



EDUCACIÓN
SECRETARÍA DE EDUCACIÓN PÚBLICA



TECNOLÓGICO
NACIONAL DE MÉXICO®

SECRETARÍA DE EDUCACIÓN PÚBLICA
TECNOLÓGICO NACIONAL DE MÉXICO

INSTITUTO TECNOLÓGICO DE VERACRUZ

**OBTENCIÓN DE LAS ECUACIONES CINEMÁTICAS DE UN ROBOT
INDUSTRIAL MARCA KUKA UTILIZANDO PROGRAMACIÓN PYTHON.**

TITULACIÓN INTEGRAL
INFORME TÉCNICO DE RESIDENCIA PROFESIONAL

CARRERA:
INGENIERÍA MECATRÓNICA

PRESENTA:
CARLOS ALI ALTAMIRANO CERVANTES

ASESOR INTERNO:
DR. JOSE ANTONIO GARRIDO NATAREN

H.VERACRUZ.VER

JUNIO 2023



Resumen

El Tecnológico Nacional de México Campus Veracruz ha llevado a cabo un proyecto enfocado en la obtención de ecuaciones cinemáticas del robot KR 6 R900 SIXX, que se encuentra en el plantel.

El objetivo principal del proyecto fue obtener dichas ecuaciones para calcular la posición y orientación del elemento terminal. Para lograrlo, se ha realizado un proceso de análisis y modelado matemático detallado, además de utilizar software de programación en ambiente Python que permite la fácil visualización de ecuaciones matemáticas entre las cuales, multiplicación, rotación y translación de matrices.

La obtención de estas ecuaciones cinemáticas KR 6 R900 SIXX tiene una gran importancia en el ámbito de la robótica y la automatización, ya que permiten mejorar la eficiencia y precisión de las operaciones realizadas por el robot. Además, este proyecto demuestra la capacidad y el compromiso del Tecnológico Nacional de México Campus Veracruz con la investigación y la innovación tecnológica en este campo.

Índice general

Resumen	II
Índice de figuras	v
Índice de cuadros	vi
1. PLANTEAMIENTO GENERAL DEL PROYECTO	1
1.1. Tecnológico Nacional de de México Campus Veracruz	2
1.1.1. Misión y Visión	3
1.1.2. Cultura y Valores	4
1.2. Problemática	5
1.3. Objetivos	6
1.3.1. Objetivos Generales	6
1.3.2. Objetivos Específicos	6
1.4. Justificación	7
2. FUNDAMENTO TEÓRICO	8
2.1. Antecedentes del Proyecto	9
2.2. Marco Teórico	10
2.2.1. La historia de la robótica	10
2.2.2. Componentes mecánicos de un robot	13
2.2.3. Conceptos Generales	14
2.2.4. Cinemática Directa	18
2.2.5. Python	24
2.2.6. Jupyter-Lab	28
2.2.7. Cinemática Inversa	30
2.2.8. Ángulos de Euler	32

3. PROCEDIMIENTO Y DESCRIPCIÓN DE LAS ACTIVIDADES REALIZADAS	34
3.1. Descripción de la metodología	35
3.1.1. Curso de programación del robot industrial marca KUKA	35
3.2. Estudio de cinemática directa	36
3.2.1. Estudio de cinemática inversa	47
3.3. Actividades Complementarias	50
3.3.1. Colocación del espacio de trabajo	50
3.3.2. Exhibición de resultados	51
4. RESULTADOS	52
4.1. Obtención de la cinemática directa	53
4.2. Obtención de la cinemática inversa	54
Conclusiones, propuesta de mejora y experiencia profesional adquirida	55
4.3. Experiencia profesional adquirida	57
Competencias desarrolladas y/o Aplicadas por el residente	58
Fuentes de información	59

Índice de figuras

1.1. Logo ITV	2
2.1. Tipos de cadenas cinemáticas	13
2.2. Articulacion de revoluta	15
2.3. Articulación prismática	15
2.4. Espacio de Trabajo	16
2.5. Tipos de arquitecturas de 3 Grados de libertad	17
2.6. Grados de libertad de un eslabón en el espacio	18
2.7. Diagrama de relación entre cinemática directa e inversa.	18
2.8. Asociación de Sistemas de referencia	21
2.9. Logo de Numpy.org	25
2.10. Entorno Jupyter	29
2.11. Representación geométrica del péndulo de 1 gdl	31
2.12. Ángulos de Euler a) de dos ejes y b) de tres ejes	33
3.1. SmartPad	35
3.2. Unidad de Control KR C4 Compact	35
3.3. Gráfico del espacio de trabajo	36
3.4. Sistema cinemático de robot manipulador KR6 R900 Sixx	36
3.5. Matrices de Transformación en el entorno Jupyter	39
3.6. Brazo Manipulador KR6 R900 Sixx en posición Home	47
3.7. Simplificación del brazo manipulador	48
3.8. Zona de trabajo del robot manipulador	50
3.9. Jovenes del CBTIS durante XII Feria de carreras 2023	51
3.10. Jovenes del CONALEP durante XII Feria de carreras 2023	51
4.1. Sistemas del robot KR6 R900 Sixx	53

Índice de cuadros

2.1. Autómatas famosos	9
2.2. Estructuras posibles de robots dependiendo de los grados de libertad.	17
2.3. tabla de los parámetros D-H	20
2.4. Palabras reservadas de Python	24
2.5. Tabla parámetros péndulo invertido	31
3.1. Tabla de los parámetros D-H del robot manipulador KR6 R900 Sixx	37

Capítulo 1

PLANTEAMIENTO GENERAL DEL PROYECTO

1.1. Tecnológico Nacional de de México Campus Veracruz

El Instituto Tecnológico de Veracruz, fundado el 14 de marzo de 1957, es una institución de educación superior en México que ofrece programas de estudios vocacionales en ciencias físico-matemáticas enfocados en las carreras de ingeniería mecánica, eléctrica, electrónica, industrial y química. A lo largo de los años, el instituto ha ampliado su oferta académica a través de la apertura de las carreras de ingeniería bioquímica, ingeniería mecatrónica, ingeniería en energías renovables e ingeniería en sistemas computacionales, también ofrece programas de maestría en bioingeniería y biotecnología. En 1985, el instituto obtuvo el tercer lugar en el Premio de Ciencia y Tecnología de los Alimentos en la categoría profesional y en 1986 se implementó el sistema de créditos con programas por unidades de aprendizaje. En resumen, el Instituto Tecnológico de Veracruz es una institución de educación superior que ha evolucionado y se ha adaptado a las necesidades cambiantes de la comunidad educativa y del sector productivo en México [1].

Domicilio: Calz. Miguel Angel de Quevedo 2779 Col. Formando Hogar, Veracruz, Ver. MÉXICO CP 91897

Giro: Educativo

Sector: Publico

Teléfono: (229) 934 15 00



Figura 1.1: Logo ITV

1.1.1. Misión y Visión

El Tecnológico Nacional de México Campus Veracruz tiene como misión "Formar profesionales en tecnologías, capaces de dominar, generar y divulgar el conocimiento científico y tecnológico de vanguardia, desde una perspectiva humanista, con compromiso de trabajo, respeto por el medio ambiente, capaces de responder de manera eficaz, a las necesidades y retos nacionales con calidad, productividad y una visión global" [2].

"Antorcha y Luz de fuego permanente"

El Sistema Nacional de Educación Superior Tecnológica, estableció su visión de "Ser uno de los pilares fundamentales del desarrollo sostenido, sustentable y equitativo de la nación", en concordancia con esta visión, el Instituto Tecnológico de Veracruz ha definido su visión hacia el año 2030 en los términos siguientes:

"Ser una institución líder y de experiencia en educación superior tecnológica del país, competitiva de clase mundial, reconocida por la calidad y espíritu de trabajo de su personal, alumnos y egresados y por su compromiso hacia el desarrollo tecnológico, científico y socialmente responsable e innovadora de nuestro país" [2].

1.1.2. Cultura y Valores

Los valores son esenciales para el Instituto Tecnológico de Veracruz, ya que sirven como guía para todas las personas que forman parte de la institución en su desempeño diario, entre ellos destacan la calidad, el respeto a la diversidad, lealtad, honestidad, responsabilidad y compromiso.

- La calidad es importante porque promueve una cultura de mejora continua y ayuda a mantener altos estándares en todos los procesos y actividades. El ser humano es el centro de la institución y es importante tratar a todos con respeto y dignidad, mientras que el espíritu de servicio ayuda a fomentar un ambiente de colaboración y trabajo en equipo.
- El respeto a la diversidad es esencial en una sociedad cada vez más globalizada, ya que ayuda a comprender y valorar las diferencias culturales, sociales y lingüísticas.
- La lealtad, honestidad, responsabilidad y compromiso son valores que ayudan a mantener la confianza y el respeto mutuo entre todas las partes involucradas. Estos valores son especialmente importantes en una institución educativa, donde se espera que los estudiantes y el personal académico se comporten de manera ética y responsable en todo momento.

1.2. Problemática

La donación de un robot industrial marca KUKA KR 6 R900 Sixx al Tecnológico Nacional de México - Campus Veracruz, presenta la oportunidad de adquirir conocimientos avanzados en programación y ecuaciones cinemáticas. Este robot, con sus 6 grados de libertad, requiere un dominio de las ecuaciones para el manejo preciso de la posición y orientación para utilizar de manera efectiva su elemento terminal en las tareas que puedan surgir.

El conocimiento de estas ecuaciones cinemáticas resulta fundamental, pero también presenta un desafío obtenerlas si no se cuenta con un software como MATLAB. Es necesario comprender a fondo los conceptos y las fórmulas asociadas con las ecuaciones cinemáticas para poder aplicarlas de manera correcta en otro tipo de lenguaje como Python.

1.3. Objetivos

Se han establecido objetivos claros que se pretenden alcanzar durante la realización de este proyecto. Estos objetivos están diseñados para alcanzar el desarrollo y el éxito del proyecto, abarcando diversos aspectos clave que se deben lograr.

1.3.1. Objetivos Generales

El propósito principal de este estudio es obtener las ecuaciones cinemáticas del Robot Agilus 2 Modelo KR 6 R900 Sixx ubicado en el Laboratorio de Robótica e Inteligencia Artificial, utilizando el lenguaje de programación Python.

1.3.2. Objetivos Específicos

- Implementar un ambiente eficiente para el uso de un robot industrial.
- Realizar pruebas con el robot para obtener datos precisos sobre su comportamiento y rendimiento.
- Calcular la cinemática directa e inversa mediante programación Python.

1.4. Justificación

El proyecto de residencias que se enfoca en la obtención de las ecuaciones cinemáticas robot KR 6 R900 Sixx mediante Python es de gran importancia, ya que permitirá obtener ecuaciones, multiplicar matrices de 6 grados de libertad para personas que no cuentan con el software MATLAB y verificar los resultados obtenidos. La obtención de estas ecuaciones permite modelar el comportamiento del robot en diferentes situaciones y realizar simulaciones para predecir su comportamiento en escenarios específicos, lo que refuerza los conocimientos de los estudiantes acerca de robots que se utilizan en las empresas.

El Tecnológico Nacional de México - Campus Veracruz se enfrenta al reto de capacitar a su personal y proporcionarles las herramientas y los recursos necesarios para adquirir las habilidades requeridas en la programación y manipulación del robot industrial KUKA.

Capítulo 2

FUNDAMENTO TEÓRICO

2.1. Antecedentes del Proyecto

A lo largo de la historia, los seres humanos han sentido un gran interés por las máquinas y dispositivos que pueden imitar las funciones y movimientos de los seres vivos. En la antigua Grecia, a estas máquinas se les llamaba “automatos”, y de ahí proviene la palabra “autómata” que conocemos hoy en día. Un autómata es una máquina que imita la apariencia y los movimientos de un ser animado [10].

Durante los siglos XV y XVI, algunos de los representantes más destacados del Renacimiento también se interesaron por los ingenios descritos y desarrollados por los griegos. Un ejemplo conocido es el “León mecánico” construido por Leonardo Da Vinci para el rey Luis XII de Francia. Este autómata abría el pecho con su garra y mostraba el escudo de armas del rey. En España, otro ejemplo es el “hombre palo”, construido por Juanelo Turriano en el siglo XVI para el emperador Carlos V. Este autómata con forma de monje caminaba y movía la cabeza, los ojos, la boca y los brazos.

Entre los siglos VIII y XV, la cultura árabe heredó y difundió los conocimientos de los griegos, utilizándolos no solo para la diversión, sino también dándoles una aplicación práctica en la vida cotidiana de la realeza.

Año	Autor	Autómata
1352	Desconocido	Gallo de la catedral de Estrasburgo
1499	L. Da Vinci	León mecánico
1525	J. Turriano	Hombre de palo
1738	J de Vaucanson	Flautista, tamborilero, pato , muñecas mecánicas de tamaño humano
1768	W. Von Kempelen	Jugador de ajedrez
1770	Familia Droz	Escriba, organista, dibujante
1805	H. Maillardet	Muñeca mecánica capaz de dibujar

Cuadro 2.1: Autómatas famosos

Fuente: Libro [10]

2.2. Marco Teórico

2.2.1. La historia de la robótica

La palabra robot fue acuñada por el checo Karel Čapek, quien en 1921 presentó una obra de teatro donde aparecían humanos artificiales. Dado que en el idioma checo y en muchos idiomas eslavos la palabra “robota” significa “trabajo” o “servidumbre”, mostraba con el término a un ser artificial creado para servir a los seres humanos. Desde sus inicios los robots han fascinado y generado temor en el ser humano, gozando hoy en día de gran popularidad en el imaginario colectivo.

Sin embargo, a nivel científico y comercial, la historia de la robótica muestra una industria muy dinámica y variada, con aplicaciones que van desde el ensamblaje industrial, pasando por la exploración espacial o la robótica quirúrgica, hasta llegar a los robots humanoides de Honda o Sony, últimos desarrollos que podrían corresponder más al término original creado por Čapek. De manera general cualquier mecanismo que opere con cierta autonomía y controlado por computador podría ser llamado un robot. Sin embargo, la expresión clásica del término describe un manipulador mecánico con ciertas similitudes a un brazo humano y controlado por un computador.

A fines del siglo XIX, el mundo también había visto una variedad de “hombres de vapor”: humanoides andantes impulsados por motores de vapor internos (los modelos posteriores fueron impulsados por motores eléctricos o motores de gasolina) que parecían tirar de carruajes o flotadores. Muchos de estos fueron desarrollados en los Estados Unidos. A principios del siglo XX, se introdujeron enfermeras mecánicas, reinas de belleza y policías, y en sus escritos Makoto Nishimura, profesor de biología marina de 40 años en la Universidad Imperial de Hokkaido, también menciona recepcionistas mecánicos, operadores de botes y policías de tránsito. Es posible que haya visto algunas de estas máquinas mientras vivía en la ciudad de Nueva York entre 1916 y 1919 y realizaba un doctorado en la Universidad de Columbia [12].

La definición de un robot industrial según la norma internacional ISO 8373:2012 es “un manipulador multifuncional, re-programable y controlado automáticamente, programable en tres o más ejes que puede estar fijo en un área o móvil para su uso en aplicaciones de automatización industrial”. Los robots industriales no suelen tener forma humanoide, aunque son capaces de reproducir movimientos y comportamientos humanos, pero con la fuerza, precisión y rapidez de una máquina. Sin embargo, la definición de la ISO se ha quedado corta con la aparición en las últimas décadas de gran cantidad de aplicaciones robóticas en campos diferen-

tes a la industria. Por eso la Federación Internacional de Robótica (International Federation of Robotics) ha introducido el término de robot de servicios como “un robot que puede operar semi o completamente autónomo para llevar a cabo tareas útiles al bienestar de los humanos y al buen funcionamiento de ciertos equipos, excluyendo las operaciones de manufactura”. La pieza clave en la definición de la ISO es el hecho de que un robot pueda ser re-programado para ejecutar diversas tareas, gracias a la programación que su cerebro electrónico (computador) pueda generar. Desde este punto de vista la robótica no es más que un paso evolutivo de los computadores. A continuación, se muestran algunos de los hitos más importantes en la historia de la robótica:

- **1947:** Primer teleoperador eléctrico, inventado por George Devol.
- **1952:** La primera máquina NC (controlada numéricamente) se construye en el MIT
- **1954:** Los ingenieros estadounidenses George Devol y Joe Engleberger diseñan el primer robot programable ”brazo”, el primer industrial del mundo robot.
- **1956:** Investigadores estadounidenses Allen Newell, Herbert Simon y John Shaw crear el Teórico Lógico, la primera inteligencia artificial programa. George Devol y Joseph Engelberger crean la primera compañía de robots: Unimation, Inc.
- **1960:** Unimation is purchased by Condec Corporation and the development of Unimate Robot Systems begins. American Machine and Foundry, later known as AMF Corporation, markets the first cylindrical robot, called the Versatran.
- **1963:** Se diseña el primer brazo robótico controlado por computadora en Rancho Los Amigos Hospital.^{en} Downey, California como una herramienta para el minusválido
- **1963:** Es desarrollado el primer sistema de visión robotizada.
- **1971:** El robot Stanford es desarrollado en la Universidad de Stanford (Estados Unidos).
- **1976:** Brazos robot son utilizados en las misiones espaciales Viking I y II a Marte.
- **1978:** Unimation desarrolla PUMA (Máquina universal programable para Asamblea), basado en el diseño de Victor Scheinman.
- **1979:** El robot SCARA (Selective Compliant Articulated Robot for Assembly) es desarrollado en Japón.

- **1985:** El brazo quirúrgico robótico PUMA 560 es conocido el primer reporte de un procedimiento quirúrgico asistido por un robot.
- **1986:** El robot submarino Jason del Instituto Oceanográfico Woods Hole explora los restos descubiertos del transatlántico Titanic.
- **1988:** Es fundada la Sociedad de Robótica y Automática de la IEEE.
- **1996:** Honda desarrolla su proyecto de robot humanoide.
- **1997:** Se lleva a cabo el primer campeonato de fútbol con robots en Japón, donde participan 40 equipos de todo el mundo.
- **2001:** Sony comercializa exitosamente el primer robot para uso doméstico, el perro Aibo.
- **2002:** iRobot lanza el robot aspirador autónomo Roomba. El profesor británico de cibernética Kevin Warwick se convierte en el primero ciborg en el mundo mediante el control de dispositivos electrónicos con su sistema nervioso a través de una matriz de 100 electrodos implantada en su brazo.
- **2009:** Prótesis robotizada de brazo de la agencia DARPA de los Estados Unidos.
- **2011:** Robonaut 2, un asistente robótico similar a un humano desarrollado conjuntamente por NASA y General Motors, se lanza al espacio en transbordador espacial Discovery como parte de la misión STS-133 para convertirse en permanente Residente de la Estación Espacial Internacional.
- **2012:** Se otorga la licencia a el primer automóvil sin conductor en Nevada.
- **2014:** hitchBOT, un robot creado por un equipo de investigadores canadienses, "hace autostop.^a través de Canadá y Europa en una misión para explorar Actitudes culturales hacia la robótica social. hitchBOT es destruido en Filadelfia en 2015.

2.2.2. Componentes mecánicos de un robot

El mecanismo de un robot manipulador consta de dos subsistemas distintos, uno (o más) efectores finales y una estructura mecánica articulada.

1. Por el término “efector-final”, nos referimos a cualquier dispositivo destinado a manipular objetos (pinzas magnéticas, eléctricas o neumáticas) o a transformarlos (herramientas, soldadura, antorchas, pintura, armas, etc.). Constituye la interfaz con la cual el robot interactúa con su entorno. Un efector final puede ser multipropósito, es decir, estar equipado con varios dispositivos, cada uno con funciones diferentes.
2. El papel de la estructura mecánica articulada es colocar el efector-final en una ubicación dada (posición y orientación) con una velocidad y aceleración deseadas. La estructura mecánica está compuesta por una cadena cinemática de enlaces rígidos articulados. Uno de los extremos de la cadena está fijo y se llama la base. El efector-final está fijado al extremo libre de la cadena.

Las cadenas cinemáticas pueden ser: a) cadenas abiertas (o seriales), b) cadenas cerradas, c) cadenas arborescentes, o d) paralelas, como se muestra en la Figura 2.1 [3].

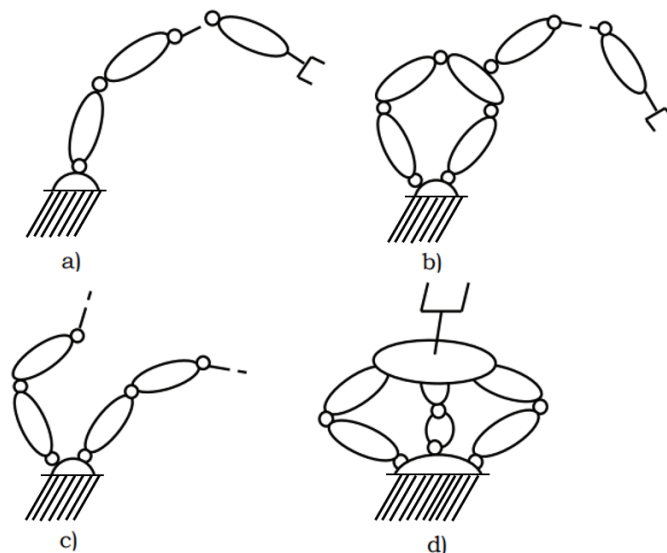


Figura 2.1: Tipos de cadenas cinemáticas

Fuente: Libro [3]

2.2.3. Conceptos Generales

A continuación, se enseñarán las principales definiciones en el campo de la robótica.

- **Mecanismo:** En cinemática, un mecanismo es un medio de transmisión, control o restricción del movimiento relativo. Los movimientos que están operados eléctricamente, magnéticamente, o neumáticamente se excluyen del concepto de mecanismo. el tema central de los mecanismos es eslabones conectados entre sí a través de articulaciones.
- **Máquina:** Es una combinación de eslabones (cuerpos rígidos), formados y conectados que se mueven con movimientos relativos definidos y transmiten fuerza desde la fuente de poder a la resistencia que hay que superar. Una maquina tiene dos funciones: transmisión de movimiento relativo definido y transmisión de fuerza.
- **Brazo:** Está formado por una serie de eslabones unidos entre sí, a través de articulaciones que permiten el movimiento de la muñeca.
- **Eslabón:** Es un cuerpo rígido. Es decir, la distancia entre dos puntos dentro del brazo no cambia permitiendo mantener unidas dos articulaciones.
- **Articulación:** El mecanismo es una conexión entre dos cuerpos consecutivos, impulsado por un motor. Las articulaciones predominantes son las rotoides, que permiten el movimiento de giro, y las prismáticas, que permiten el desplazamiento lineal. Sin embargo, también existen combinaciones de ambos tipos de articulaciones, así como articulaciones pasivas que reproducen cualquiera de estos dos movimientos sin la presencia de un motor.
- **Articulación de revoluta:** El movimiento de rotación sobre un eje común , se da entre dos eslabones. La ubicación relativa entre los dos enlaces está dada por el ángulo alrededor de este eje.

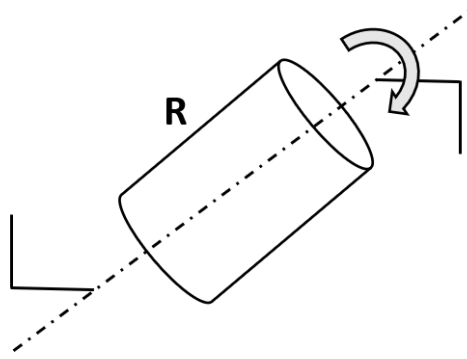


Figura 2.2: Articulación de revoluta

Fuente: Libro [5]

- **Articulación prismática:** El movimiento de traslación se lleva a cabo a lo largo del eje compartido por dos cuerpos. La posición relativa entre los dos cuerpos se determina mediante la distancia a lo largo de este eje.

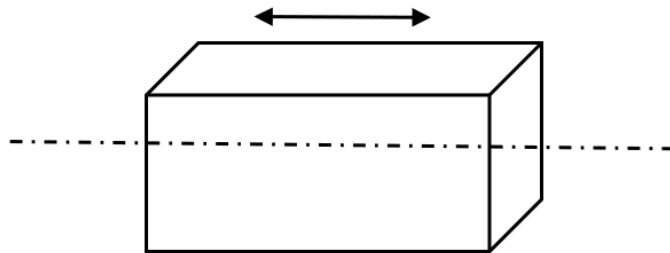


Figura 2.3: Articulación prismática

Fuente: Libro [5]

- **Espacio articular:** Se refiere al entorno donde se describe la posición de todos los componentes del robot. Es como el idioma que utiliza el mecanismo para realizar movimientos, ya sean rotacionales o de desplazamiento. El tamaño de este espacio, denotado como N , depende de cuántas formas de movimiento libre tiene la estructura. En una estructura abierta o en forma de árbol, las articulaciones pueden moverse de manera independiente, pero en una estructura cerrada es necesario establecer relaciones entre las diferentes articulaciones.
- **Espacio de trabajo:** En robótica, la definición más común del término espacio de trabajo viene dado por Yi Cao [7]. “El espacio de trabajo de un robot está definido como el grupo de puntos que pueden ser alcanzados por su efector-final”. El espacio de trabajo de los

robots convencionales ha sido estudiado para más de tres décadas y muchos métodos han sido propuesto. En el cálculo del espacio de trabajo del robot, lo que más preocupante es su correspondiente forma y tamaño. El curvas límite (para manipuladores robóticos 2D) y superficies (para manipuladores de robots 3D) del espacio de trabajo del robot han sido estudiado utilizando métodos gráficos, analíticos o numéricos.

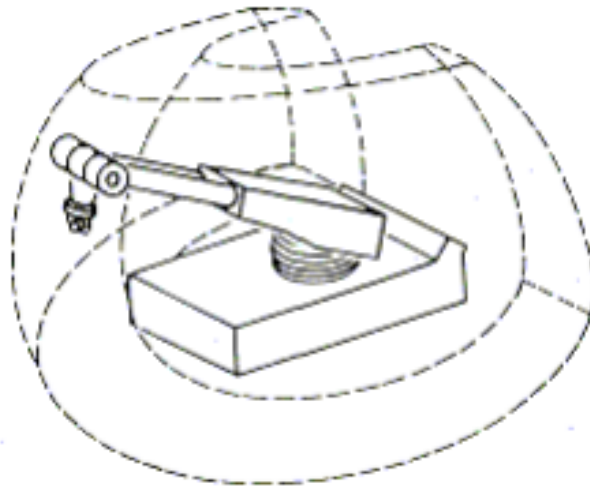


Figura 2.4: Espacio de Trabajo

Fuente: Pagina Web [6]

- **Arquitectura de robots manipuladores:** El estudio de las estructuras en forma de árbol y de cadenas cerradas se puede reducir a algunas cadenas abiertas equivalentes. Por lo tanto, la clasificación presentada a continuación es relevante para las arquitecturas de cadenas abiertas simples, pero también se puede generalizar a cadenas complejas. Para contar las posibles arquitecturas, solo consideramos articulaciones de revolución o prismáticas cuyos ejes consecutivos son paralelos o perpendiculares. Generalmente, con algunas excepciones, los ejes consecutivos de los robots utilizados actualmente son paralelos o perpendiculares. Las diferentes combinaciones de estos cuatro parámetros determinan el número de arquitecturas posibles en función del número de articulaciones, con 6 grados de libertad es posible construir hasta 3,508 robots completamente diferentes, Como lo menciona Wisama Khalil e Etienne Dombre [3], lo cual muestra la gran diversidad de aplicaciones para las cuales pueden ser construidos.

Grados de Libertad	No. de estructuras
2	8
3	36
4	168
5	776
6	3508

Cuadro 2.2: Estructuras posibles de robots dependiendo de los grados de libertad.

Fuente: Libro [3]

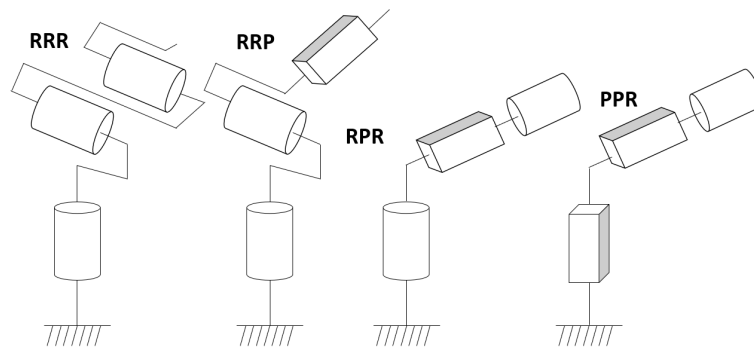


Figura 2.5: Tipos de arquitecturas de 3 Grados de libertad

Fuente: Libro [3]

- Grado de libertad:** se refiere a las diferentes formas en que el robot puede desplazarse o rotar en el espacio. Los movimientos independientes se refieren a los grados de libertad del robot. En un espacio tridimensional, para ubicar un objeto de manera precisa, se requieren tres grados de libertad correspondientes a cada dimensión: uno para el movimiento en el eje X, otro para el eje Y y uno más para el eje Z. Sin embargo, para que un robot pueda posicionar y orientar un objeto de manera completa en el espacio, se necesitan seis grados de libertad. Los tres grados de libertad mencionados anteriormente para el posicionamiento en el espacio y otros tres grados de libertad adicionales para lograr cualquier tipo de rotación deseada. Esto significa que un robot con menos de seis grados de libertad no podrá alcanzar cualquier punto del espacio de trabajo con una orientación arbitraria. Los grados de libertad adicionales permiten al robot ajustar su orientación en diferentes ángulos y ejes para adaptarse a las necesidades específicas de manipulación y colocación del objeto en el espacio tridimensional.

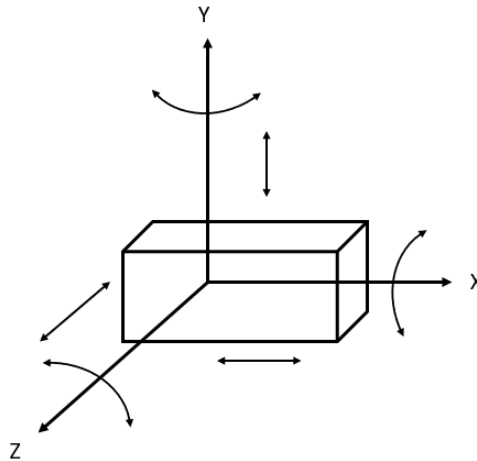


Figura 2.6: Grados de libertad de un eslabón en el espacio

Fuente: Libro [5]

2.2.4. Cinemática Directa

La cinemática del robot estudia el movimiento del mismo con respecto a un sistema de referencia dado sin considerar las fuerzas que intervienen. Así, la cinemática se interesa por la descripción analítica del movimiento espacial del robot como una función del tiempo, y en particular por las relaciones entre la posición y la orientación del extremo final del robot con los valores que toman sus coordenadas articulares [8]. Para abordar la cinemática del robot, se presentan dos conceptos fundamentales. El primero es conocido como cinemática directa, el cual consiste en determinar la posición y orientación del extremo final del robot en relación a un sistema de coordenadas de referencia. Esto se logra utilizando los valores de las articulaciones y los parámetros geométricos de los componentes del robot. La cinemática inversa, resuelve la configuración que el robot debe adoptar para lograr una posición y orientación específicas en su extremo final, ya conocidas. por lo cual se entiende la relación que se hay entre ellas como se muestra en la Figura (2.7) [10].

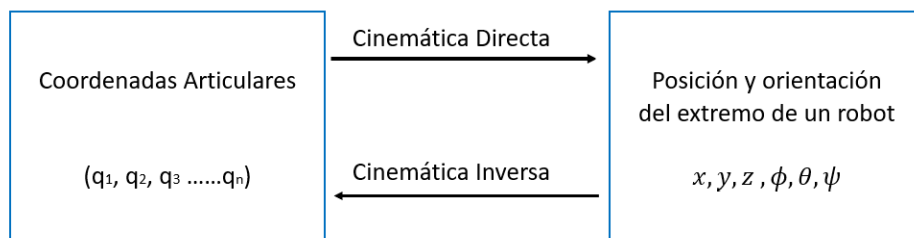


Figura 2.7: Diagrama de relación entre cinemática directa e inversa.

Fuente: Libro [13]

Parámetros Denavit y Hartenberg Denavit y Hartenberg propusieron un método sistemático para describir y representar la geometría espacial de los elementos en una cadena cinemática, como en el caso de un robot, en relación a un sistema de referencia fijo. [13] Este enfoque se basa en el uso de matrices de transformación homogéneas que permiten describir la relación espacial entre elementos adyacentes. Así, el problema cinemático directo se simplifica al encontrar una matriz de transformación homogénea 4x4 que relacione la posición y orientación del extremo del robot con el sistema de coordenadas en su base.

La resolución del problema cinemático directo nos brinda información precisa sobre la posición y orientación del extremo del robot cuando se establecen valores específicos para las variables que definen la configuración de las articulaciones.

Dado que las variables articulares pueden ser medidas directamente por sensores controlados por la unidad de control del robot, el modelo cinemático directo se utiliza para proporcionar información relevante al usuario acerca de la ubicación del extremo del robot, entre otros usos.

Para obtener los parámetros D-H se siguen los siguientes pasos:

- Asignar los ejes de acción en cada junta del robot.
- Asignar los orígenes del robot.
- Asignar los sistemas coordenados de cada origen.
- Llenar la tabla de los parámetros D-H.

En donde las reglas para obtener los parámetros existen reglas las siguientes

- a_i = Distancia a lo largo del eje x_i , de 0_i a la intersección de z_{i-1} y x_i .
- d_i = Distancia a lo largo de z_{i-1} de 0_{i-1} a la intersección de z_{i-1} y x_i . d_i es variable de juntas prismáticas.
- α_i = Ángulo entre z_{i-1} y z_i medido alrededor de x_i .
- θ_i = Ángulo entre x_i y x_{i-1} medido alrededor de z_{i-1} . Si la junta es rotacional θ_i = es variable.

Hay que hacer notar que, si bien en general la matriz de transformación homogénea queda definida por seis grados de libertad, el método de Denavit-Hartenberg, permite, en eslabones rígidos, reducir éste a 4 con la correcta elección de los sistemas de coordenadas. Estas 4 transformaciones básicas consisten en una sucesión de rotaciones y traslaciones que permiten relacionar

i	a_{i-1}	α_{i-1}	d_i	θ_i
1	0	0	0	θ_1
2	0	0	0	θ_2
3	0	0	0	θ_3

Cuadro 2.3: tabla de los parámetros D-H

Fuente: Tabla Propia

el sistema de referencia del elemento $i-1$ con el sistema del elemento i . Las transformaciones en cuestión son las siguientes:

- Rotación alrededor del eje z_{i-1} un ángulo θ_i .
- Traslación a lo largo de z_{i-1} una distancia d_i .
- Traslación a lo largo de x_i una distancia a_i .
- Rotación alrededor del eje x_i un ángulo α_i .

$${}^{i-1}A_i = Rot_z, \theta_i \ Trans_z, d_i \ Transl_x, a_i \ Rot_x, \alpha_i$$

$$\begin{vmatrix} C_{\theta_i} & -S_{\theta_i} & 0 & 0 \\ S_{\theta_i} & C_{\theta_i} & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{vmatrix} \begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{vmatrix} \begin{vmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix} \begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & C_{\alpha_i} & -S_{\alpha_i} & 0 \\ 0 & S_{\alpha_i} & C_{\alpha_i} & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

$$= \begin{vmatrix} C_{\theta_i} & -S_{\theta_i}C_{\alpha_i} & S_{\theta_i}S_{\alpha_i} & a_iC_{\theta_i} \\ S_{\theta_i} & C_{\theta_i}C_{\alpha_i} & -C_{\theta_i}S_{\alpha_i} & a_iS_{\theta_i} \\ 0 & S_{\alpha_i} & C_{\alpha_i} & d_i \\ 0 & 0 & 0 & 1 \end{vmatrix} \quad (2.1)$$

Transformaciones homogéneas: La representación y descripción de la posición de un objeto en el espacio tridimensional con respecto a un eje de referencia fijo se logra exclusivamente utilizando álgebra vectorial y matricial. En el caso de un robot, que puede considerarse como una cadena cinemática compuesta por objetos rígidos o eslabones unidos por articulaciones, se puede establecer un sistema de referencia fijo en la base del robot para describir la ubicación de cada eslabón con respecto a este sistema de referencia. Por lo tanto, el problema de la cinemática directa consiste en encontrar una matriz de transformación homogénea, denotada como T , que relaciona la posición y orientación del extremo del robot con respecto al sistema de referencia fijo ubicado en su base. Esta matriz T es una función de las coordenadas articulares del robot.

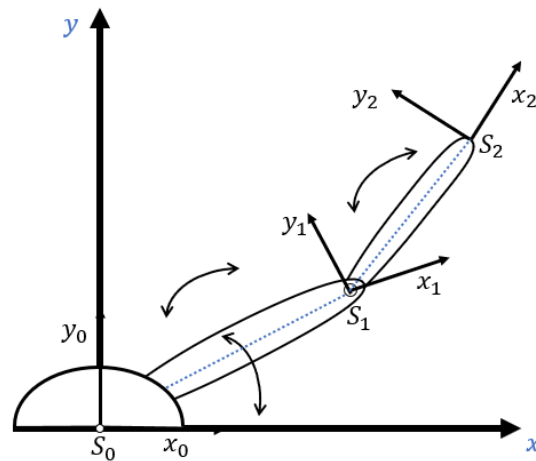


Figura 2.8: Asociación de Sistemas de referencia

Fuente Propia

En general, un robot de n grados de libertad está formado por n eslabones unidos por $n - 1$ articulaciones, de forma que cada par articulación-eslabón constituye un grado de libertad, a cada eslabón se le puede asociar un sistema de referencia de la Figura 2.8 solidario a él y, utilizando las transformaciones homogéneas, es posible representar las rotaciones y traslaciones relativas entre los distintos eslabones que componen el robot. Normalmente, la matriz de transformación homogénea que representa la posición y orientación relativa entre los sistemas asociados a dos eslabones consecutivos del robot se suele denominar matriz ${}^{i-1}A_i$. Así pues 0A_1 describe la posición y orientación del sistema de referencia solidario al primer eslabón con respecto al sistema de referencia solidario a la base, 1A_2 describe la posición y orientación del segundo eslabón respecto del primero, etc. Del mismo modo, denominando 0A_k a las matrices resultantes del producto de las matrices ${}^{i-1}A_i$ con i desde 1 hasta k , se puede representar de

forma total o parcial la cadena cinemática que forma el robot. Así, por ejemplo, la posición y orientación del sistema solidario con el segundo eslabón del robot con respecto al sistema de coordenadas de la base se puede expresar mediante la matriz 0A_2 .

$${}^0A_2 = {}^0A_1 {}^1A_2$$

Cuando se consideran todos los grados de libertad, a la matriz 0A_n se le suele denominar T . Así, dado un robot de seis grados de libertad, se tiene que la posición y orientación del eslabón final vendrá dada por la matriz T :

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

Cada una de las matrices iA_i representa el cambio de base que permite pasar del sistema asociado al eslabón $i - 1$ al asociado eslabón i . Esta matriz dependerá, además de constantes geométricas propias del eslabón, del grado de libertad q_i . Por tanto, la expresión anterior, se puede escribir como:

$$T(q_1 \dots q_n) = {}^0A_1(q_1) {}^1A_2(q_2) \dots {}^{n-1}A_n(q_n)$$

Como resultado, la relación entre los sistemas de coordenadas de la base y del extremo del robot se establece mediante una matriz de transformación homogénea T , que es una función de las coordenadas articulares.

Esta matriz T debe ser igual a la matriz de transformación homogénea correspondiente a la ubicación deseada para posicionar al robot.

Ejemplo: Robot de 2 grados del libertad que se muestra en la Figura 2.8. El paso de S_0 a S_1 se obtiene mediante un giro q_1 entorno al eje z y una traslación l_1 a lo largo del nuevo eje x , Tal que:

$$\begin{aligned} {}^0A_1 &= \text{Rot}_z(q_1) T(l_1, 0, 0) \\ &= \begin{vmatrix} C\theta_1 & -S\theta_1 & 0 & 0 \\ S\theta_1 & C\theta_1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{vmatrix} \begin{vmatrix} 1 & 0 & 0 & l_1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix} \\ &= \begin{vmatrix} C\theta_1 & -S\theta_1 & 0 & l_1 C\theta_1 \\ S\theta_1 & C\theta_1 & 0 & l_1 S\theta_1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{vmatrix} \end{aligned} \tag{2.2}$$

El paso de S_1 a S_2 se consigue mediante un giro de q_2 entorno al eje z y un desplazamiento de l_2 a lo largo del nuevo eje x , tal que:

$$\begin{aligned}
{}^0A_1 &= \text{Rot}z(q_2) T(l_2, 0, 0) \\
&= \begin{vmatrix} C\theta_2 & -S\theta_2 & 0 & 0 \\ S\theta_2 & C\theta_2 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{vmatrix} \begin{vmatrix} 1 & 0 & 0 & l_2 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix} \\
&= \begin{vmatrix} C\theta_2 & -S\theta_2 & 0 & l_2C\theta_2 \\ S\theta_2 & C\theta_2 & 0 & l_2S\theta_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix} \tag{2.3}
\end{aligned}$$

Se obtiene entonces:

$$\begin{aligned}
T &= {}^0A_1 {}^0A_2 \\
&= \begin{vmatrix} C\theta_1S\theta_2 - S\theta_1S\theta_2 & -C\theta_1S\theta_2 - S\theta_1C\theta_2 & 0 & l_1C\theta_1 + l_2C\theta_{12} \\ S\theta_1C\theta_2 + C\theta_1S\theta_2 & -S\theta_1S\theta_2 + C\theta_1C\theta_2 & 0 & l_1S\theta_1 + l_2S\theta_{12} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix} \tag{2.4}
\end{aligned}$$

Donde,

$$\begin{vmatrix} n_x & o_x & a_x \\ n_y & o_y & a_y \\ n_z & o_z & a_z \end{vmatrix} \tag{2.5}$$

Presenta la orientación mientras que

$$\begin{vmatrix} n_x \\ n_y \\ n_z \end{vmatrix} \tag{2.6}$$

Presenta la posición.

2.2.5. Python

Python es un lenguaje de programación de alto nivel, interpretado y de propósito general. Fue creado a finales de la década de 1980 por Guido van Rossum y ha experimentado un crecimiento exponencial en popularidad desde entonces. Es un software libre, o sea que puede ser utilizado gratuitamente gracias al trabajo de Python Foundation, y de innumerables colaboradores, puede utilizar Python prácticamente en cualquier arquitectura de computadoras o sistemas operativos como Linux, FreeBSD, Microsoft Windows o Mac Os X.

Una de las principales razones de la popularidad de Python es su enfoque en la productividad y la facilidad de uso. Su sintaxis intuitiva permite a los programadores escribir código más rápido y con menos esfuerzo en comparación con otros lenguajes. Además, Python cuenta con una amplia comunidad de desarrolladores que han creado una gran cantidad de bibliotecas y módulos, lo que hace que sea fácil encontrar soluciones preexistentes para una amplia variedad de problemas.

Las variables se utilizan para dar nombres a los valores (por ejemplo, Celsius). Técnicamente, todos estos nombres se denominan identificadores. Python tiene algunas reglas sobre cómo se forman los identificadores. Todo identificador debe comenzar con una letra o guion bajo que puede ir seguido de una secuencia de letras, dígitos o guiones bajos. Esto implica que un solo identificador no puede contener espacios.

Otra cosa importante a tener en cuenta es que algunos identificadores son parte del mismo Python. Estos nombres se denominan palabras reservadas y no se pueden utilizar como identificadores ordinarios. La lista completa de palabras reservadas de Python se muestra en la figura.

2.4

and	del	for	is	raise
assert	elif	from	lambda	return
break	else	global	not	try
class	except	if	or	while
continue	exec	import	pass	yield
def	finally	in	print	

Cuadro 2.4: Palabras reservadas de Python

Fuente: Pagina Web [15]

Librerías Python

Python tiene una amplia capacidad para utilizar bibliotecas y módulos que amplían su funcionalidad y permiten realizar tareas específicas de manera más eficiente. Estas bibliotecas son conjuntos de código predefinido que contienen funciones, clases y métodos diseñados para abordar problemas específicos.

En el ámbito de la cinemática, hay varias bibliotecas populares que pueden ser útiles. A continuación, se presentan dos ejemplos:

- **NumPy:** NumPy es una biblioteca fundamental para la computación científica en Python. Proporciona una potente estructura de datos llamada “array” y un amplio conjunto de funciones para realizar operaciones numéricas eficientes. NumPy es especialmente útil para realizar cálculos con vectores y matrices, lo cual es relevante en problemas de cinemática. Permite realizar operaciones matemáticas rápidas y eficientes, como multiplicación de matrices, cálculos estadísticos y manipulación de datos numéricos. Además, ofrece herramientas para generar y manipular secuencias de números y crear gráficos.
- **SciPy:** SciPy es una biblioteca que se basa en NumPy y proporciona funcionalidades adicionales para la ciencia y la ingeniería. Incluye módulos para la interpolación, optimización, álgebra lineal, procesamiento de señales, transformadas de Fourier, estadísticas y más. En el contexto de la cinemática, SciPy puede ser útil para tareas como ajuste de curvas, resolución numérica de ecuaciones y análisis de datos.



Figura 2.9: Logo de Numpy.org

Fuente: Pagina Web [16]

Matrices de rotación

Las matrices homogéneas de rotación son empleadas para rotar una figura en un sistema de origen XYZ en torno a uno de sus ejes, de modo que hay tres matrices distintas que representan la rotación para los ejes x, y y z.

Matrices de rotación en Python.

```
import numpy as np
from sympy import *

def degToRad(angulo):
    anguloRad = angulo * (np.pi/180)
    return anguloRad

def matrizHomogeneaRotacionX(angulo):
    ARad = degToRad(angulo)
    rotacionX= np.array([[1, 0, 0, 0],
                        [0,np.cos(ARad), -np.sin(ARad),0],
                        [0,np.sin(ARad), np.cos(ARad), 0],
                        [0, 0, 0, 1]])

    return rotacionX

def matrizHomogeneaRotacionY(angulo):
    ARad = degToRad(angulo)
    rotacionY= np.array([[np.cos(ARad), 0, np.sin(ARad), 0],
                        [0, 1, 0, 0],
                        [-np.sin(ARad),0, np.cos(ARad), 0],
                        [0, 0, 0, 1]])

    return rotacionY

def matrizHomogeneaRotacionZ(angulo):
    ARad = degToRad(angulo)
    rotacionZ= np.array([[np.cos(ARad), -np.sin(ARad), 0, 0],
                        [-np.sin(ARad), np.cos(ARad), 0, 0],
                        [0, 0, 1, 0],
                        [0, 0, 0, 1]])

    return rotacionZ
```


Matrices de traslación

La matriz de transformación homogénea de traslación es una herramienta fundamental en la geometría computacional y la robótica, ya que permite describir y realizar movimientos y desplazamientos en el espacio tridimensional de manera precisa y eficiente. Se utiliza para calcular el desplazamiento del sistema actual XYZ al sistema destino XYZ. Esta transformación se define mediante una matriz que se muestra a continuación.

Matriz de traslación en Python.

```
import numpy as np
from sympy import *

def matrizHomogeneaTraslacion(px,py ,pz):
    translacion= np.array([[1, 0, 0, px],
                           [0, 1, 0, py],
                           [0, 0, 1, pz],
                           [0, 0, 0, 1]])

    return translacion

px=100
py=100
pz=0
translacion =matrizHomogeneaTraslacion(px,py,pz)
```

2.2.6. Jupyter-Lab

JupyterLab es una interfaz de usuario web desarrollada por Project Jupyter [9], que proporciona un entorno interactivo para el desarrollo de código, análisis de datos y creación de documentos colaborativos. Es una evolución de Jupyter Notebook, que amplía sus capacidades y proporciona una interfaz más flexible y poderosa, cuenta con algunas características y usos de JupyterLab:

- **Entorno de trabajo flexible:** JupyterLab ofrece una interfaz basada en pestañas que permite organizar y trabajar con diferentes elementos en un mismo entorno. Puedes abrir notebooks, editores de código, terminales, visores de datos, archivos y otros componentes en pestañas separadas y moverlas según tus necesidades.
- **Notebooks interactivos:** Al igual que en Jupyter Notebook, JupyterLab permite crear y ejecutar notebooks interactivos. Los notebooks son documentos en los que puedes combinar código, texto explicativo, imágenes, gráficos y otros elementos enriquecidos. El código se ejecuta en celdas, lo que permite una ejecución interactiva y la visualización de los resultados directamente en el mismo documento.
- **Soporte para múltiples lenguajes de programación:** JupyterLab es compatible con una amplia variedad de lenguajes de programación, incluyendo Python, R, Julia, Scala y muchos otros. Esto significa que puedes escribir y ejecutar código en diferentes lenguajes dentro de los notebooks y otros editores de código disponibles en JupyterLab.
- **Exploración de datos:** JupyterLab proporciona herramientas y extensiones para el análisis y exploración de datos. Puedes cargar conjuntos de datos, visualizarlos con gráficos interactivos, realizar transformaciones y manipulaciones, y realizar análisis estadísticos avanzados, todo dentro del entorno de JupyterLab.

La instalación de jupyter es gratuita y fácil de utilizar luego de unos sencillos pasos.

- **Paso1:Instalación de Jupyter** Una vez que tengas Python instalado, puedes instalar Jupyter utilizando pip, el administrador de paquetes de Python. Abre una terminal o línea de comandos y ejecuta el siguiente comando: **pip install jupyter**. Esto descargará e instalará la paquetería Jupyter y sus dependencias.
- **Paso 2: Ejecución de Jupyter Notebook** Una vez instalado Jupyter, puedes ejecutar Jupyter Notebook abriendo una terminal o línea de comandos y ejecutando el siguiente comando: **jupyter notebook**.Esto iniciará el servidor de Jupyter y abrirá una nueva pestaña o ventana en tu navegador web predeterminado.
- **Paso 3: Creación de un nuevo notebook** Dentro de la interfaz de Jupyter Notebook en tu navegador, puedes hacer clic en el botón "New" seleccionar "Python 3" para crear un nuevo notebook de Python. Se abrirá una nueva pestaña con el notebook vacío, listo para que empieces a trabajar.

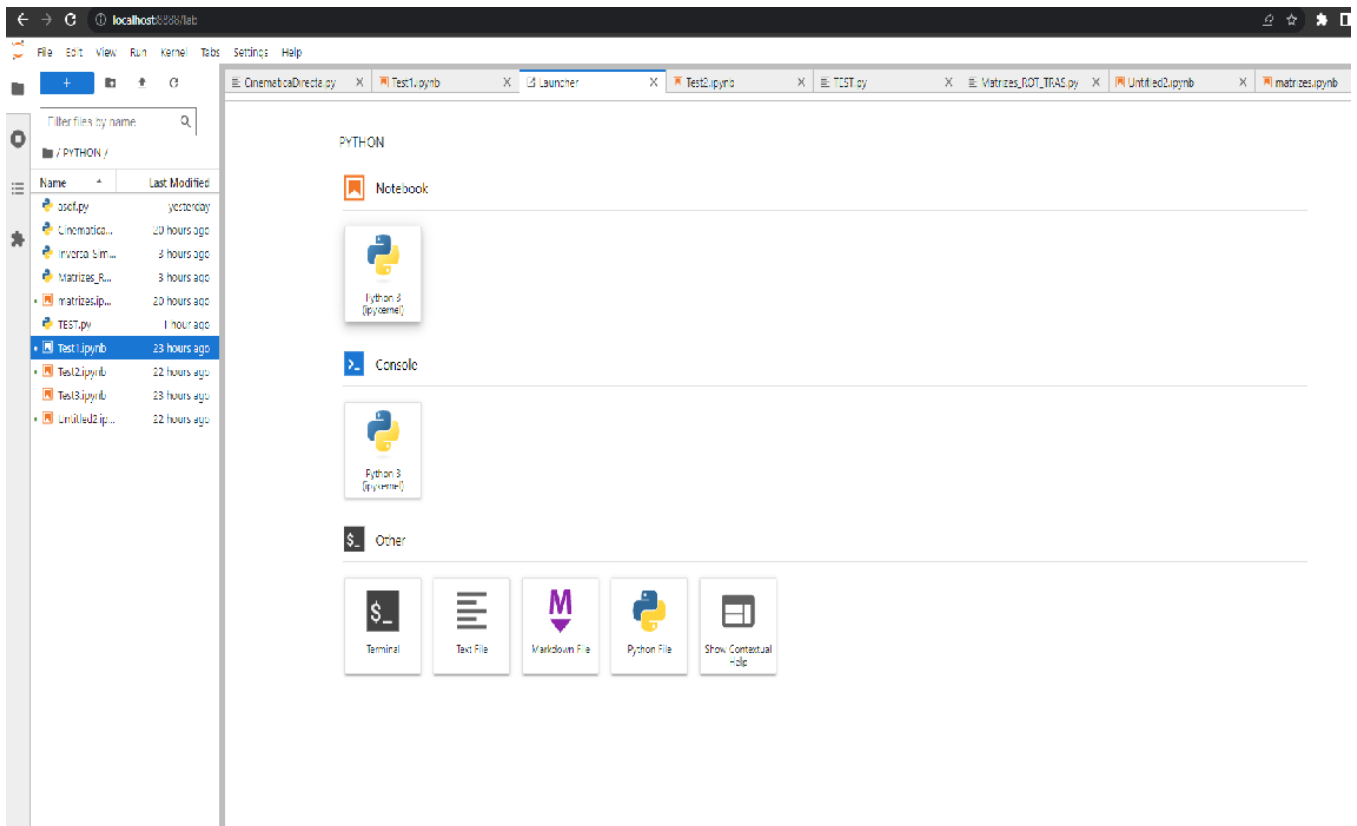


Figura 2.10: Entorno Jupyter

Fuente: PaginaWeb [9]

2.2.7. Cinemática Inversa

El análisis de la cinemática inversa de un robot consiste en hallar los valores de las coordenadas articulares del robot $q = [q_0, q_1, q_2, \dots, q_n]^T$ conocida la posición y orientación del extremo del robot. , todas las posiciones y orientaciones iniciales.

El método geométrico: Consiste en descomponer el problema cinemático inverso del robot estudiado en varios problemas de geometría plana, y resolver dichos problemas empleando relaciones trigonométricas [10]. Por su propia definición, es obvio que este método depende muy fuertemente de la geometría de la estructura del robot, y cualquier secuencia de pasos empleada para obtener el IKM de un sistema robotizado no será válida para otra estructura diferente.

El método analítico: Para el cálculo del IKM consiste en despejar el vector de estado de los actuadores del robot del sistema de ecuaciones que define su cinemática directa. El inconveniente de esta metodología radica en que el mencionado sistema de ecuaciones normalmente este compuesto por ecuaciones no lineales, por lo que dicho despeje no se puede efectuar empleando procedimientos tradicionales de álgebra matricial, sino que se deben buscar, dentro del sistema de ecuaciones estudiado, relaciones específicas para cada caso. Esto significa que el método analítico tampoco es un procedimiento sistemático, porque los pasos seguidos para calcular el IKM de un robot con una determinada estructura, no necesariamente sean válidos o aplicables para otro robot diferente.

El desacoplo cinemático: Es un enfoque utilizado en robots para separar los problemas de posición y orientación en su funcionamiento. Este método es aplicable a robots cuyos últimos tres grados de libertad se cortan en un punto, lo que se conoce como la "muñeca" del robot.

El desacoplo cinemático aprovecha el hecho de que los tres últimos grados de libertad se cortan en un punto para separar los problemas de obtención del modelo cinemático inverso de posición y orientación. Para lograr esto, se establecen las coordenadas del punto de corte de los tres últimos ejes y se calculan los valores de las primeras tres variables articulares para lograr esa posición. Luego, utilizando los datos de orientación deseados para el extremo del robot y los valores calculados anteriormente, se obtienen los valores de las variables articulares restantes.

Ejemplo: Un péndulo de un grado de libertad, obtendremos la cinemática inversa por medio del método geométrico. El péndulo se representa geoméricamente en la siguiente Figura 2.11.

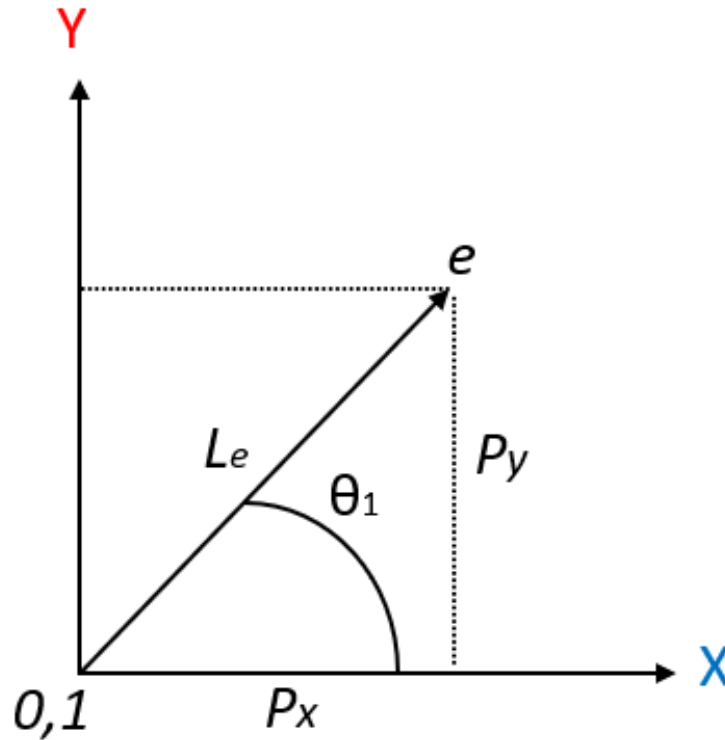


Figura 2.11: Representación geométrica del péndulo de 1 gdl

Fuente: Libro [17]

Teniendo en cuenta la relación $\tan(\theta)$ podemos determinar el valor de θ por medio de un despeje muy simple el cual nos da como resultado el valor que buscamos el cual es:

$$= \text{atan2}(Py, Px) \quad (2.7)$$

Siendo *atan2* un comando de programación el cual devuelve la tangente inversa de las coordenadas especificadas. Analizando el mismo ejemplo, pero ahora usando el método analítico, teniendo la tabla de parámetros y la matriz de transformación del sistema completo [17].

i	a_{i-1}	α_{i-1}	d_i	θ_i
1	0	0	0	θ_1
e	L_e	0	0	0

(2.8)

Cuadro 2.5: Tabla parámetros péndulo invertido

Fuente: Libro [17]

Se deduce la matriz deseada, se observa que en el péndulo puede tener una orientación “ ϕ ”, una distancia x y una distancia y respecto a la base.

$$\begin{vmatrix} \cos(\phi) & -\sin(\phi) & 0 & P_x \\ \sin(\phi) & \cos(\phi) & 0 & P_x \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix} \quad (2.9)$$

Posteriormente se iguala con la matriz T_e^0

$$T_e^0 = \begin{vmatrix} \cos(\phi) & -\sin(\phi) & 0 & P_x \\ \sin(\phi) & \cos(\phi) & 0 & P_x \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix} = \begin{vmatrix} \cos(\theta_i) & -\sin(\theta_i) & 0 & L_e \cos(\theta_i) \\ \sin(\theta_i) & \cos(\theta_i) & 0 & L_e \sin(\theta_i) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix} = T_d \quad (2.10)$$

$$\cos(\theta_1)L_e = P_x \longrightarrow \cos(\theta_1) = \frac{P_x}{L_e} \quad (2.11)$$

$$\sin(\theta_1)L_e = P_x \longrightarrow \sin(\theta_1) = \frac{P_x}{L_e}$$

De uno de los teoremas que existen para la simplificación tenemos la siguiente ecuación donde $\sin\theta = a$ y $\cos\theta = b$, tal que,

la solución es:

$$\theta = \text{atan2}(a, b) \quad (2.13)$$

$$\theta = \text{atan2}(Py, Px) \quad (2.14)$$

2.2.8. Ángulos de Euler

Los ángulos de Euler son 3 rotaciones simples ψ, θ, Φ y sucesivas alrededor de 3 ejes, cada uno de los cuales es orientado por las rotaciones anteriores. Al ser rotaciones simples, en las máquinas a menudo quedan materializadas por pares cinemáticos cilíndricos o de revolución. En la fig 2.12 se muestran los dos juegos de ángulos de Euler: a) ángulos de Euler empleados tradicionalmente en la orientación de rotores rápidos, por ejemplo giroscopios, y b) ángulos de Euler de tres ejes empleados en la orientación. A partir de las matrices de cambio elementales asociadas a cada una de las rotaciones introducidas por los ángulos de Euler $[\phi], [\theta]$ y $[\Phi]$ la matriz de cambio total es $[S] = [\phi], [\theta]$ y $[\Phi]$ [14].

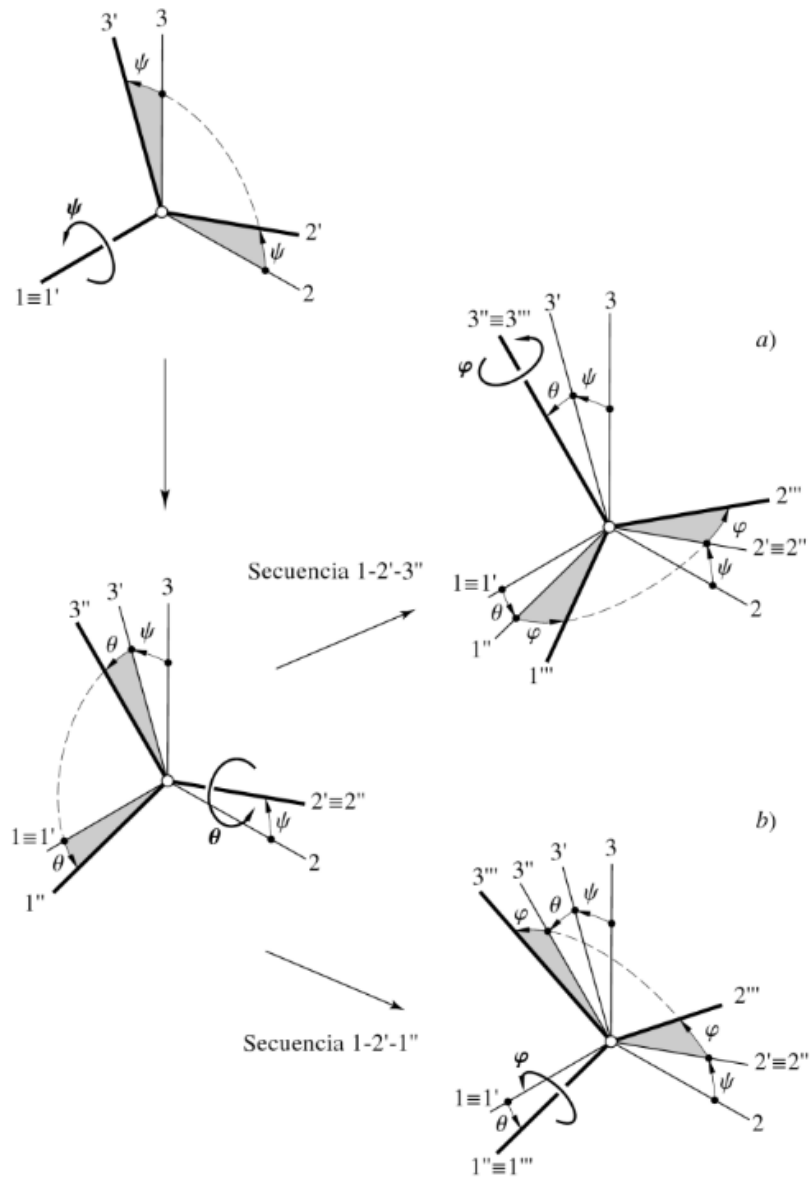


Figura 2.12: Ángulos de Euler a) de dos ejes y b) de tres ejes

Fuente: Libro [14]

Capítulo 3

PROCEDIMIENTO Y DESCRIPCIÓN DE LAS ACTIVIDADES REALIZADAS

3.1. Descripción de la metodología

3.1.1. Curso de programación del robot industrial marca KUKA

Recibí un curso del robot industrial, se aprendieron conocimientos básicos del robot manipulador, la unidad de programación KUKA SmartPad Figura 3.1, la interfaz entre la unidad de control KR C4 Compact Figura 3.2 y la programación mediante el lenguaje KRL - KUKA Robot Language.

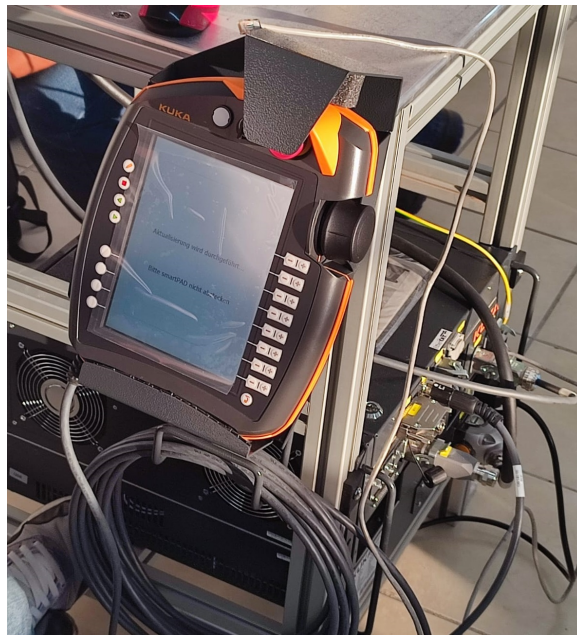


Figura 3.1: SmartPad

Fuente: Propia



Figura 3.2: Unidad de Control KR C4 Compact

Fuente: Propia

3.2. Estudio de cinemática directa

La cinemática directa consiste en identificar la orientación del efector final del manipulador y posición con ángulo de articulación conocido (θ_i). El KUKA KR6 R900 tiene seis ángulos de articulación ($\theta_1, \theta_2, \theta_3, \theta_4, \theta_5$ y θ_6) como se ilustra en la Figura 3.3.

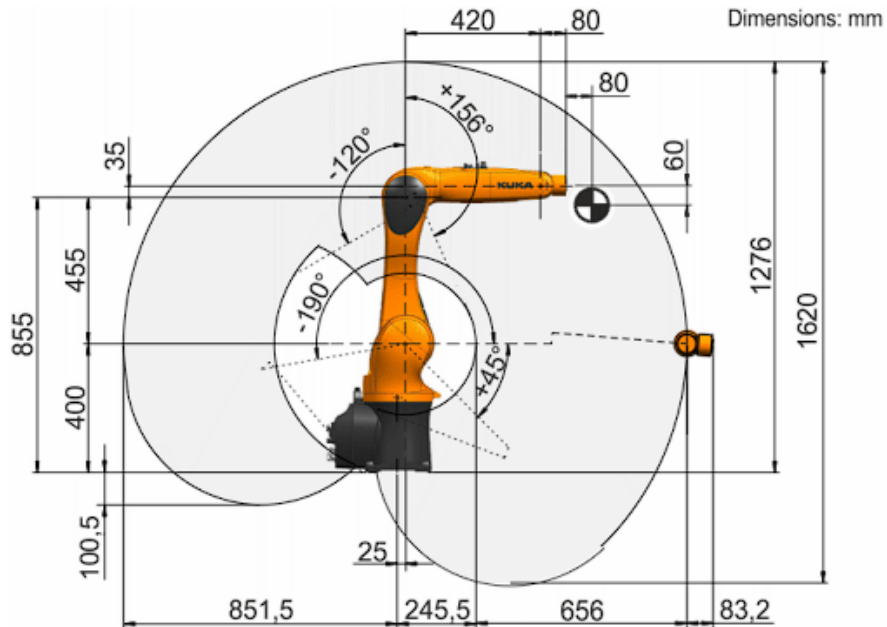


Figura 3.3: Gráfico del espacio de trabajo

Fuente: Hoja de datos del robot KR6 R900 Sixx

Posteriormente se establecen su sistema cinemático, Figura 3.4.

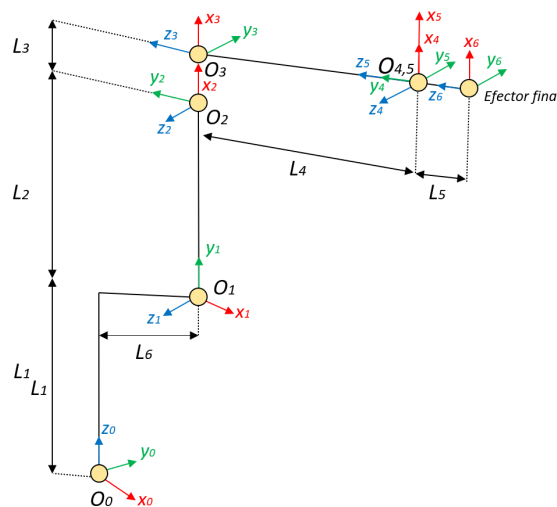


Figura 3.4: Sistema cinemático de robot manipulador KUKA R900 Sixx

Fuente: Propia

Ahora obtener la tabla de parámetros Denavit-Hartenberg

i	a_{i-1}	α_{i-1}	d_i	θ_i
1	$L6$	-90°	$L1$	θ_1
2	$L2$	0	0	$\theta_2 - 90$
3	$L3$	90°	0	θ_3
4	0	-90°	$-L4$	θ_4
5	0	-90°	0	θ_5
6	0	0	$L5$	θ_6

Cuadro 3.1: Tabla de los parámetros D-H del robot manipulador KR6 R900 Sixx

Fuente: Propia

Una vez obtenida la tabla de parámetros Denavit-Hartenberg se formula el código en “Visual Studio Code” , se comienza por importar las librerías sympy y numpy , posteriormente por definir las variables que se usarían para nuestro parámetros D-H , (A_{i-1} , α_{i-1} , d_i , θ_i).

Matriz de transformación homogénea código Python

```

from sympy import *
import numpy as np

n = 6 # Grados de libertad
# As
a1, a2, a3 = symbols("L6,L2,L3")
a = [a1, a2, a3, 0, 0, 0]
# Alphas
alpha = [-np.pi/2, 0,np.pi/2, -np.pi/2,-np.pi/2, 0]
# d's
d1, d4, d5 = symbols("L1,L4,L5")
d = [d1, 0,0,d4, 0,d5]
# Thetas
q1, q2, q3 = symbols("theta_1,theta_2-90,theta_3")
q4, q5, q6 = symbols("theta_4,theta_5,theta_6")
theta = [q1, q2, q3, q4, q5, q6]
Ts = []

```

```

def MTH(theta, alpha, d, a):
T = Matrix([[cos(theta), -sin(theta), 0, a],
            [sin(theta)*round(np.cos(alpha)), cos(theta)*round(np.cos(alpha))
            , round(np.sin(alpha)), -d*round(np.sin(alpha))],
            [sin(theta)*round(np.sin(alpha)), cos(theta)*round(np.sin(alpha)),
            round(np.cos(alpha)), d*round(np.cos(alpha))],
            [0, 0, 0, 1]])

    return T

for i in range(n):
    Ts.append(MTH(theta[i], alpha[i], d[i], a[i]))

Ts
T01 = Ts[0]
T12 = Ts[1]
T23 = Ts[2]
T34 = Ts[3]
T45 = Ts[4]
T56 = Ts[5]
T06 = simplify(T01 * T12* T23 * T34 * T45 *T56)
print(T06)

```

Entramos al entorno Jupyter el cual permite una visualización de las ecuaciones como se muestra en la Figura 3.5.

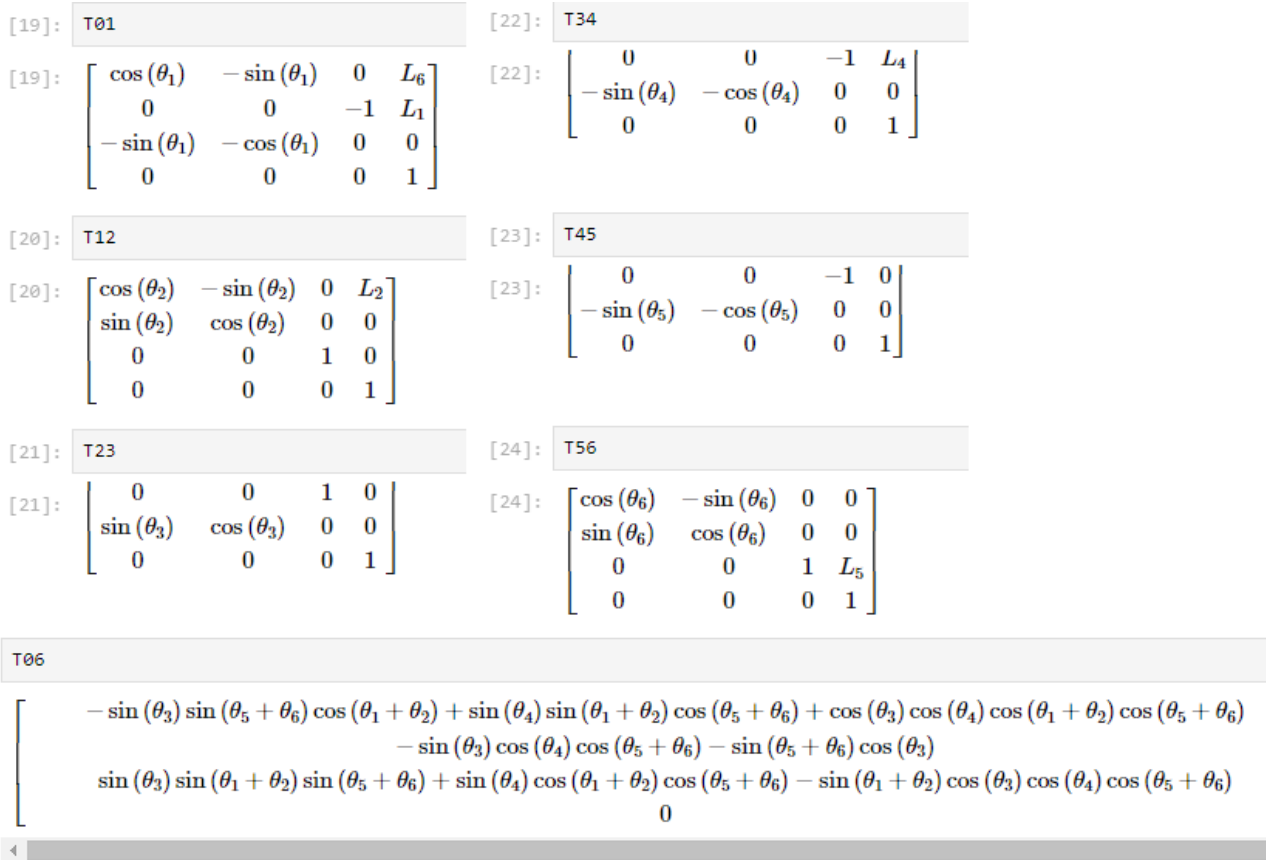


Figura 3.5: Matrices de Transformación en el entorno Jupyter

Fuente: Propia

De manera practica se simplifica la escritura de las ecuaciones c representa coseno y S también representa seno.

Ejemplo:

$$C_{1,2,3} = \cos(\theta_1)\cos(\theta_2)\cos(\theta_3) \text{ o } S_{1,2,3} = \sin(\theta_1)\sin(\theta_2)\sin(\theta_3)$$

De igual manera escribí un código en Matlab para comprobar los resultados de la cinemática directa, el cual se muestra en el Listing 3.1.

Listing 3.1: Código Matlab Comprobación

```
1 clear; % Limpiamos el Workspace
2 clc; % Limpiamos el Comand Window
3 close all; %Cerramos todo
4 syms theta1 L1 L2 L3 L4 L5 L6 Q P W F E D
5
6 P1=0
7 P2=0
8 P3=0
9 P4=0
10 P5=0
11 P6=0
12 %syms di theta_i ai alfa_i
13 theta_i= 0
14 ai= 25
15 alfa_i= -90
16 di= 400
17
18 %Matriz sistema uno-----
19 % Primera Matriz de Rotacion Angular Theta en Z
20 T1=[cosd(theta_i) -sind(theta_i) 0 0;
21     sind(theta_i) cosd(theta_i) 0 0;
22     0 0 1 0;
23     0 0 0 1];
24 % Segunda Translacion en z, Di
25 T2=[1 0 0 0;
26     0 1 0 0;
27     0 0 1 di;
28     0 0 0 1];
29
30 % Tercera Matriz de Translacion en X ai
```

```

31 T3=[1 0 0 ai;
32     0 1 0 0;
33     0 0 1 0;
34     0 0 0 1];
35
36 % Cuarta Matriz de rotacion angular en X, alfa_i
37 T4=[1 0 0 0;
38     0 cosd(alfa_i) -sind(alfa_i) 0;
39     0 sind(alfa_i) cosd(alfa_i) 0;
40     0 0 0 1 ];
41
42 T_T= T1*T2*T3*T4
43
44 P1= T_T
45 %Matriz sistema dos-----
46 theta_i= -90
47 ai= 455
48 alfa_i= 0
49 di= 0
50
51 % Primera Matriz de Rotacion Angular Teta en Z
52 T1=[cosd(theta_i) -sind(theta_i) 0 0;
53     sind(theta_i) cosd(theta_i) 0 0;
54     0 0 1 0;
55     0 0 0 1];
56
57 % Segunda Translacion en z, Di
58 T2=[1 0 0 0;
59     0 1 0 0;
60     0 0 1 di;
61     0 0 0 1];
62
63 % Tercera Matriz de Translacion en X ai

```

```

64 T3=[1 0 0 ai;
65     0 1 0 0;
66     0 0 1 0;
67     0 0 0 1];
68
69 % Cuarta Matriz de rotacion angular en X, alfa_i
70 T4=[1 0 0 0;
71     0 cosd(alfa_i) -sind(alfa_i) 0;
72     0 sind(alfa_i) cosd(alfa_i) 0;
73     0 0 0 1 ];
74
75 T_T= T1*T2*T3*T4
76
77 P2=T_T
78
79 %Matriz sistema tres-----
80 theta_i= 0
81 ai= 35
82 alfa_i= 90
83 di= 0
84
85 % Primera Matriz de Rotacion Angular Teta en Z
86 T1=[cosd(theta_i) -sind(theta_i) 0 0;
87     sind(theta_i) cosd(theta_i) 0 0;
88     0 0 1 0;
89     0 0 0 1];
90
91 % Segunda Translacion en z, Di
92 T2=[1 0 0 0;
93     0 1 0 0;
94     0 0 1 di;
95     0 0 0 1];
96

```



```

97 % Tercera Matriz de Translacion en X ai
98 T3=[1 0 0 ai;
99     0 1 0 0;
100    0 0 1 0;
101    0 0 0 1];
102
103 % Cuarta Matriz de rotacion angular en X, alfa_i
104 T4=[1 0 0 0;
105     0 cosd(alfa_i) -sind(alfa_i) 0;
106     0 sind(alfa_i) cosd(alfa_i) 0;
107     0 0 0 1 ];
108
109 T_T= T1*T2*T3*T4
110
111 P3=T_T
112 %Matriz sistema cuatro-----
113 theta_i= 0
114 ai= 0
115 alfa_i= -90
116 di= -500
117
118 % Primera Matriz de Rotacion Angular Teta en Z
119 T1=[cosd(theta_i) -sind(theta_i) 0 0;
120     sind(theta_i) cosd(theta_i) 0 0;
121     0 0 1 0;
122     0 0 0 1];
123
124 % Segunda Translacion en z, Di
125 T2=[1 0 0 0;
126     0 1 0 0;
127     0 0 1 di;
128     0 0 0 1];
129

```

```

130 % Tercera Matriz de Translacion en X ai
131 T3=[1 0 0 ai;
132     0 1 0 0;
133     0 0 1 0;
134     0 0 0 1];
135
136 % Cuarta Matriz de rotacion angular en X, alfa_i
137 T4=[1 0 0 0;
138     0 cosd(alfa_i) -sind(alfa_i) 0;
139     0 sind(alfa_i) cosd(alfa_i) 0;
140     0 0 0 1 ];
141
142 T_T= T1*T2*T3*T4
143
144 P4=T_T
145
146 %Matriz sistema cinco-----
147 theta_i= 0
148 ai= 0
149 alfa_i= -90
150 di= 0
151
152 % Primera Matriz de Rotacion Angular Teta en Z
153 T1=[cosd(theta_i) -sind(theta_i) 0 0;
154     sind(theta_i) cosd(theta_i) 0 0;
155     0 0 1 0;
156     0 0 0 1];
157
158 % Segunda Translacion en z, Di
159 T2=[1 0 0 0;
160     0 1 0 0;
161     0 0 1 di;
162     0 0 0 1];

```

```

163
164 % Tercera Matriz de Translacion en X ai
165 T3=[1 0 0 ai;
166     0 1 0 0;
167     0 0 1 0;
168     0 0 0 1];
169
170 % Cuarta Matriz de rotacion angular en X, alfa_i
171 T4=[1 0 0 0;
172     0 cosd(alfa_i) -sind(alfa_i) 0;
173     0 sind(alfa_i) cosd(alfa_i) 0;
174     0 0 0 1 ];
175
176 T_T= T1*T2*T3*T4
177
178 P5=T_T
179
180 %Matriz sistema seis -----
181 theta_i= 0
182 ai= 0
183 alfa_i= 0
184 di= 80
185
186 % Primera Matriz de Rotacion Angular Teta en Z
187 T1=[cosd(theta_i) -sind(theta_i) 0 0;
188     sind(theta_i) cosd(theta_i) 0 0;
189     0 0 1 0;
190     0 0 0 1];
191
192 % Segunda Translacion en z, Di
193 T2=[1 0 0 0;
194     0 1 0 0;
195     0 0 1 di;

```

```

196     0 0 0 1];
197
198 % Tercera Matriz de Translacion en X ai
199 T3=[1 0 0 ai;
200     0 1 0 0;
201     0 0 1 0;
202     0 0 0 1];
203
204 % Cuarta Matriz de rotacion angular en X, alfa_i
205 T4=[1 0 0 0;
206     0 cosd(alfa_i) -sind(alfa_i) 0;
207     0 sind(alfa_i) cosd(alfa_i) 0;
208     0 0 0 1 ];
209
210 T_T= T1*T2*T3*T4
211
212 P6= T_T
213 %Multiplicamos todas las matrices
214 TrasHomFin = P1*P2*P3*P4*P5*P6

```

Resultado del código se obtiene la siguiente matriz;

$$\text{TrasHomFin} = \begin{vmatrix} 0 & 0 & 1 & 605 \\ 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 890 \\ 0 & 0 & 0 & 1 \end{vmatrix} \quad (3.1)$$

3.2.1. Estudio de cinemática inversa

Los manipuladores de seis grados de libertad que tienen una muñeca esférica, se pueden estudiar en dos sistemas separados. Las primeras tres articulaciones se determinan gráficamente utilizando la información de la posición actual. Las últimas tres articulaciones se pueden encontrar analíticamente utilizando ángulos de Euler. Dada una configuración deseada del efector final $[X, Y, Z, A, B, C]$, se busca obtener $[\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6]$, que son los ángulos de las articulaciones deseadas para alcanzar la posición deseada.



Figura 3.6: Brazo Manipulador KR6 R900 Sixx en posición Home

Fuente: Propia

Para encontrar los primeros tres ángulos se simplificó de manera más sencilla en dos modelos propuestos, como se muestra en la imagen [Figura 3.7].

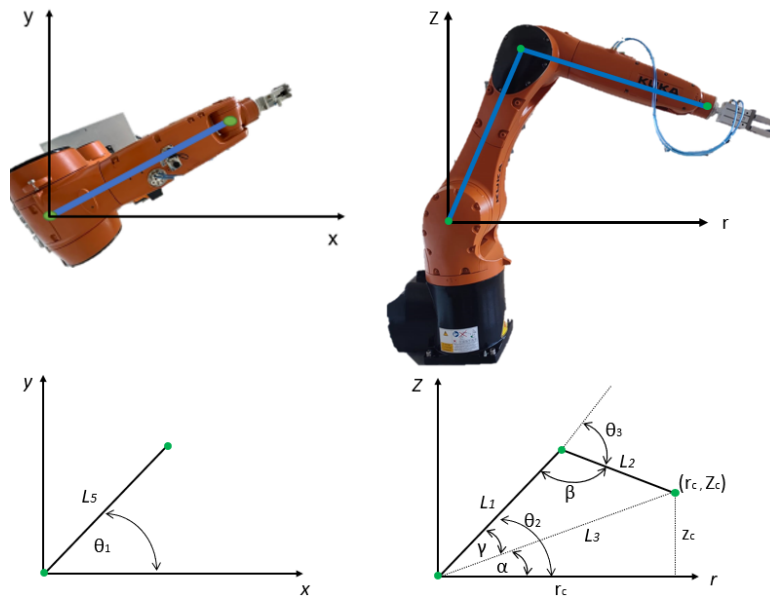


Figura 3.7: Simplificación del brazo manipulador

Fuente: Propia

Despejando y obteniendo las ecuaciones correspondientes para los primeros 3 ángulos (θ_1 , θ_2 y θ_3). Para encontrar los últimos tres valores conjuntos: cualquier rotación en el espacio de un cuerpo rígido se puede capturar utilizando tres ángulos conocidos como ángulos de Euler (α, β, γ). Las representaciones de ángulos de Euler tienen doce conjuntos diferentes de ángulos. Se utiliza la representación de ángulos de Euler Z Y Z. [14].

Tal que teniendo toda la teoría se formula un código en Python para obtener las ecuaciones, como se muestra a continuación.

Cinemática Inversa código Python

```
import numpy as np
from sympy import symbols, atan2, acos
from sympy import cos, sin

# Definir los símbolos de las variables simbólicas
theta1, theta2, theta3 = symbols('theta1 theta2 theta3')
theta4, theta5, theta6 = symbols(' theta4 theta5 theta6')

# Definir las matrices de rotación conocidas R6_3
```

```

s11, s12, s13, s21, s22 = symbols('s11 s12 s13 s21 s22')
s23, s31, s32, s33 = symbols(' s23 s31 s32 s33')
#R6_3 = np.array([[s11, s12, s13],
#                [s21, s22, s23],
#                [s31, s32, s33]])

R6_3 = np.array([[np.cos(theta4)*np.cos(theta5)*np.cos(theta6))
                 -(np.sin(theta4)*np.sin(theta6)),
                 -(np.cos(theta6)*np.sin(theta4))-(np.cos(theta4)*
                 np.cos(theta5)*np.sin(theta6))
                 ,np.cos(theta4)*np.sin(theta5)],
                 [(np.cos(theta5)*np.cos(theta6)*np.sin(theta4))
                 +(np.cos(theta4)*np.sin(theta6)),
                 (np.cos(theta5)*np.sin(theta4)*np.sin(theta6))
                 -(np.cos(theta5)*np.sin(theta4)*np.sin(theta6))
                 , np.sin(theta4)*np.sin(theta5)],
                 [-np.cos(theta6)*np.sin(theta5),
                 np.sin(theta5)*np.sin(theta1), np.cos(theta5)]]])

# Calcular los valores de las últimas tres articulaciones (4, 5, 6)
theta4 = atan2(R6_3[1, 2], R6_3[0, 2])
theta5 = atan2(np.sqrt(R6_3[0, 2]**2 + R6_3[1, 2]**2), R6_3[2, 2])
theta6 = atan2(R6_3[2, 1], -R6_3[2, 0])

# Calcular los valores de las primeras tres articulaciones (1, 2, 3)
xc, yc, zc = symbols('xc yc zc')
L1, L2, L3 = symbols('L1 L2 L3')

theta1 = atan2(yc, xc)
rc = np.sqrt(xc**2 + yc**2)
alpha = atan2(zc, rc)
gamma = acos((L2**2 + L3**2 - L1**2) / (2 * L2 * L3))

```

```
theta2 = -(alpha + gamma)
theta3 = np.pi - acos((L1**2 + L2**2 - L3**2) / (2 * L1 * L2))

# Imprimir las soluciones
print("1:", theta1)
print("2:", theta2)
print("3:", theta3)
print("4:", theta4)
print("5:", theta5)
print("6:", theta6)
```

3.3. Actividades Complementarias

3.3.1. Colocación del espacio de trabajo

Durante la obtención de parámetros del robot industrial marca KUKA se necesitaba una zona preventiva cuadrada de $2.66\text{ m} \times 2.66\text{ m}$ (el máximo alcance del robot industrial es de 900 mm como lo marca su hoja de datos), cuya finalidad es una mayor seguridad y evitar choques u/o accidentes en contra del robot provenientes de terceras personas.



Figura 3.8: Zona de trabajo del robot manipulador

Fuente: Propia

3.3.2. Exhibición de resultados

En la XII Feria de carreras 2023 del 14 al 17 de marzo en el Tecnológico Nacional de México Campus Veracruz , durante ese tiempo se impartieron cursos básicos del Brazo manipulador KUKA KR6 R900 Sixx y los resultados de los cálculos de cinemática directa e inversa a jóvenes de diversas instituciones de bachillerato publicas tales como COBAEV, CONALEP, CBTIS, etc.



Figura 3.9: Jovenes del CBTIS durante XII Feria de carreras 2023

Fuente: Propia



Figura 3.10: Jovenes del CONALEP durante XII Feria de carreras 2023

Fuente: Propia

Capítulo 4

RESULTADOS

4.1. Obtención de la cinemática directa

Se obtuvieron cálculos de cinemática directa, los cuales son importantes en el campo de la robótica y la ingeniería de control de movimiento. Estos cálculos son fundamentales para comprender y controlar los movimientos de los robots y otros sistemas mecánicos.

Se obtuvieron diversos resultados, tales como;

- Determinación de la posición y orientación del extremo del robot: Estos resultados se expresan utilizando matrices de posición y orientación como se muestra en la matriz (4.1).

$$T_6^0 = \begin{vmatrix} 0 & 0 & 1 & 605 \\ 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 890 \\ 0 & 0 & 0 & 1 \end{vmatrix} \quad (4.1)$$

Matriz de transformación homogénea

- Trayectoria del extremo del robot: Obtención de la trayectoria completa que sigue el extremo del robot al moverse desde un punto a otro. Esto implica obtener información acerca de los puntos de inicio y fin, así como los puntos intermedios que el extremo del robot atraviesa durante el movimiento.

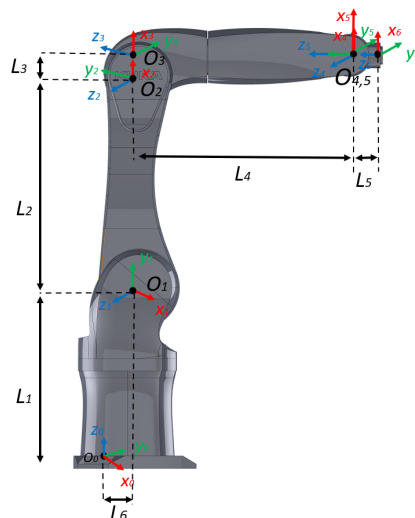


Figura 4.1: Sistemas del robot KR6 R900 Sixx

Fuente: Propia

4.2. Obtención de la cinemática inversa

En la cinemática inversa el objetivo fue determinar los valores de las articulaciones y parámetros de entrada necesarios para lograr una posición y orientación deseadas en el extremo del robot o sistema. Esto es control cinemático y preciso de los robots, permite traducir los objetivos de posición y orientación en comandos de movimiento para las articulaciones. Ecuaciones de los ángulos θ del robot industrial KR 6 R900 Sixx.

$$\theta_1 = \arctan^2 \frac{y_c}{x_c} \quad (4.2)$$

$$\theta_2 = [-(\arctan^2(z_c, r_c))] + [\cos^{-1}(\frac{L_3^2 + L_1^2 - L_2^2}{2 * L_1 * L_2})] \quad (4.3)$$

$$\theta_3 = \pi - [\cos^{-1}(\frac{L_1^2 + L_2^2 - L_3^2}{2 * L_1 * L_2})] \quad (4.4)$$

$$\theta_4 = \arctan^2(s_{2,3}, s_{1,3}) \quad (4.5)$$

$$\theta_5 = \arctan^2(\sqrt{s_{2,3} + s_{1,3}}, s_{3,3}) \quad (4.6)$$

$$\theta_6 = \arctan^2(s_{3,2} - s_3, 1) \quad (4.7)$$

Cada una de las ecuaciones θ_0 a θ_6 representa un cálculo específico basado en las coordenadas y dimensiones del sistema. Estas ecuaciones utilizan funciones trigonométricas, como arco tangentes y cosenos inversos, así como términos algebraicos que involucran longitudes y relaciones entre los componentes del robot.

En conclusión, estas ecuaciones de la cinemática inversa proporcionan una herramienta matemática para determinar los valores de los ángulos necesarios para lograr una configuración deseada del robot, lo cual es fundamental en aplicaciones industriales y de automatización donde se requiere un movimiento preciso y controlado.

Conclusiones, propuesta de mejora y experiencia profesional adquirida

Conclusión

En resumen, utilizar Python para obtener ecuaciones cinemáticas proporciona una herramienta poderosa y versátil para analizar el movimiento en sistemas robóticos. La combinación de la programación y la matemática permite desarrollar algoritmos eficientes que simplifican el cálculo de variables clave tales como matrices de rotación, parámetros DH y la posición y orientación. Al aprovechar la flexibilidad de Python y sus bibliotecas especializadas entre ellas Sympy y Numpy, podemos automatizar tareas repetitivas. Las ecuaciones de cinemática directa e inversa del robot industrial KUKA KR6 R900 Sixx, son fundamentales para lograr una programación y control eficiente del robot, y proporcionan una base sólida para futuros desarrollos y mejoras en su desempeño. Al programarse, basándose en los cálculos obtenidos, se logra compartir conocimientos con otros estudiantes interesados en el tema. Esta labor de enseñanza contribuye a la difusión y promoción de la robótica, fomentando el crecimiento de la comunidad académica y profesional en estos campos.

Propuestas de Mejora

Se observan problemas con la base del laboratorio de Robótica e inteligencia artificial donde esta instalado el robot, el cual con la reproducción de programas del Robot industrial marca KUKA “KR6 R900 Sixx” a una velocidad media-alta, se presentan vibraciones en los cimientos los cuales podrían afectar a la estructura del edificio en un futuro.

Además, la ausencia de un compresor neumático impide al robot KUKA KR6 R900 Sixx llevar a cabo diversas operaciones, como la sujeción y liberación de objetos mediante dispositivos tipo pinza. Esta limitación tiene un impacto significativo en las funcionalidades y aplicaciones del robot, ya que se ve imposibilitado de desempeñar tareas que requieran este tipo de manipulación.

4.3. Experiencia profesional adquirida

- Programación y manipulación del robot industrial.
- Mantenimiento preventivo y correctivo en equipos eléctricos y mecánicos.
- Conocimiento de softwares de diseño como SolidWorks.
- Uso de Latex-Overlaf de forma avanzada.
- Uso de Microsoft Office de forma avanzada.

Competencias desarrolladas y/o Aplicadas por el residente

- Coordinar y dirigir grupos multidisciplinarios fomentando el trabajo en equipo para la implementación de proyectos mecatrónicos, asegurando su calidad, eficiencia, productividad y rentabilidad.
- Integrar, instalar, construir, optimizar, controlar, mantener, administrar y/o automatizar sistemas mecánicos utilizando tecnología eléctricas, electrónicas y herramientas computacionales.
- Ser creativo, emprendedor y comprometido con la actualización profesional continua e autónoma, para estar a la vanguardia en los cambios científicos y tecnológicos que se dan en el ejercicio de su profesión.
- Desarrollar capacidad de liderazgo, comunicación e interrelaciones personales para transmitir ideas, facilitar conocimientos, trabajar en equipos multidisciplinarios y multiculturales con responsabilidad colectiva para la solución de problemas y desarrollo de proyectos con un sentido crítico y autocrítico.

Fuentes de información

- [1] Instituto Tecnológico de Veracruz. (s.f.). *TecNM* Disponible en: <https://www.veracruz.tecnm.mx/> Fecha de acceso: 30 de abril de 2023.
- [2] Instituto Tecnológico de Veracruz. (s.f.). *Misión, Visión, Valores.*, Disponible en: <https://www.veracruz.tecnm.mx/index.php/itver/mision-vision-valores> Fecha de acceso: 30 de abril de 2023.
- [3] Khalil W. and Dombre, E. *Modeling, Identification and Control of Robots*. London,UK. Ediciones Hermes Penton Science, 2002.
- [4] Zhang,Dhan .*Parallel Robotic Machine Tools.*..New York, USA. Editorial Springer US.(2010)
- [5] Andres Vivas *DISEÑO Y CONTROL DE ROBOTS INDUSTRIALES: TEORÍA Y PRÁCTICA* Colombia , Publisher: Elaleph.
- [6] Instituto Politecnico N. *Estructura de un robot industrial* Disponible en :<https://www.cecyt3.ipn.mx/estudiantes/plan%20continuidad/Archivo%20comprimido2/morfologia%20de%20un%20robot.pdf> Fecha de acceso: 30 de abril de 2023
- [7] Yi Cao, Ke Lu, Xiujuan Li and Yi Zang *Accurate Numerical Methods for Computing 2D and 3D Robot Workspace* . Institute of Advanced Automation Technology, Henan University of Technology, China , 2011.
- [8] Ignacio J. V. ,José L., Sergio H., Martin O., Beatrice P. *SIMULADOR DE UN ROBOT DE 6GL USANDO LabVIEW Y PYTHON*. Publicado en REVISTA ELECTRO, Vol. 44 pp. 188-193, Oct 2022, Chihuahua, Chih. México
- [9] " JupyterLab Documentation." Web. Fecha de acceso: abril de 2023. <https://jupyterlab.readthedocs.io/en/latest/>
- [10] Barrientos, A., Peñín, L. F., Balaguer, C., 2007, Fundamentos de robótica, 2a. ed.

- [11] KUKA Roboter GmbH *KUKA System Software*. Version: KSS 8.3 SI V4 ©2015
- [12] Yulia Frumer *The Short, Strange Life of the First Friendly Robot*, *IEEE SPECTRUM*
 Disponible en : <https://spectrum.ieee.org/the-short-strange-life-of-the-first-friendly-robot>.
 21 MAY 2020
- [13] Denavit, J. and Hartenberg, R.S. (1955). A kinematic notation for lower-pair mechanisms based on matrices. *Journal of Applied Mechanics*, Vol. 22, pp. 215–221.
- [14] Cardona i Foix, S. *Teoría de máquinas*. España:Edicions UPC.(2000).
- [15] El Libro de Python *Palabras reservadas en Python*, Disponible en:<https://ellibrodepython.com/palabras-reservadas-python> Fecha de acceso: mayo de 2023.
- [16] Numpy *NumPy documentation* Disponible en: <https://numpy.org/doc/stable/> Fecha de acceso: mayo de 2023.
- [17] Martín H. O. , Manuel O. M. , Carlos C. A. , Juan R. P. *Robótica: Análisis, modelado, control e implementación* OmniaScience (Omnia Publisher SL) 2015.