}^3A_4)^{-1} ({}^2A_3)^{-1} ({}^1A_2)^{-1} ({}^0A_1)^{-1} T &= \begin{pmatrix} f_{41}(n) & f_{41}(o) & f_{41}(a) & f_{41}(p) \\ f_{42}(n) & f_{42}(o) & f_{42}(a) & f_{42}(p) \\ f_{43}(n) & f_{43}(o) & f_{43}(a) & f_{43}(p) \\ 0 & 0 & 0 & 1 \end{pmatrix} = {}^4A_5
 \end{aligned}$$

...

Para el análisis de los datos se deben extraer siempre, como subproductos de la aplicación, los siguientes datos desde MATLAB (datos simbólicos), tanto simplificados (MATLAB: simplify) como no simplificados:

- Matrices con todos sus componentes completos y simplificados.
- Matrices tanto normales como inversas:
 - Matrices normales de cada articulación por separado (0A_1 , 1A_2 , 2A_3 , ...)
 - Matrices inversas de cada articulación por separado ($({}^0A_1)^{-1}$, $({}^1A_2)^{-1}$, ...)
 - Matriz total como producto de todas las matrices normales (

$$T = {}^0A_1 {}^1A_2 {}^2A_3 {}^3A_4 {}^4A_5 = {}^0A_5$$

- Matrices normales (derecha de la igualdad) multiplicadas, en cada igualdad (${}^1A_2 {}^2A_3 {}^3A_4 {}^4A_5 = {}^1A_5$, ${}^2A_3 {}^3A_4 {}^4A_5 = {}^2A_5$, ...)
- Matrices inversas (izquierda de la igualdad) multiplicadas, en cada igualdad ($({}^1A_2)^{-1}({}^0A_1)^{-1}$, $({}^2A_3)^{-1}({}^1A_2)^{-1}({}^0A_1)^{-1}$, ...)
- Matrices inversas y la total de la izquierda de la igualdad multiplicadas, en cada igualdad ($({}^0A_1)^{-1}T$, $({}^1A_2)^{-1}({}^0A_1)^{-1}T$, ...)
- Matrices normales de cada par de articulaciones por separado (Art1-Art2, Art2-Art3, Art3-Art4, ...)
- Matrices normales de cada tres articulaciones por separado (Art1-Art2-Art3, Art2-Art3-Art4, Art3-Art4-Art5, ...)
- Matrices normales de cada tres articulaciones por separado y sin un orden consecutivo (Art1-Art3-Art4, Art1-Art4-Art5, ..., Art1-Art2-Art4, Art1-Art2-Art5, ..., Art2-Art4-Art5, Art2-Art5-Art6, ...)
- Matrices normales de cada cuatro articulaciones por separado (Art1-Art2-Art3-Art4, Art2-Art3-Art4-Art5, Art3-Art4-Art5-Art6, ...)
- Matrices normales de cada cuatro articulaciones por separado y sin un orden consecutivo (Art1-Art3-Art4-Art5, Art1-Art4-Art5-Art6, ..., Art2-Art4-Art5-Art6, ..., Art1-Art2-Art4-Art5, ..., Art1-Art2-Art3-Art5, ...)
- ...Por tanto, de forma general... Matrices normales de cada subconjunto de n articulaciones por separado, con y sin orden consecutivo, hasta que se llegue al máximo posible en cada problema.
- Resolver todos los casos anteriores de productos de matrices, agrupando las articulaciones desde el principio, y desde el final del robot (Art1; Art1-Art2; Art1-Art2-Art3;... Art5; Art4-Art5; Art3-Art4-Art5, ...)
- Para cada caso de estudio, extraer y calcular:
 - Los elementos de cada componente del punto desde la última columna de T : P_x , P_y , P_z
 - Calcular el cuadrado de los elementos de cada componente del punto desde la última columna de T : P_x^2 , $(P_y)^2$, $(P_z)^2$
 - Calcular la suma de los cuadrados de los elementos de cada componente del punto desde la última columna de T :
 - $(P_x)^2 + (P_y)^2$
 - $(P_x)^2 + (P_z)^2$
 - $(P_y)^2 + (P_z)^2$
 - $(P_x)^2 + (P_y)^2 + (P_z)^2$
 - Multiplicar, en el lado izquierdo de cada igualdad, la primera fila del producto de las inversas por la última columna de la matriz T ($({}^0A_1)^{-1}T$, $({}^1A_2)^{-1}({}^0A_1)^{-1}T$, ...), determinando su formato ($f_{i1} = C\theta_1 x + S\theta_1 y + L_1 t$, f_{i2} , f_{i3}) y generando las igualdades correspondientes, al igualarlas al elemento correspondiente del producto de las matrices normales de la derecha (${}^1A_2 {}^2A_3 {}^3A_4 {}^4A_5 = {}^1A_5$, ${}^2A_3 {}^3A_4 {}^4A_5 = {}^2A_5$, ...):

Formatos:

$$f_{11} = C\theta_1 x + S\theta_1 y$$

$$f_{12} = z - L_1 t$$

$$f_{13} = S\theta_1 x - C\theta_1 y$$

$$\begin{aligned}
f_{21} &= -C\theta_1 S\theta_2 x - S\theta_2 S\theta_1 y + C\theta_2 z - (L'_2 + L_1 C\theta_2)t \\
f_{22} &= S\theta_1 x - C\theta_1 y - L_2 t \\
f_{23} &= C\theta_1 C\theta_2 x + C\theta_2 S\theta_1 y + S\theta_2 z - L_1 S\theta_2 t \\
\\
f_{31} &= (S\theta_1 S\theta_3 - C\theta_1 S\theta_2 C\theta_3)x + (-S\theta_2 C\theta_3 S\theta_1 - C\theta_1 S\theta_3)y + C\theta_3 C\theta_2 z - \dots \\
&\quad \dots - (L'_3 + L'_2 C\theta_3 + L_2 S\theta_3 + L_1 C\theta_3 C\theta_2)t \\
f_{32} &= (C\theta_3 S\theta_1 + C\theta_1 S\theta_2 S\theta_3)x - (C\theta_1 C\theta_3 - S\theta_2 S\theta_1 S\theta_3)y - C\theta_2 S\theta_3 z + \dots \\
&\quad \dots + (L'_2 S\theta_3 - L_2 C\theta_3 + L_1 C\theta_2 S\theta_3)t \\
f_{33} &= C\theta_1 C\theta_2 x + S\theta_1 C\theta_2 y + S\theta_2 z - (L_1 S\theta_2 - L_3)t
\end{aligned}$$

...

Igualdades:

$$\begin{aligned}
f_{11}(n) &= C\theta_1 x + S\theta_1 y = C\theta_1 f'_{11}(n) + S\theta_1 f'_{12}(n) = -S\theta_2 C(\theta_3 + \theta_4 + \theta_5) \\
f_{12}(n) &= z - L_1 t = f'_{13}(n) = C\theta_2 C(\theta_3 + \theta_4 + \theta_5) \\
f_{13}(n) &= S\theta_1 x - C\theta_1 y = S\theta_1 f'_{11}(n) - C\theta_1 f'_{12}(n) = S(\theta_3 + \theta_4 + \theta_5)
\end{aligned}$$

$$\begin{aligned}
f_{11}(o) &= C\theta_1 x + S\theta_1 y = C\theta_1 f'_{11}(o) + S\theta_1 f'_{12}(o) = S\theta_2 S(\theta_3 + \theta_4 + \theta_5) \\
f_{12}(o) &= z - L_1 t = f'_{13}(o) = -C\theta_2 S(\theta_3 + \theta_4 + \theta_5) \\
f_{13}(o) &= S\theta_1 x - C\theta_1 y = S\theta_1 f'_{11}(o) - C\theta_1 f'_{12}(o) = C(\theta_3 + \theta_4 + \theta_5)
\end{aligned}$$

$$\begin{aligned}
f_{11}(a) &= C\theta_1 x + S\theta_1 y = C\theta_1 f'_{11}(a) + S\theta_1 f'_{12}(a) = C\theta_2 \\
f_{12}(a) &= z - L_1 t = f'_{13}(a) = S\theta_2 \\
f_{13}(a) &= S\theta_1 x - C\theta_1 y = S\theta_1 f'_{11}(a) - C\theta_1 f'_{12}(a) = 0
\end{aligned}$$

$$\begin{aligned}
f_{11}(p) &= C\theta_1 x + S\theta_1 y = C\theta_1 f'_{11}(p) + S\theta_1 f'_{12}(p) = \\
&\quad -L'_2 S\theta_2 + L_3 C\theta_2 - L'_3 S\theta_2 C\theta_3 - \dots \\
&\quad \dots + L_5 S(\theta_3 + \theta_4) S\theta_2 S\theta_5 - L_4 S\theta_2 C\theta_3 C\theta_4 + L_4 S\theta_2 S\theta_3 S\theta_4 + \dots \\
&\quad \dots - L_5 C(\theta_3 + \theta_4) S\theta_2 C\theta_5 \\
f_{12}(p) &= z - L_1 t = f'_{13}(p) = \\
&\quad L'_2 C\theta_2 + L_3 S\theta_2 + L'_3 C\theta_3 C\theta_2 - L_5 S(\theta_3 + \theta_4) C\theta_2 S\theta_5 + \dots \\
&\quad \dots + L_4 C\theta_3 C\theta_4 C\theta_2 - L_4 C\theta_2 S\theta_3 S\theta_4 + L_5 C(\theta_3 + \theta_4) C\theta_5 C\theta_2 \\
f_{13}(p) &= S\theta_1 x - C\theta_1 y = S\theta_1 f'_{11}(p) - C\theta_1 f'_{12}(p) = L_2 + L_4 S(\theta_3 + \theta_4) + \dots \\
&\quad \dots + L_3 S\theta_3 + L_5 S(\theta_3 + \theta_4 + \theta_5)
\end{aligned}$$

Donde, en este caso se sustituirían los términos f'_{ij} por sus datos, por ejemplo:

$$f'_{13}(p) = L_2 + L_4 S(\theta_3 + \theta_4) + L_3 S\theta_3 + L_5 S(\theta_3 + \theta_4 + \theta_5)$$

...

- El mismo caso anterior pero dejando sin sustituir las componentes del punto de la cuarta columna de la derecha del lado izquierdo:

$$\begin{aligned}
f_{11}(p) &= C\theta_1 P_x + S\theta_1 P_y = \\
&\quad -L'_2 S\theta_2 + L_3 C\theta_2 - L'_3 S\theta_2 C\theta_3 + L_5 S(\theta_3 + \theta_4) S\theta_2 S\theta_5 + \dots \\
&\quad \dots - L_4 S\theta_2 C\theta_3 C\theta_4 + L_4 S\theta_2 S\theta_3 S\theta_4 - L_5 C(\theta_3 + \theta_4) S\theta_2 C\theta_5 \\
f_{12}(p) &= P_z - L_1 = L'_2 C\theta_2 + L_3 S\theta_2 - L'_3 C\theta_3 C\theta_2 - L_5 S(\theta_3 + \theta_4) C\theta_2 S\theta_5 + \dots
\end{aligned}$$

$$\dots + L_4 C\theta_3 C\theta_4 C\theta_2 - L_4 C\theta_2 S\theta_3 S\theta_4 + L_5 C(\theta_3 + \theta_4) C\theta_5 C\theta_2$$

$$f_{13}(p) = S\theta_1 P_x - C\theta_1 P_y = L_2 + L_4 S(\theta_3 + \theta_4) + L_3 S\theta_3 + L_5 S(\theta_3 + \theta_4 + \theta_5)$$

Donde los términos P_x , P_y , P_z no se sustituyen por ser las componentes del punto destino, y por tanto, valores conocidos.

...

- Probar a extraer una solución para todas sus variables articulares de la Cinemática Inversa:
 - Intentar localizar las igualdades de las componentes P_x , P_y , P_z a una constante o “0” cuando sólo tienen una variable articular.
 - Intentar localizar si alguno de los productos $(P_x)^2 + (P_y)^2$, $(P_x)^2 + (P_z)^2$, $(P_y)^2 + (P_z)^2$ y $(P_x)^2 + (P_y)^2 + (P_z)^2$ tienen una única variable tras ser simplificados.
 - Intentar localizar si alguna de las igualdades $f_{11} = C\theta_1 x + S\theta_1 y$, $f_{12} = z - L_1 t$, $f_{13} = S\theta_1 x - C\theta_1 y$, $f_{21} = -C\theta_1 S\theta_2 x - S\theta_2 S\theta_1 y + C\theta_2 z - (L'_2 + L_1 C\theta_2)t$, $f_{22} = S\theta_1 x - C\theta_1 y - L_2 t$, ... tienen una única variable y son igualadas a una constante o “0”.
- Intentar en cada caso resolver (MATLAB: solve) cada variable articular en cada caso y extraer las ecuaciones resultantes, tanto en las ecuaciones completas, como en las simplificadas, determinando si se encuentra una solución.

Para el ejecución de la aplicación se debe poder determinar qué tipo de articulaciones existen y tratar de resolver cada problema mediante los siguientes pasos:

- Petición de datos para resolver los parámetros de DH.
- Análisis de los parámetros de DH para determinar qué articulaciones tienen ejes en paralelo dado que, si se encuentran, determinarán la forma de calcular las soluciones.
- Si se encuentran posibles agrupaciones de articulaciones con ejes paralelos o codos, se debe pedir qué plano o segmento de los introducidos son dominantes para determinar el plano de ataque de las articulaciones agrupadas (vertical u oblicuo, y con qué ángulo), y el tipo de ángulo de ataque del último segmento (más vertical o más plano, o con qué ángulo).
- Hacer el análisis de los datos de las matrices resultantes, tal y como se ha definido anteriormente.
- En caso de no poder determinar una solución calculable mediante las ecuaciones correspondientes a las diferentes variables articulares, se debe intentar realizar el tratamiento separando los grupos de articulaciones y calculando cada grupo por separado, para después, unir los resultados obtenidos, de forma lógica.

Determinación de grupos de articulaciones si no pueden ser calculados con otras metodologías:

- Identificación de las articulaciones y sus motores cuando no son articulaciones prismáticas.
- Localizar los ejes de los motores.
- Agrupar los motores consecutivos con ejes paralelos
- Identificar las características de los planos y ángulos de ataque deseados.
- Resolver las agrupaciones por separado atendiendo a sus elementos comunes de ajuste (distancias, ángulos, relaciones proporcionales, etc). En caso de tener ejes paralelos, resolver los dos últimos predefiniendo valores para los motores anteriores. Y en caso de tener varios motores a predefinir, hay que resolverlos como una única unidad aplicando, si es necesario, el mismo modelo de desacoplo de motores de forma iterativa.

A la hora de calcular los parámetros de Denavit-Hartenberg, se debe determinar el ángulo de giro alrededor de Z (θ_i). Cuando este ángulo es distinto de nulo, se deben definir la matrices directa e inversa, y hacer los correspondientes cálculos con estas matrices. Así, se usarán las siguientes conversiones y reglas trigonométricas:

- La suma de dos ángulos: $\cos(A+B)=\cos(A)\cos(B)-\sin(A)\sin(B)$ y $\sin(A+B)=\sin(A)\cos(B)+\cos(A)\sin(B)$.

Los elementos incluidos en estas matrices serían:

$$\begin{aligned}\cos(\pi/2+\theta_2) &= \cos(\pi/2)\cos(\theta_2)-\sin(\pi/2)\sin(\theta_2)=0*\cos(\theta_2)-1*\sin(\theta_2)=-\sin(\theta_2) \\ \sin(\pi/2+\theta_2) &= \sin(\pi/2)\cos(\theta_2)+\cos(\pi/2)\sin(\theta_2)=1*\cos(\theta_2)+0*\sin(\theta_2)=\cos(\theta_2)\end{aligned}$$

Y en el caso del siguiente ejemplo:

$${}^1A_2 = \begin{pmatrix} C(\pi/2+\theta_2) & 0 & S(\pi/2+\theta_2) & r_2 C(\pi/2+\theta_2) \\ S(\pi/2+\theta_2) & 0 & -C(\pi/2+\theta_2) & r_2 S(\pi/2+\theta_2) \\ 0 & 1 & 0 & d_2 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$[{}^1A_2]^{-1} = \begin{pmatrix} C(\pi/2+\theta_2) & S(\pi/2+\theta_2) & 0 & -r_2 \\ 0 & 0 & 1 & -d_2 \\ S(\pi/2+\theta_2) & -C(\pi/2+\theta_2) & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

...las matrices 1A_2 y $[{}^1A_2]^{-1}$ anteriores se convierten en:

$${}^1A_2 = \begin{pmatrix} -S\theta_2 & 0 & C\theta_2 & -r_2 S\theta_2 \\ C\theta_2 & 0 & S\theta_2 & r_2 C\theta_2 \\ 0 & 1 & 0 & d_2 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad [{}^1A_2]^{-1} = \begin{pmatrix} -S\theta_2 & C\theta_2 & 0 & -r_2 \\ 0 & 0 & 1 & -d_2 \\ C\theta_2 & S\theta_2 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

- La resta de dos ángulos: $\cos(A-B)=\cos(A)\cos(B)+\sin(A)\sin(B)$, $\sin(A-B)=\sin(A)\cos(B)-\cos(A)\sin(B)$.

Los elementos incluidos en estas matrices serían:

$$\begin{aligned}\cos(-\pi/2+\theta_2) &= \cos(\pi/2)\cos(\theta_2)+\sin(\pi/2)\sin(\theta_2)=0*\cos(\theta_2)+1*\sin(\theta_2)=\sin(\theta_2) \\ \sin(-\pi/2+\theta_2) &= \sin(\pi/2)\cos(\theta_2)-\cos(\pi/2)\sin(\theta_2)=1*\cos(\theta_2)-0*\sin(\theta_2)=\cos(\theta_2)\end{aligned}$$

Y en el caso del siguiente ejemplo:

$${}^0A_1 = \begin{pmatrix} C(\theta_1-\pi/2) & 0 & -S(\theta_1-\pi/2) & r_1 C(\theta_1-\pi/2) \\ S(\theta_1-\pi/2) & 0 & C(\theta_1-\pi/2) & r_1 S(\theta_1-\pi/2) \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$[{}^0A_1]^{-1} = \begin{pmatrix} C(\theta_1 - \pi/2) & S(\theta_1 - \pi/2) & 0 & -r_1 \\ 0 & 0 & -1 & 0 \\ -S(\theta_1 - \pi/2) & C(\theta_1 - \pi/2) & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

...las matrices 0A_1 y $[{}^0A_1]^{-1}$ anteriores se convierten en:

$${}^0A_1 = \begin{pmatrix} S\theta_1 & 0 & C\theta_1 & r_1 S\theta_1 \\ -C\theta_1 & 0 & S\theta_1 & -r_1 C\theta_1 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad [{}^0A_1]^{-1} = \begin{pmatrix} S\theta_1 & -C\theta_1 & 0 & -r_1 \\ 0 & 0 & -1 & 0 \\ C\theta_1 & S\theta_1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

5.4 Casos planteados y resueltos

Casos analizados – Motor de cada articulación respecto al anterior y a la base

	Arti-1	Arti-2	Arti-3	Arti-4	Arti-5	
Caso 3-0	Perpen	Perpen				DH
Caso 3-1	Perpen	Perpen	Paralelo			DH+Geométrico
Caso 4-0	Paralelo	Perpen	Paralelo			Cuadrúpodo resuelto por DH
Caso 4-1	Paralelo	Perpen	Paralelo			Cambio de ejes sobre Caso 4-0
Caso 4-2	Paralelo	Paralelo	Paralelo			
Caso 5-0	Perpen	Perpen	Perpen	Paralelo		
Caso 1-0...	Perpen	Perpen	Perpen	Paralelo	Paralelo	
... Arti. 1	Perpen	Fija	Fija	Fija	Fija	DH
... Arti. 2 *	Fija	Perpen	Fija	Fija	Fija	DH-Sin giro de 90°
... Arti. 2	Fija	Perpen	Fija	Fija	Fija	DH-Con giro de 90°
... Arti. 3	Fija	Fija	Perpen	Fija	Fija	DH
... Arti. 4	Fija	Fija	Fija	Paralelo	Fija	DH
... Arti. 5	Fija	Fija	Fija	Fija	Paralelo	DH
... Arti. 1-2	Perpen	Perpen	Fija	Fija	Fija	DH
... Arti. 2-3 *	Fija	Perpen	Perpen	Fija	Fija	DH-Sin giro de 90°
... Arti. 2-3	Fija	Perpen	Perpen	Fija	Fija	DH-Con giro de 90°
... Arti. 3-4	Fija	Fija	Perpen	Paralelo	Fija	DH
... Arti. 4-5	Fija	Fija	Fija	Paralelo	Paralelo	DH
... Arti. 1-2-3	Perpen	Perpen	Perpen	Fija	Fija	DH
... Arti. 1-3-4	Perpen	Fija	Perpen	Paralelo	Fija	DH
... Arti. 2-3-4	Fija	Perpen	Perpen	Paralelo	Fija	DH
... Arti. 3-4-5	Fija	Fija	Perpen	Paralelo	Paralelo	DH
... Arti. 1-2-3-4	Perpen	Perpen	Perpen	Paralelo	Fija	DH
... Arti. 2-3-4-5	Fija	Perpen	Perpen	Paralelo	Paralelo	DH
... Arti. Desacopladas 1-2 + 3-4-5	Perpen	Perpen	Perpen	Paralelo	Paralelo	Brazo resuelto por DH y Acoplamiento, con plano y ángulo de ataque
Caso 7-0-Pierna	Perpen	Paralelo	Paralelo	Paralelo	Perpen	Humanoide resuelto por DH
Caso 7-0-Brazo	Perpen	Paralelo	Paralelo			Humanoide resuelto por DH

6 Trabajos relacionados

Debido a que desde la revolución industrial se ha disparado el uso de los brazos robóticos, existe una gran cantidad de estudios y trabajos relacionados con este TFG. En este punto se analizan dos tipos de trabajos relacionados; aquellos relacionados con la funcionalidad y la implementación y aquellos relacionados con el algoritmo para resolver los problemas de cinemática directa y cinemática inversa.

En cuanto a los utilizados en la fabricación masiva dentro de la industria, no hay información pública que aportar para su análisis. Y en cuanto a los ejemplos comerciales que se exponen, tampoco pueden ser analizados más allá de su apariencia, debido a la ausencia de información pública sobre ellos. No obstante, dan una idea de la similitud que tienen con este TFG.

De hecho, como elemento común en ambos casos, por las fotografías que pueden encontrarse, está la aparente linealidad de las articulaciones, o lo que es lo mismo, por encontrarse en una gran mayoría de casos, formando un plano único, y evitando así problemas de análisis y caracterización.

6.1 Relacionados con la funcionalidad

6.1.1 DOBI

DOBI es un bípedo con características similares al bípedo de ejemplo desarrollado en este TFG. Se desconoce cómo se calculan sus movimientos, pudiendo ser calculados, tanto trigonométricamente, como analíticamente, o numéricamente. Tiene un sistema de auto-estabilización que evita que se caiga.



Ilustración 37: DOBI Humanoid Intelligent Programmable RC Robot [WWWauselectronicsdirectDoc0]

6.1.2 Humanoide 17DOF

A continuación se muestra un ejemplo muy parecido al bípedo de ejemplo desarrollado en este TFG. Tiene una estructura y piezas muy similares. En este caso, no viene incluido con ningún soporte para el software por lo que se deja en manos del cliente su desarrollo.

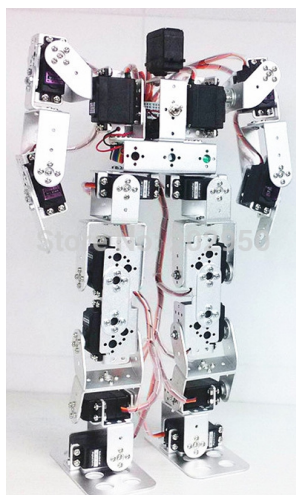


Ilustración 38: Robot educacional de 17DOF Biped [WWWaliexpressProd148]

Piezas utilizadas en el robot:



Ilustración 39: Humanoid Robotic Hand [WWWaliexpressProd148]

6.1.3 MarchProyect

Este proyecto es el menos parecido de los expuestos, pero da una visión distinta de las utilidades y distintas implementaciones que puede tener la robótica en nuestro día a día. Este proyecto tiene la forma de concurso, y en cada edición se presentan diferentes exoesqueletos para personas con daños en la médula espinal.



Ilustración 40: Exoesqueleto "MarchProject" [WWWprojectmarchDoc1]

Al fin y al cabo, no dejan de ser piernas robóticas que tienen que calcular de alguna manera el movimiento que simulan las extremidades, y se hacen de forma similar a como se ha desarrollado en el presente TFG.

6.1.4 Hexápodo *Phoenix* de Lynxmotion

Este ejemplo, al igual que lo anteriores, no explica el procedimiento usado para la obtención de las funciones que definen el movimiento. Es por tanto similar solo en cuanto a la funcionalidad. En el caso de este robot, sí que es público el código, pero no el cómo se llega a las ecuaciones que rigen el movimiento.

Por tanto, tener el código no aporta nada en este caso, ya que se han calculado las ecuaciones de forma externa al robot.



Ilustración 41: Hexápodo Phoenix de Lynxmotion [WWWaskixDoc0]

6.2 Relacionados con la implementación

6.2.1 Modelado cinemático directo de robot esférico

En este trabajo se resuelven problemas simples de brazos robóticos con cinemática directa y cinemática inversa. De hecho, los problemas resueltos sólo tienen tres articulaciones.

Resuelve de una forma similar la obtención de las funciones obtenidas a partir de las matrices. Pero no implementa los resultados en ningún robot, por tanto, no soluciona problemas derivados de su montaje.

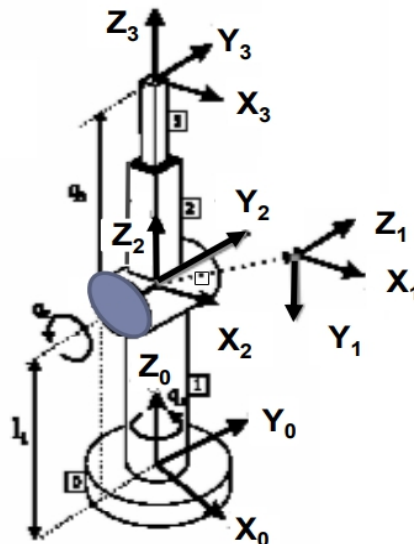


Ilustración 42: Modelado cinemático directo de robot esférico con 3 grados de libertad [WWWocw.ehu.eusDoc1]

6.2.2 Modelación Dinámica y Simulación de Robots

En este trabajo, la primera parte es similar al planteamiento teórico que se hace en este TFG, ya que calcula la cinemática directa usando el procedimiento de Denavit-Hartenberg. No obstante, a la hora de resolver la cinemática inversa, utiliza un algoritmo de Newton-Euler. Este algoritmo es un algoritmo recursivo que se plantea exclusivamente de forma teórica, sin ejemplos aplicados.

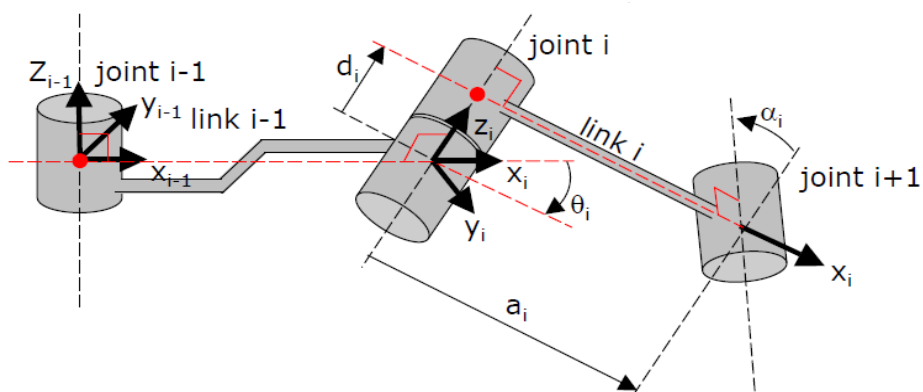


Ilustración 43: Algoritmo de Denavit-Hartenberg modificado (John J. Craig) [WWWmtorrest.sitios.ing.uc.clDoc0]

6.2.3 Forward Kinematics: The Denavit-Hartenberg convention

En este trabajo se realiza un análisis muy exhaustivo de los pasos de Denavit-Hartenberg y sobre su aplicación en la cinemática directa.

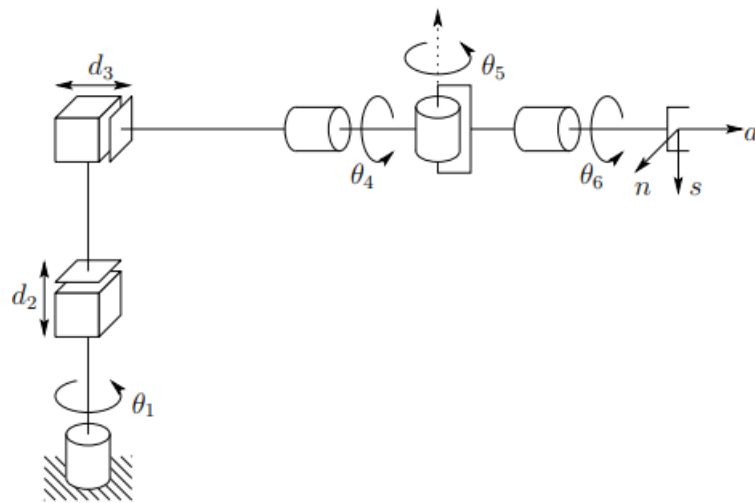


Ilustración 44: Forward Kinematics: The Denavit-Hartenberg convention [WWWusers.cs.duke.eduDoc0]

6.2.4 Comparación trabajos relacionados

Trabajo	<i>Modelado cinemático directo de robot esférico</i>	<i>Algoritmo de Denavit- Hartenberg modificado (John J. Craig)</i>	<i>Forward Kinematics: The Denavit- Hartenberg convention</i>	TFG
Utilizan DH	Sí	Sí	Sí	Sí
Resuelven CD	Sí	Sí	Si	Sí
Resuelven CI	Si	Sí	No	Sí
Nº de ejemplos resueltos CD	2	2	4	27
Nº de ejemplos resueltos CI	1	0	0	33
Ejemplos aplicados CD	No	No	No	Sí
Ejemplos aplicados CI	No	No	No	Sí
Nº máximo de articulaciones CD	4	6	4	Sin máximo*
Nº máximo de articulaciones CI	3	-	-	Sin máximo*

Tabla 3: Diferencias trabajos relacionados

* La única limitación se encuentra en la cantidad de recursos que puede asignar la máquina a esta tarea. Esto se debe a que, si un problema es complejo, la solución se basa en el desacoplamiento del problema.

Se puede observar como los trabajos analizados, más allá de su publicación anterior a este TFG y de alguna explicación más profunda, no tiene unas mejores características que este TFG, ya que en este TFG se aporta un gran análisis teórico con su correspondiente aplicación práctica.

6.3 Alternativas al algoritmo de Denavit-Hartenberg para resolución de problemas de Cinemática Inversa

Existen alternativas al cálculo analítico de Denavit-Hartenberg para la resolución de problemas de cinemática inversa. Todas las soluciones pueden ser clasificadas en métodos analíticos, numéricos y geométricos. Cada uno de estos métodos tiene ventajas e inconvenientes como se va a ver brevemente en este apartado.

Los métodos analíticos (el caso de este TFG) para la cinemática inversa son computacionalmente

rápidos y exactos, pero generalmente no están disponibles, por lo que hay que utilizar otras técnicas para solucionar los problemas que surgen (como el desacoplamiento). Los métodos numéricos son más disponibles pero no son tan exactos y rápidos. Los enfoques de redes neuronales artificiales como el "Multi-Layer Perceptron" (MLP) pueden clasificarse como algo intermedio. Si bien los modelos basados en redes neuronales pueden ser computacionalmente complejos, necesitan un tiempo relativamente alto de entrenamiento el cual aumenta rápidamente a la par que los grados de libertad. Los algoritmos genéticos como el "*Memetic Algorithm*" (MA) alcanzan una convergencia más rápidamente y no presentan dificultades al aumentar el número de grados de libertad.

Afrontar de forma geométrica este problema, si bien es como se ha hecho tradicionalmente, tienen una gran complejidad en cuanto el número de grados de libertad crece mínimamente.

A continuación, se puede ver una comparativa de algunas propiedades basada en algunos de los algoritmos de resolución de Cinemática Inversa. [WWWzalo.github.ioDoc0]

Properties of Various IK Algorithms

IK Algorithms	Analytic	Automatic Differentiation	Jacobian Transpose	FABRIK	Quaternion CCDIK
Implementation Complexity?	Extremely Complex	Hard	Hard	Easy	Easy
Speed	Extremely Fast	Slow To Converge	Slow To Converge	Fast	Fast
Hinge Joints	Only Hinges?	Yes	Yes	No!	Yes
Joint Limits	Difficult	Yes	No	Conical Limits	Yes
Hits Singularities	Never	Often	Often	Never (w/out hinges)	Rarely (often anneals through them)
Convergence Behaviour	Instant	Stable	Stable	Very Well Behaved	Well Behaved across short distances
Number of Joints	Max ~5	Arbitrary	Arbitrary	Arbitrary	Arbitrary

Tabla 4: Propiedades de varios Algoritmos de Cinemática Inversa [WWWplatea.pntic.mecDoc0]

7 Conclusiones y líneas de trabajo futuras

7.1 Conclusiones relacionadas con el proyecto

Existe una gran cantidad de trabajos realizados con Denavit-Hartenberg tanto para cinemática directa como para cinemática inversa. No obstante, no se han encontrado apenas ejercicios resueltos como se proponen en este documento y sus anexos. Además las resoluciones de cinemática inversa que se han encontrado son simples y muy repetidos y referenciados.

El cálculo de la cinemática inversa ha resultado más complejo y costoso de lo esperado, por lo que finalmente se ha tenido que utilizar un motor de cálculo potente (*MATLAB*), obligando a dejar de lado la idea inicial que pretendía que se calculase todo dentro del propio Arduino (*PORTENTA*, *MKRxxx*). Además ha resultado ser necesario un mayor conocimiento por parte del usuario del programa y los datos que devuelve, así como un mayor conocimiento general en el campo de las matemáticas. Sin estos conocimientos es difícil interpretar las soluciones dadas por los programas.

No obstante, se ha hecho una importante aportación a este campo creando una Herramienta y proponiendo una Metodología, aplicables a todos los casos, y que simplifican de manera notable la tarea de encontrar las ecuaciones que caracterizan cualquier brazo robótico, tanto para la cinemática directa, como inversa. De hecho, permite evitar el costoso calculo trigonométrico con el que, tradicionalmente, se han solucionado estos problemas.

Como prueba de ello, entre los 27 ejercicios realizados, de mayor o menor complejidad, en el ejemplo de brazo robótico resuelto y presentado en este TFG (“Ejemplo 1-0” de la Documentación Técnica), se aplica precisamente el método de Desacoplamiento junto con Plano y Ángulo de Ataque, y funciona correctamente.

Un segundo robot, un Humanoide con piernas con 5 motores, también ha sido resuelto, en este caso, mediante condiciones añadidas, como la necesaria relación de paralelismo de los pies con el suelo al pisar para mantener el equilibrio necesario.

Y, como tercer problema resuelto, también con condiciones añadidas, el “Ejemplo 4-2” de la Documentación Técnica, presenta un brazo con 4 motores.

7.2 Líneas de trabajo futuras

Se pueden diferenciar dos líneas de trabajo futuras principales. Ambas líneas están relacionadas con la profesionalización de la herramienta. La primera se dirige hacia la realización de mejoras de lo ya existente para aumentar la experiencia del usuario, y la segunda estaría relacionada con profundizar y mejorar los resultados ofrecidos.

Siendo los módulos implementados, programas para la comprobación de la teoría (aunque éste no fuese su objetivo), hay varias mejoras en cuanto a la usabilidad. Sería interesante introducir una interfaz amigable que facilite el uso al usuario. No haría falta cambiar de lenguaje ya existente ya que hay *frameworks* en C que facilitan la creación de botones, paneles... Uno de estos *frameworks* es por ejemplo GTK pero existen muchos más. Además, habría que clarificar algunas preguntas, así como mejorar el almacenaje y exposición de los resultados, tanto los parciales, como el resultado final.

En cuanto al carácter más técnico de TFG, sería interesante ser capaces de detectar cuales son los elementos de la Matriz de Transformación Homogénea derivada de la cinemática inversa que puedan ser utilizados para generar las ecuaciones que caractericen el brazo robótico, así como la existencia de agrupaciones de los motores para el desacoplamiento, y todo ello de forma automática. Esta posibilidad está ya al alcance dado que la Metodología aportada da la guía a seguir

para su codificación, y las pruebas realizadas los ejemplos y sistema de pruebas necesarios. Esta mejora permitiría facilitar el cálculo de las esas ecuaciones al máximo, siendo probablemente una idea muy atractiva para empresas de distintos sectores.

Por otro lado, con el planteamiento teórico ya existente, se puede hacer un prototipo de piernas para una personas con cierto grado severo de discapacidad física. Este proyecto sería muy ambicioso, pero sería de gran interés social. Uno de los principales problemas sería el gran coste de unos motores capaces de sujetar tanto peso en el arranque, así como del resto de piezas necesarias. Sin embargo, en esta línea, las pruebas realizadas con la generación de trayectorias del Humanoide presentado y los sensores de equilibrado ya probados, parece apuntar a que esta posibilidad es factible en un corto espacio de tiempo.

8 Referencias Bibliográficas

Bibliografía

1. WWWdlscribDoc0: Richard P Paul, Robot manipulators: mathematics, programming, and control - The computer control of robot manipulators, 1981, https://dlscrib.com/queue/mit-press-series-in-artificial-intelligence-richard-p-paul-robot-manipulators-mathematics-programming-and-control-the-computer-control-of-robot-manipulators-mit-press-1981pdf_5a39d5b4e2b6f59d048c5f70_pdf?queue_id=5a39d5c3e2b6f56a2f8c4002
2. WWWUVADoc1: Carlos Martínez Pablos, Modelado Cinemático inverso y seguimiento de trayectorias de un manipulador BHR (FABRIK), 12-1-2023, <https://uvadoc.uva.es/bitstream/handle/10324/45449/TFG-I-1778.pdf;jsessionid=0ECBAC32E739366222BC861106B6C19A?sequence=1>
3. WWWTeklaDoc0: Trimble, Regla de la mano derecha, 2022, http://www.wag.caltech.edu/home/ajaramil/libro_robotica/cinematica.pdf
4. WWWjsvanegascDoc1: Ingenius Military, Regla de la mano derecha, 11-9-2023, <https://jsvanegasc.wordpress.com/fisica-ecci/1-corte/regla-de-la-mano-derecha-2/>
5. WWWwikipediaDoc0: Wikipedia, Cuadrante Coordenadas, 15-1-2023, https://es.wikipedia.org/wiki/Sistema_de_coordenadas
6. WWWwebs.ucmDoc0: Jon Legarreta / Raquel Martinez, TRIGONOMETRÍA, 2022, <https://webs.ucm.es/info/Geofis/practicas/trigonometria.htm>
7. WWWsuperprofDoc0: Jon Legarreta / Raquel Martinez, Ejercicios resueltos sobre las ecuaciones trigonométricas, 2022, <https://www.superprof.es/apuntes/escolar/matematicas/trigonometria/ejercicios-resolver-las-ecuaciones-trigonometricas.html>
8. WWWlaplace.usDoc0: Jon Legarreta / Raquel Martinez, Tabla de fórmulas de trigonometría, 2022, http://laplace.us.es/wiki/index.php/Tabla_de_fórmulas_de_trigonometría
9. WWWwikipediaDoc18: Empresa, List of trigonometric identities, 2022, https://en.wikipedia.org/wiki/List_of_trigonometric_identities
10. WWWyoutubeDoc84: Empresa, Análisis y cálculo de cinemática inversa "EIR2015: Cinemática directa e inversa -- parte 3, 9-01-2023, <https://www.youtube.com/watch?v=wMTjrFN6CG0>
11. WWWdocplayerDoc0: Empresa, Introducción. TEMA 2. MORFOLOGÍA, 20-03-2020, <https://docplayer.es/16277041-Introduccion-tema-2-morfologia.html>
12. WWWslideshareDoc2: Empresa, Cinemática de los manipuladores, 20-03-2020, <https://es.slideshare.net/omarspp/cinemtica-de-los-manipuladores>
13. WWWdocplayerDoc1: Empresa, Morfología del Robot, 20-03-2020, <https://docplayer.es/7407135-Morfologia-del-robot.html>
14. WWWdocplayerDoc7: Dr. José Antonio Garrido Natarén, José Antonio Hernandez Reyes, MC. José Luis Fernando Palomeque Loyo, Carlos Armando Gutiérrez González, ESTUDIO DE UNA CADENA CINEMÁTICA UTILIZANDO LOS MÉTODOS DENAVIT-HARTENBERG Y CRAIG MODIFICADO, 2022, <https://docplayer.es/95237332-Estudio-de-una-cadena-cinematica-utilizando-los-metodos-denavit-hartenberg-y-craig-modificado.html>

15. WWWdiag.uniroma1Doc16: Richard P. Paul and Bruce E. Shimano, Kinematic Control Equations for Simple Manipulators, 1978, http://www.diag.uniroma1.it/~deluca/rob1_en/Article_KinInvPuma600.pdf
16. WWWocw.ehu.eusDoc1: Jon Legarreta / Raquel Martinez, TEMA 5. MODELADO GEOMÉTRICO Y CINEMÁTICO DEL ROBOT, 2022, https://ocw.ehu.eus/pluginfile.php/50445/mod_resource/content/8/T5%20CINEMATICA%20OCW_Revision.pdf
17. BookBarrientosDoc0: Antonio Barrientos, Capítulo 4 - Fundamentos de robótica -- DENAVIT & HARTENBERG, 2007
18. WWWyoutubeDoc76: Antonio Barrientos, Rob 3 5 Matrices de Transformacion Homogenea, 2020, <https://www.youtube.com/watch?v=Y1GWG923vE4>
19. WWWocw.ehu.eusDoc0: Jon Legarreta / Raquel Martinez, TEMA 4. HERRAMIENTAS MATEMÁTICAS PARA LA LOCALIZACIÓN ESPACIAL, 2022, https://ocw.ehu.eus/pluginfile.php/50444/mod_resource/content/4/T4%20herramientas%20matemáticas%20para%20la%20localizació%20espacial_OCW_Revision.pdf
20. WWWkramirez.netDoc0: Kryscia Ramírez Benavides, Cinemática Inversa del Robot, 2022, <https://www.kramirez.net/Robotica/Material/Presentaciones/CinematicaInversaRobot.pdf>
21. WWWwag.caltech.eduDoc0: Andrés Jaramillo Botero, Cinemática de Manipuladores Robóticos, 2022, http://www.wag.caltech.edu/home/ajaramil/libro_robotica/cinematica.pdf
22. WWWyoutubeDoc89: Alessandro De Luca, Robotics 1 - Prof. De Luca Lecture 12 (24 Oct 2014) - Minimal representations of orientation (Euler and roll-pitch-yaw angles) - Homogeneous transformations, 2014, <https://www.youtube.com/watch?v=ZAhsnfnM88o&list=PLAQopGWllcyaqDBW1zSKx7IHfVcOmWSWt&index=16>
23. WWWyoutubeDoc77: Empresa, Cinemática inversa de un robot de 6 GDL PRR+RRR (solución paso a paso), 2022, https://www.youtube.com/watch?v=ooMc_V83xRs
24. WWWatlassianDoc0: Atlassian, Qué es Git, 11-9-2023, <https://www.atlassian.com/es/git/tutorials/what-is-git>
25. WWWGitDoc1: Git, Git, 11-9-2023, <https://git-scm.com/images/logos/downloads/Git-Logo-2Color.png>
26. WWWgithubDoc1: GitHub, GitHub, 11-9-2023, <https://github.com/logos>
27. WWWgitlabDoc1: GitLab, GitLab, 11-9-2023, <https://about.gitlab.com/blog/2015/07/03/our-new-logo/>
28. WWWSourceForgeDoc1: SourceForge, SourceForge, 11-9-2023, <https://sourceforge.net/p/forge/documentation/SourceForge%20Logo/>
29. WWWkinstaDoc: Salman Ravoof, GitLab vs GitHub: Descubre Sus Principales Diferencias y Similitudes, 11-9-2023, <https://kinsta.com/es/blog/gitlab-vs-github/>
30. WWWMetodologíaScrumDoc0: Redacción APD, Cómo aplicar la metodología Scrum y qué es el método Scrum, 11-9-2023, <https://www.apd.es/metodologia-scrum-que-es/#:~:text=La%20metodología%20Scrum%20es%20un,resultado%20de%20un%20proyecto%20determinado.>
31. WWWganttproDoc1: Anastasia Stsepanets, ¿Cómo gestionar trabajo y proyectos complejos de manera fácil y eficiente?, 11-9-2023, <https://blog.ganttpro.com/es/marco-scrum-metodologia-agil/>
32. WWWsoftonicDoc0: Softonic, Entorno de desarrollo multilenguaje ligero pero potente,

11-11-2023, <https://geany.softonic.com/>

33. WWWatmosDoc0: Alexey Zimin, Introduction to Programming in Matlab with MEX, 11-11-2023, https://www2.atmos.umd.edu/~ide/data/teaching/amsc663/14fall/amsc663_141118_mex_aleksey_zimin.pdf
34. WWWxatakaDoc0: Yúbal Gernández, Qué es Arduino, cómo funciona y qué puedes hacer con uno, 11-1-2023, <https://www.xataka.com/basics/que-arduino-como-funciona-que-puedes-hacer-uno>
35. WWWyoutube0: Noah Del Coro, Finger Scanning Garage Door opener!, 11-1-2023, https://www.youtube.com/watch?v=uK8gm9a0k-g&t=1s&ab_channel=NoahDelCoro
36. WWWmathworksDoc0: Robotics System Toolbox, Robotics System Toolbox, 11-11-2023, <https://es.mathworks.com/products/robotics.html>
37. WWWmathworksDoc1: Robotics System Toolbox, Symbolic Math Toolbox, 11-11-2023, <https://es.mathworks.com/products/symbolic.html>
38. WWWRobotics-ToolboxDoc0: Peter Corke, Robotics Toolbox, 11-11-2023, <https://petercorke.com/toolboxes/robotics-toolbox/>
39. WWWRobotics-ToolboxDoc1: Peter Corke, Robotics Toolbox, 11-11-2023, https://es.mathworks.com/videos/robotics-system-toolbox-history-and-development-123408.html?s_tid=srchtitle
40. WWWauselectronicsdirectDoc0: Empresa, DOBI HUMANOID INTELLIGENT PROGRAMMABLE RC ROBOT, 20-03-2020, <https://www.auselectronicsdirect.com.au/dobi-humanoid-intelligent-programmable-rc-robot>
41. WWWaliexpressProd148: Empresa, Robot educacional de 17DOF Biped, 20-03-2020, <https://es.aliexpress.com/item/32645309538.html?spm=a2g0s.9042311.0.0.274263c0daoCce>
42. WWWprojectmarchDoc1: Empresa, MarchProject, 20-03-2020, <https://twitter.com/projectmarch>
43. WWWaskixDoc0: empresa, Arduino Mega Hexápodo, 9-01-2023, <https://www.askix.com/arduino-mega-hexapodo.html>
44. WWWmtorrest.sitios.ing.uc.clDoc0: Miguel Torres Torriti, Modelación Dinámica y Simulación de Robots, 10-08-2022, https://mtorrest.sitios.ing.uc.cl/downloads/robotics_tutorial/pres_robotics_summer_school_111214.pdf%20
45. WWWusers.cs.duke.eduDoc0: Desconocido, Modelación Dinámica y Simulación de Robots, 9-1-2023, <https://users.cs.duke.edu/~brd/Teaching/Bio/asmb/current/Papers/chap3-forward-kinematics.pdf>
46. WWWzalo.github.ioDoc0: , Inverse Kinematics, 2022, <https://zalo.github.io/blog/inverse-kinematics/>
47. WWWplatea.pntic.mecDoc0: Empresa, Actuadores, 20-03-2020, http://platea.pntic.mec.es/vgonzale/cyr_0204/cyr_01/robotica/sistema/actuadores.htm