

DESARROLLO DE SOFTWARE DE SIMULACION QUE PERMITA RECREAR EL MOVIMIENTO DEL MODELO SIMPLIFICADO DEL ROBOT MITSUBISHI MOVEMASTER RVM1 A TRAVES DE ALGORITMOS DE CINEMATICA DIRECTA Y CINEMATICA INVERSA.

(RAE)

1. **TIPO DE DOCUMENTO:** Trabajo realizado sobre el desarrollo de un software que recree el movimiento de un robot Mitsubishi movemaster rv m1, para obtener el título de ingeniero mecatronico.
2. **TITULO:** Desarrollo de software de simulacion que permita recrear el movimiento del modelo simplificado del robot mitsubishi movemaster rvm1 a traves de algoritmos de cinematica directa y cinematica inversa.
3. **AUTORES:** Jaime Andres Fernandez Vega, Jorge Andres Rodriguez Carrizosa
4. **LUGAR:** Bogota (Cundinamarca)
5. **FECHA:** Julio 2011
6. **PALABRAS CLAVES:** Cinematica inversa, Cinematica Directa, Articulacion, Robot, Actuadores, Manipulador, Sensores, Interfaz, Software, Arquitectura, C++, Grados Libertad, Controladores, Comunicación, Dispositivo, Scara.
7. **DESCRIPCION DEL TRABAJO:** El presente trabajo es una propuesta de un software de simulación que permita recrear el movimiento del modelo simplificado del robot Mitsubishi movemaster RV M1 El robot Mitsubishi Movemaster RVM1 que se encuentra en el laboratorio de automatización, se desea emplear como elemento para la enseñanza en cátedras de robótica, sistemas flexibles de manufactura y otras similares dentro del currículo de Ingeniería Mecatrónica de la Universidad San Buenaventura sede bogota.
8. **LINEA DE INVESTIGACION:** Este trabajo se desarrolla en el marco de la línea institucional de la Facultad De Ingenieria 'profesionales con conocimientos científico, critico, y humano'.

- 9. FUENTES CONSULTADAS:** Libro Theory of Applied Robotics: kinematics, dynamics and control de Reza Jazar. Australia. Libro Robotica Industrial de Victor Santos, Portugal, Aveiro, Noviembre2002. Herramienta para la programación y simulación de movimientos del manipulador robotico Mitsubishi movemaster RV-m1, colombia, Cali, Agosto 2006. Manual de operación Basico de Uriel Guterrez Salazar, 1987. Manual básico Manipulador Robot Mitsubishi RV-M1 de Beatriz Delgado Fernandez. Libro Robotica de manipuladores: Teoria y algoritmos para modelado, simulación y control, Andres Jaramillo, 2006. Manual Mitsubishi Micro-Robot System RV-M1 MoveMaster EX Technical, Mitsubishi Corporation, Japón, 1996.
- 10. CONTENIDOS:** En este trabajo se encuentra el planteamiento del Desarrollo de software de simulación que permita recrear el movimiento del modelo simplificado del robot Mitsubishi Movemaster RVM1 a través de algoritmos de cinemática directa y cinemática inversa, analizando desde aspectos generales como particulares en la creación del software. El proyecto a desarrollar permite subsanar falencias asociadas entre el usuario y el manipulador robótico, con fin de conocer sus características y poder programar movimientos que permitan ejecutar tareas de índole industrial.
- 11. METODOLOGIA:** Este trabajo tiene todos los detalles relacionados con el desarrollo de la interfaz gráfica que en últimas se considera el producto de este proyecto, allí se podrán encontrar los pormenores y todo el proceso que permitió llevar a una aplicación los resultados obtenidos durante la etapa de cálculos, los resultados de estas simulaciones son comparados de manera directa con un software comercial como una forma de validar los datos obtenidos.
- 12. CONCLUSIONES:** Con esta propuesta permite la creación de una aplicación que simular visualiza un brazo robot Mitsubishi RV M1 de 5 grados de libertad, permitiendo configurar el tipo de trayectoria, y calculando de manera apropiada las posiciones de cada una de las articulaciones. Esto hará de la herramienta un elemento apropiado para la enseñanza de temas de Robótica a nivel de pregrado y muy probablemente permita el avance de nuevos proyectos investigativos en el tema. Además la herramienta resultante es bastante liviana, podrá ser ejecutada en un ordenador de capacidades de cómputo normales, facilita la creación de laboratorios virtuales económicamente más accesibles a las instituciones educativas y matemáticamente confiables.

DESARROLLO DE SOFTWARE DE SIMULACION QUE PERMITA RECREAR EL
MOVIMIENTO DEL MODELO SIMPLIFICADO DEL ROBOT MITSUBISHI
MOVEMASTER RVM1 A TRAVES DE ALGORITMOS DE CINEMATICA DIRECTA
Y CINEMATICA INVERSA.

JAIME ANDRÉS FERNANDEZ VEGA
JORGE ANDRÉS RODRÍGUEZ CARRIZOSA

UNIVERSIDAD DE SAN BUENAVENTURA-BOGOTÁ
FACULTAD DE INGENIERÍA
PROGRAMA ACADÉMICO DE INGENIERÍA MECATRÓNICA
JULIO DE 2011

DESARROLLO DE SOFTWARE DE SIMULACION QUE PERMITA RECREAR EL
MOVIMIENTO DEL MODELO SIMPLIFICADO DEL ROBOT MITSUBISHI
MOVEMASTER RVM1 A TRAVES DE ALGORITMOS DE CINEMATICA DIRECTA
Y CINEMATICA INVERSA.

JAIME ANDRÉS FERNANDEZ VEGA
JORGE ANDRÉS RODRÍGUEZ CARRIZOSA

Trabajo de Grado presentado como requisito para optar al título de Ingeniero
Mecatrónico

Docente:

LEONARDO RODRÍGUEZ ORTIZ

UNIVERSIDAD DE SAN BUENAVENTURA-BOGOTA
FACULTAD DE INGENIERÍA
PROGRAMA ACADÉMICO DE INGENIERÍA MECATRÓNICA
BOGOTÁ
JULIO DE 2011

Nota de aceptación

Presidente de jurado

Jurado

Jurado

Bogotá D.C., Julio de 2011

DEDICATORIA

Este trabajo de grado está dedicado a nuestras familias, que se sacrificaron y se esforzaron por nosotros, a la infinita paciencia y colaboración que nos brindaron para cumplir nuestro proyecto de vida como Ingenieros Mecatrónicos y llegar hasta donde estamos llegando.

AGRADECIMIENTOS

A mi Padre, por el esfuerzo y trabajo que ha hecho por brindarme apoyo económico y moral, también por los consejos brindados para hacer de mi una mejor persona a lo largo de mi vida y de toda la carrera.

A mi Madre por estar siempre conmigo en las buenas y en las malas situaciones de mi vida, además de la orientación y paciencia durante estos 5 años de estudio.

A todos mis hermanos que siempre están conmigo ofreciéndome algo que aprender de cada una de sus vidas, y en general a toda mi familia.

A mi compañero Andrés por haber cursado junto a mi gran parte de este proceso y apoyar para que el proceso siguiera adelante.

Jorge Andrés

A mi mamá, por su apoyo moral y económico durante toda mi vida y mi carrera.

A mi papá, por el ejemplo que me ofrecido.

A mi hermana, que siempre estuvo a mi lado.

A toda mi familia, que es mi fuerza y puerto seguro.

A Jorge, más que un compañero, amigo ejemplar en la carrera.

A todos los docentes que siempre me enseñaron de la mejor manera y fueron un ejemplo en la carrera.

Jaime Andrés

Agradecimientos especiales del grupo de trabajo:

Al Ingeniero Leonardo Rodríguez Ortiz por su paciencia y conocimiento brindado a lo largo de estos dos años, además del compromiso que adquirió con el grupo de trabajo.

A los rectores Fray Fernando Garzón (Q.E.P.D) y Fray José Wilson Téllez por su acompañamiento durante todo el proceso educativo su permanente disposición y desinteresada ayuda.

A la universidad por generarnos los espacios y el conocimiento adecuado para desarrollar el proyecto.

Y a todas aquellas personas que de una u otra forma, colaboraron o participaron en la realización de esta investigación hacemos extensivo nuestro gran sincero agradecimiento.

TABLA DE CONTENIDO

ABSTRACT	2
RESUMEN.....	2
INTRODUCCIÓN.....	3
1 OBJETIVOS	5
1.1 Objetivo General.....	5
1.2 Objetivos Específicos.....	5
2 PLANTEAMIENTO DEL PROBLEMA	6
2.1 Antecedentes o Estado del Arte.....	6
2.1.1 OpenGL:.....	6
2.1.2 Wardy 2:.....	8
2.1.3 Robot Manipulator Simulation (RMS).....	9
2.1.4 A PC-based Open Architecture Controller for Robots (OACR).....	9
2.1.5 Effective Simulation and Control Techniques for Alleviating Access to High-Cost Manipulators.....	11
2.1.6 Web-based 3D Simulator and Command Line Interface for the Mitsubishi Movemaster EX	12
2.1.7 ROBOMOSP: Robotics Modeling and Simulation Platform.....	13
2.2 Descripción y Formulación del Problema	15
2.3 Justificación	15
3 MARCO DE REFERENCIA	16
3.1 Componentes y mecanismos de un sistema robótico.....	17
3.1.1 Eslabón	17
3.1.2 Articulación	18
3.1.3 Manipulador	20
3.1.4 La muñeca	20
3.1.5 Actuadores.....	21
3.1.6 Sensores.....	21
3.1.7 Controladores.....	21
3.2 Clasificación de los Robots.....	22
3.2.1 Por su geometría	22
3.2.2 Por sus Actuadores.....	26

3.2.3	Control.....	27
3.2.4	Por su aplicación	27
3.3	Movimiento cinemático	29
3.3.1	Movimiento de cuerpo rígido.....	29
3.3.2	Transformación homogénea.....	30
3.3.3	Transformación Homogénea Inversa	33
3.4	Cinemática Directa	35
3.4.1	Notación Denavit-Hartenberg	35
3.4.2	Transformación de coordenadas entre dos marcos adyacentes	38
3.4.3	Posición cinemática directa de los robots.....	41
3.5	Cinemática Inversa.....	43
3.5.1	Técnica de Desacoplamiento	43
3.5.2	Técnica de transformación inversa.....	44
3.5.3	Técnica iterativa.....	45
4	METODOLOGÍA.....	49
5	ALCANCES Y LIMITACIONES	50
6	DESARROLLO INGENIERIL Y ANÁLISIS DE RESULTADOS	52
6.1	Características del robot Mitsubishi Movemaster RV-M1	52
6.1.1	Dimensiones.....	55
6.2	Método Denavit-Hartenberg.....	56
6.3	Matrices de transformación.....	58
6.4	Cinemática Directa	59
6.4.1	Ejemplo:	64
6.5	Cinemática Inversa.....	66
6.5.1	Ejemplo	69
6.6	Creación de la Interfaz.....	72
6.7	Comparación software CIROS Festo Automation con el software desarrollado en este proyecto.	77
6.7.1	Cinemática directa.....	77
6.7.2	Cinemática inversa.....	87
7	CONCLUSIONES.....	98
8	BIBLIOGRAFÍA.....	100

9	ANEXOS	101
9.1	Manual del software	101
9.2	Desarrollo algebraico cinemática inversa	113

ÍNDICE DE FIGURAS

Figura 1 Interfaz Software OpenGL.....	7
Figura 2 Interfaz principal Software Wardy 2.....	8
Figura 3 Aplicación `Robot Manipulator Simulation'.	9
Figura 4 Aplicación realizada 'A PC-based Open Architecture Controller for Robots'	11
Figura 5 Simulación del manipulador del proyecto en Visual Basic.....	12
Figura 6 Proyecto `Web-based 3D Simulator and Command Line Interface for the Mitsubishi Movemaster', junto a una cámara Web.	13
Figura 7 ROBOMOSP, con un Mitsubishi PA-10 y la dinámica de una trayectoria.....	14
Figura 8 Mecanismo retorno de 5 eslabones.....	18
Figura 9 Tipos de Articulación.....	18
Figura 10 Ejemplo de manipulador.....	20
Figura 11 Funcionamiento de robot SCARA.....	23
Figura 12 Robot Puma.....	24
Figura 13 Funcionamiento robot esférico	25
Figura 14 Funcionamiento de un robot cilíndrico	25
Figura 15 Robot cartesiano.....	26
Figura 16 Movimiento de cuerpo rígido en un sistema de coordenadas.....	29
Figura 17 Rotación y translación de un punto arbitrario B.....	31
Figura 18 Ejemplo de Matrices de transformación y rotación	33
Figura 19 Rotación y translación de un cuerpo en un sistema de coordenadas B con respecto al sistema G	34
Figura 20 Notación Denavit-Hartenberg parámetros cinemáticos.....	35
Figura 21 Articulaciones (i-1) , (i) y (i+1) en dos sistemas de coordenadas.....	36
Figura 22 Posición del sistema final en base al sistema base.....	42
Figura 23 Fotografía Robot Mitsubishi Movemaster RV-M1	52
Figura 24 Ejes de las articulaciones del robot Mitsubishi Movemaster RV-M1	53
Figura 25 Dimensiones y límites del movimiento articular del robot Mitsubishi Movemaster RV-M1	54
Figura 26 Relación movimientos del robot y nomenclatura.....	54
Figura 27 Movimientos del robot en coordenadas cartesianas	55
Figura 28 Nomenclatura de los eslabones del robot	56
Figura 29 Primera implementación de la interfaz.....	72
Figura 30 Segunda Implementación de la interfaz.....	73
Figura 31 Tercera implementación de la interfaz	74
Figura 32 Cuarta implementación de la interfaz.....	75
Figura 33 Desarrollo final de la interfaz	76
Figura 34 Posición del manipulador simulada en CIROS Festo Automation.....	78
Figura 35 valores ingresados a CIROS Festo Automation.....	78
Figura 36 Resultado de las coordenadas finales en CIROS Festo Automation.....	79

Figura 37 Posición del manipulador simulada en el software del proyecto	80
Figura 38 valores ingresados a el software del proyecto.....	80
Figura 39 Resultado de las coordenadas finales en el software del proyecto.....	81
Figura 40 Posición del manipulador simulada en CIROS Festo Automation.....	81
Figura 41 Valores ingresados a CIROS Festo Automation	82
Figura 42 Resultado de las coordenadas finales en CIROS Festo Automation.....	82
Figura 43 Posición del manipulador simulada en el software del proyecto	83
Figura 44 Valores ingresados a el software del proyecto	83
Figura 45 Resultado de las coordenadas finales en el software del proyecto.....	84
Figura 46 Posición del manipulador simulada en CIROS Festo Automation.....	84
Figura 47 Valores ingresados a CIROS Festo Automation	85
Figura 48 Resultado de las coordenadas finales en CIROS Festo Automation.....	85
Figura 49 Posición del manipulador simulada en el software del proyecto	86
Figura 50 Valores ingresados a el software del proyecto	86
Figura 51 Resultado de las coordenadas finales en el software del proyecto.....	87
Figura 52 Posición del manipulador simulada en CIROS Festo Automation.....	88
Figura 53 Valores ingresados a CIROS Festo Automation	88
Figura 54 Resultado de las coordenadas finales en CIROS Festo Automation.....	89
Figura 55 Posición del manipulador simulada en el software del proyecto	90
Figura 56 Valores ingresados a el software del proyecto	90
Figura 57 Resultado de las coordenadas finales en el software del proyecto.....	91
Figura 58 Posición del manipulador simulada en CIROS Festo Automation.....	91
Figura 59 Valores ingresados a CIROS Festo Automation	92
Figura 60 Resultado de las coordenadas finales en CIROS Festo Automation.....	92
Figura 61 Posición del manipulador simulada en el software del proyecto	93
Figura 62 Valores ingresados a el software del proyecto	93
Figura 63 Resultado de las coordenadas finales en el software del proyecto.....	94
Figura 64 Posición del manipulador simulada en CIROS Festo Automation.....	94
Figura 65 Valores ingresados a CIROS Festo Automation	95
Figura 66 Resultado de las coordenadas finales en CIROS Festo Automation.....	95
Figura 67 Posición del manipulador simulada en el software del proyecto	96
Figura 68 Valores ingresados a el software del proyecto	96
Figura 69 Resultado de las coordenadas finales en el software del proyecto.....	97

ÍNDICE DE TABLAS

Tabla 1	Espacio de operación y dimensiones del robot Mitsubishi Movemaster RV-M1	50
Tabla 2	Dimensiones del robot Mitsubishi Movemaster RV-M1	55
Tabla 3	Clasificación geométrica del robot Mitsubishi Movemaster RV-M1	57
Tabla 4	Parámetros DH para el manipulador Mitsubishi RV-M1	57

ABSTRACT

This project describes the phases for the construction of a virtual simulation of a robot with 5 degrees of freedom, using a mathematical model to simulate the full movement of the different joints and their position in space through mathematical software like Matlab ®, all of this possible thanks to algorithms for forward and inverse kinematics applied to the Mitsubishi robot Movemaster RVM1.

RESUMEN

El robot Mitsubishi Movemaster RVM1 que se encuentra en el laboratorio de automatización se desea emplear como elemento para la enseñanza en cátedras de robótica, sistemas flexibles de manufactura y otras similares dentro del currículo de Ingeniería Mecatrónica de la Universidad San Buenaventura.

La aplicación aquí propuesta permitirá interactuar con el robot a través de una simulación de sus trayectorias sin importar la complejidad de ellas, teniendo en cuenta las características físicas del mismo, facilitando su uso y aprendizaje. Mostrando una interfaz grafica en un plano tridimensional en el que se visualicen sus movimientos cinemáticos en tiempo real, a través de un código en un lenguaje conocido en entornos académicos.

El proyecto se desarrollará dese las etapas básicas del diseño, selección de plataforma de software y de simulaciones mecánicas, lo cual llevará al usuario a aprender el manejo del robot y la programación de sus movimientos, el conocimiento en profundidad de su cinemática directa e inversa, alineación de ejes, configuraciones singulares, etc.

INTRODUCCIÓN

La Universidad busca formar a los futuros profesionales tanto en lo teórico como en lo práctico, por eso, trabajos investigativos como los de este tipo que pretenden llevar a la práctica los conocimientos adquiridos en las aulas de clase resultan una excelente alternativa para la creación de una nueva cultura al interior de la Academia.

Este proyecto cuyo título es “Desarrollo de software de simulación que permita recrear el movimiento del modelo simplificado del robot Mitsubishi Movemaster RVM1 a través de algoritmos de cinemática directa y cinemática inversa” desenvuelve el problema, analizando desde aspectos generales como particulares en la creación del software. El proyecto a desarrollar permite subsanar falencias asociadas entre el usuario y el manipulador robótico, con fin de conocer sus características y poder programar movimientos que permitan ejecutar tareas de índole industrial.

Un correcto análisis de la situación en cuanto a trabajos similares desarrollados en diferentes lugares del planeta soportan inicialmente los objetivos planteados, así mismo esta primera investigación aportó una posible metodología de trabajo que se plasma de manera apropiada dentro del capítulo de desarrollo ingenieril, allí se muestra la totalidad de las ecuaciones y procedimientos matemáticos que se tuvieron en cuenta para poder determinar con exactitud la manera como el manipulador robótico realizará sus movimientos, obviamente para llegar hasta este punto se cubren algunos aspectos teóricos básicos que se consideran fundamentales para el lector menos relacionado con temas de robótica.

El trabajo cierra con todos los detalles relacionados con el desarrollo de la interfaz gráfica que en últimas se considera el producto de este proyecto, allí se podrán encontrar los pormenores y todo el proceso que permitió llevar a una aplicación los resultados obtenidos durante la etapa de cálculos, los resultados de estas simulaciones son comparados de manera directa con un software comercial como una forma de validar los datos obtenidos.

Tanto la propuesta como sus resultados muestran una solución para la recuperación del equipo de robótica en cuestión (Robot Mitsubishi RV-M1),

brindando de esta forma aportes para entornos académicos y logrando un beneficio para todos aquellos interesados en aprender del tema. Se espera que a través de su lectura el interés por la creación de nuevos grupos de investigación y nuevos desarrollos en el tema, pues aunque el trabajo es amplio, existen aún muchos aspectos por explorar en esta interesante área

1 OBJETIVOS

1.1 Objetivo General

Desarrollar una herramienta que permita manipular el robot Mitsubishi de forma simulada o en tiempo real, para aprender a utilizar el robot y facilitar su programación, con el fin de ejecutar tareas en entornos académicos.

1.2 Objetivos Específicos

- Recolectar la adecuada información acerca de los antecedentes trabajados en este proyecto.
- Desarrollar una interfaz gráfica que permita la representación de las trayectorias de cada una de las articulaciones del robot, y a su vez una utilización sencilla de las opciones mostradas.
- Evaluar las trayectorias dadas por el usuario, considerando las características físicas del robot, comparar los resultados con software comercial o el sistema real.
- Incrementar el número de funciones del robot que se encuentren a disposición del usuario no experimentado para que este no requiera entrenamiento para programar el robot.
- Permitir la interacción del usuario con el sistema de almacenamiento de datos del controlador del robot, con el fin de facilitar la gestión de listado de posiciones y el programa almacenado.
- Facilitar la comunicación entre el robot y el PC, permitiendo el envío de datos completos o línea por línea, según los requerimientos del usuario, en relación con el manejo de los programas y los listados de posiciones.

2 PLANTEAMIENTO DEL PROBLEMA

2.1 Antecedentes o Estado del Arte

Antes de comenzar se hace necesario aclarar que gracias al convenio institucional entre la Universidad y Festo Ltda. se logró la donación del robot Mitsubishi RV-M1 para fines investigativos, el hardware del equipo funcionaba apropiadamente pero tanto el sistema de control como las herramientas de programación se encontraban con daños permanente o simplemente no existían, así que con el firme propósito de recuperar la herramienta se generaron dos proyectos de investigación uno de ellos en la parte de diseño electrónico y otro enfocado en las herramientas de software, este proyecto muestra los resultados obtenidos en esta última.

El robot como tal en su estado original incluía unas herramientas de simulación adecuadas para la programación del mismo, de hecho, y con la ayuda de Festo se logró llevar este simulador hasta las aulas de clase a través de la herramienta Ciroso[®], sin embargo, debido a que el modelo data de finales de los 80, el soporte técnico ha sido totalmente discontinuado dejando al usuario final la responsabilidad por la creación de aplicaciones como las que se propone en este trabajo. A partir de lo anterior varios esfuerzos se han generado en diversas instituciones educativas en el mundo, algunas de las cuales se resumen a continuación:

2.1.1 OpenGL:

OpenGL¹ (Open Graphics Library) es una especificación estándar que define una API multilenguaje y multiplataforma para escribir aplicaciones que produzcan gráficos 2D y 3D. La interfaz consiste en más de 250 funciones diferentes que pueden usarse para dibujar escenas tridimensionales complejas a partir de primitivas geométricas simples, tales como puntos, líneas y triángulos. Fue desarrollada originalmente por Silicon Graphics Inc. (SGI) en 1992 y se usa ampliamente en CAD, realidad virtual, representación científica, visualización de información y simulación de vuelo. También se usa en desarrollo de videojuegos, donde compite con Direct3D en plataformas Microsoft Windows.

¹Extraído de <http://es.wikipedia.org/wiki/OpenGL>

Fundamentalmente OpenGL es una especificación, es decir, un documento que describe un conjunto de funciones y el comportamiento exacto que deben tener. Partiendo de ella, los fabricantes de hardware crean implementaciones, que son bibliotecas de funciones que se ajustan a los requisitos de la especificación, utilizando aceleración hardware cuando es posible. Dichas implementaciones deben superar unos test de conformidad para que sus fabricantes puedan calificar su implementación como conforme a OpenGL y para poder usar el logotipo oficial de OpenGL.

OpenGL tiene dos propósitos esenciales:

- Ocultar la complejidad de la interfaz con las diferentes tarjetas gráficas, presentando al programador una API única y uniforme.
- Ocultar las diferentes capacidades de las diversas plataformas hardware, requiriendo que todas las implementaciones soporten la funcionalidad completa de OpenGL (utilizando emulación software si fuese necesario).

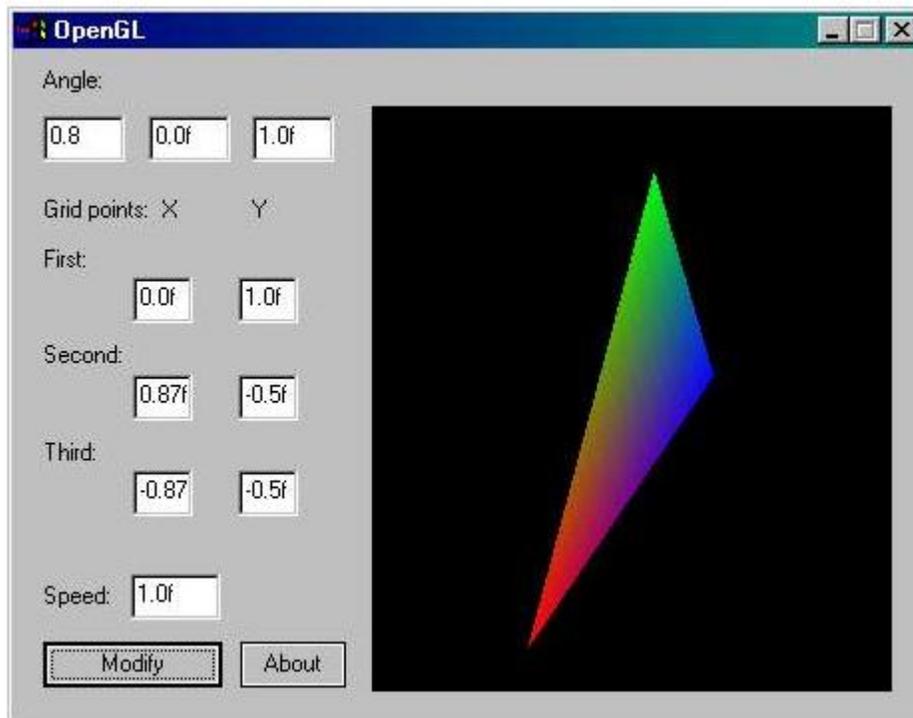


Figura 1 Interfaz Software OpenGL

2.1.2 Wardy 2:

Wardy es una herramienta básica, cuya capacidad está limitada a las siguientes condiciones:

Recibir el listado de posiciones grabadas en el controlador del manipulador, almacenándolas en un archivo de texto.

Enviar desde un archivo de texto un listado de posiciones, para ser grabadas en el controlador del manipulador.

Recibir, en un archivo de texto, el programa cargado actualmente en la memoria del controlador del manipulador.

Enviar, desde un archivo de texto, un programa, para ser almacenado en la memoria del controlador del manipulador.

Su aplicación sólo envía y recibe archivos de texto, a través del puerto serial, con algunas consideraciones específicas para los manipuladores del Robot Mitsubishi RV-M1 y RV-M2.



Figura 2 Interfaz de selección de manipulador Software Wardy 2.

Este simulador tiene varios defectos conocidos, que causan inconvenientes a los usuarios, por ejemplo se bloquea con gran facilidad tras cualquier modificación.

2.1.3 Robot Manipulator Simulation (RMS)

Este es un proyecto de simulación del manipulador en cuestión con OpenGL en Windows. Puede proporcionar una visualización 3D de alta calidad en tiempo real con animación de simulación de hardware de bajo nivel

Incluye entre otras funciones las siguientes:

- Simulación 3D animados en tiempo real.
- Cálculo de la Cinemática Manipulador, incluyendo PUMA-560 y EX-Movemaster.
- 3 métodos de control: control manual, control de Pose, Mixta de Controlde cambio

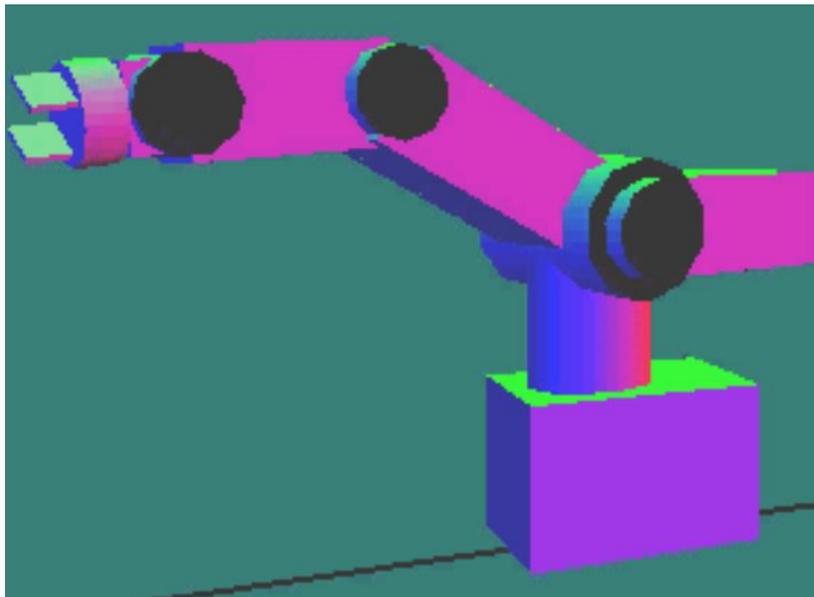


Figura 3Aplicación `Robot Manipulator Simulation`.

2.1.4 A PC-based Open Architecture Controller for Robots (OACR)².

CORBA (arquitectura común de intermediarios en peticiones a objetos), es un estándar que establece una plataforma de desarrollo de sistemas distribuidos facilitando la invocación de métodos remotos bajo un paradigma orientado a objetos.

² Pan, Liandong, Xinhan, Huang y Arif, Mohammad. <http://www.ansinet.org/fulltM1t/itj/itj33296-302.pdf>.

CORBA fue definido y está controlado por el Grupo de Gestión de Objetos (OMG) que define la interfaz de programación de aplicaciones (API), el protocolo de comunicaciones y los mecanismos necesarios para permitir la interoperabilidad entre diferentes aplicaciones escritas en diferentes lenguajes y ejecutadas en diferentes plataformas, lo que es fundamental en computación distribuida.

En un sentido general, CORBA "envuelve" el código escrito en otro lenguaje, en un paquete que contiene información adicional sobre las capacidades del código que contiene y sobre cómo llamar a sus métodos.

El desarrollo (OACR) Plantea una arquitectura basada en CORBA, para controlar cualquier manipulador, mediante componentes con interfaces estandarizadas. Su finalidad es reemplazar las unidades de control especializadas de los manipuladores, por un sistema genérico basado en computadores personales y hardware más económico.

Para evaluar la arquitectura, se realiza una prueba de concepto, reemplazando la unidad de control del Movemaster RV-M1 por una genérica, basada en una tarjeta PCI, para controlar directamente los servos del manipulador. Cuenta con un motor de cinemática y dinámica muy completo, una representación tridimensional del robot y la creación de trayectorias sencillas.

Generalmente las pruebas se realizan en lenguaje C++ bajo Windows, pero al ser CORBA, en teoría, los componentes podrían estar en cualquier plataforma.

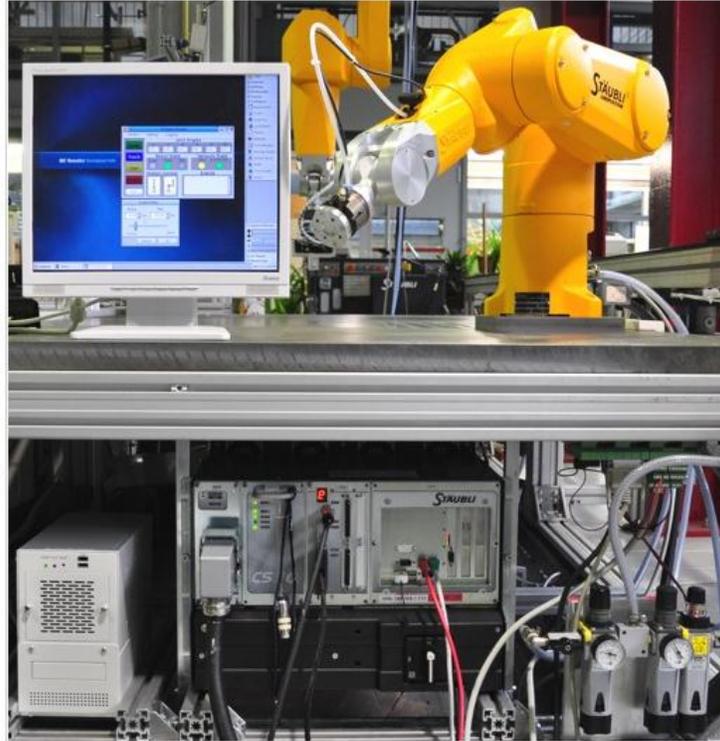


Figura 4 Aplicación realizada 'A PC-based Open Architecture Controller for Robots'

2.1.5 Effective Simulation and Control Techniques for Alleviating Access to High-Cost Manipulators³.

Es un proyecto que fue desarrollado como recurso para la enseñanza de diferentes cursos, en Bridgeport University de Connecticut, USA, con lo cual se pretendía reducir el desgaste que presentaban los manipuladores, debido al uso excesivo que hacían, de ellos, los estudiantes. Este proyecto incluye, entre sus características: el cálculo de la cinemática directa e inversa, el control del robot por medio del movimiento articular o cartesiano, y la coordinación entre los movimientos del robot virtual y el real, cuando este último se encuentra conectado al PC.

Otra característica que puede resaltarse del proyecto es que permite el uso de diversos tipos de curvas, con un número parametrizable de puntos de control e intermedios, para el diseño de trayectorias complejas. Igualmente, cabe resaltar la capacidad para operar en red, de tal forma que el estudiante visualiza en pantalla, en calidad de cliente, lo mismo que el instructor, y puede tener acceso, desde su equipo, al control del robot, si este último lo autoriza. Como complemento de lo anterior, la aplicación permite el control del robot, desde teléfonos móviles.

³ Mihali, Raúl y Sobh, Tarek, <http://www.bridgeport.edu/~sobh/pdf/paper43.doc>

Esta aplicación se desarrolló en Visual Basic, por lo cual su uso está limitado a Windows, y utiliza OpenGL para la vista en tres dimensiones.

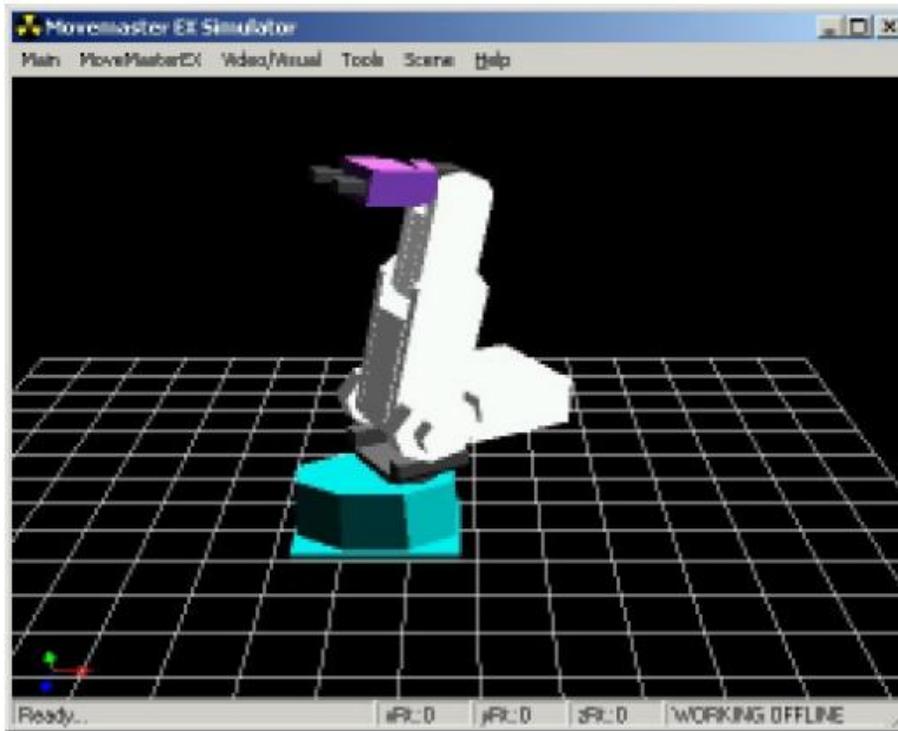


Figura 5 Simulación del manipulador del proyecto en Visual Basic

2.1.6 Web-based 3D Simulator and Command Line Interface for the Mitsubishi Movemaster EX⁴.

Proyecto desarrollado por la Universidad de las Filipinas, en el año 2000. Implementa una aplicación Web, con un simple simulador de 3D del robot en VRML⁵, la cual soporta el ingreso remoto de un número limitado de comandos del robot, simulando su ejecución en la pantalla. Puede estar conectada a un robot real y transmitirle los comandos ingresados por el usuario. Una alternativa, que se ilustra en la Figura 6, es colocar la aplicación en una página junto a una cámara Web, lo cual permitiría observar los movimientos, desde lugares remotos. Implementa la cinemática inversa.

⁴ Restituto-Tecson, Joel y Cruz-Santos, Marianne.
<http://www.upd.edu.ph/~yamatake/old/projects/1999-2000/telerobot.pdf>

⁵ Virtual Reality Markup Language (VRML), <http://en.wikipedia.org/wiki/VRML>

Es una aplicación Web realizada en Java, que presenta el robot virtual por medio de VRML, y ejecuta el código Perl CGI, en el servidor. Por esto, para el cliente es multiplataforma.

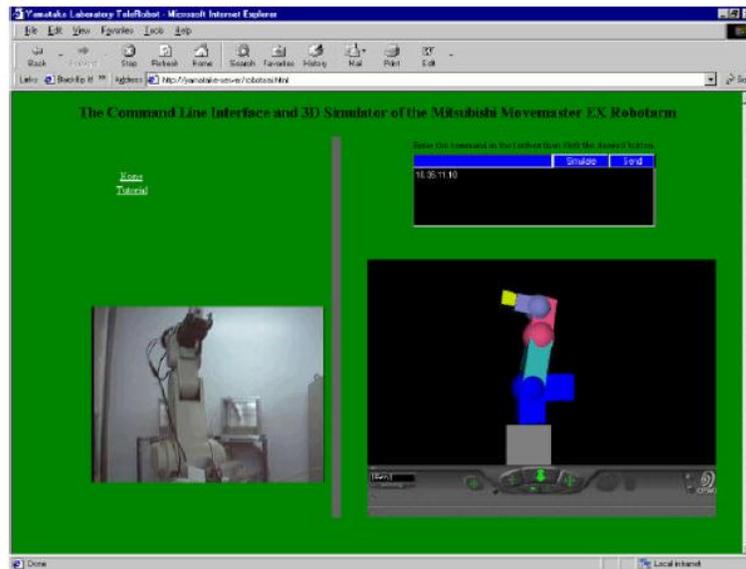


Figura 6 Proyecto 'Web-based 3D Simulator and Command Line Interface for the Mitsubishi Movemaster', junto a una cámara Web.

2.1.7 ROBOMOSP: Robotics Modeling and Simulation Platform⁶.

Proyecto académico desarrollado por el mismo grupo de investigación de este proyecto⁷. Implementa una plataforma modular que permite diseñar, simular y controlar cualquier tipo de robot manipulador. Es un sistema CAD⁸ en 3 dimensiones, que cuenta con múltiples capacidades como: modelado de cualquier manipulador y de sus componentes físicos; composición del entorno de trabajo con varios manipuladores; composición de trayectorias complejas; simulación y control dinámico del manipulador, con modelado de torques de sus masas; simulación y control cinemática; programación de tareas complejas, mediante un lenguaje de alto nivel; interfaz de programación (API), para ampliar sus capacidades, y operación remota, por medio de sockets TCP/IP.

⁶ <http://gar.puj.edu.co/Investigacion/Grupos/GAR/TmpGar/projects/robomosp/>. Dos artículos citados en la

bibliografía, por Jaramillo Botero, Andrés, et al.

⁷ Grupo de Automatización Y Robótica de la Pontificia Universidad Javeriana Cali <http://gar.puj.edu.co/>

⁸ Computer Aided Design, Diseño Asistido por Computadora

Es una herramienta con un objetivo general, ya que se aplica para cualquier tipo de manipulador y ambiente de trabajo, de naturaleza avanzada, con una arquitectura abierta y modular. Su modelado de la dinámica es completo, siendo esta una característica que lo diferencia, estando ausente aun en las soluciones comerciales evaluadas.

Es un proyecto diseñado, desde su inicio, como multiplataforma, construido con herramientas de software de dominio público, y puede ejecutarse en sistemas operativos variados, como Linux, MacOS X, y Microsoft Windows.

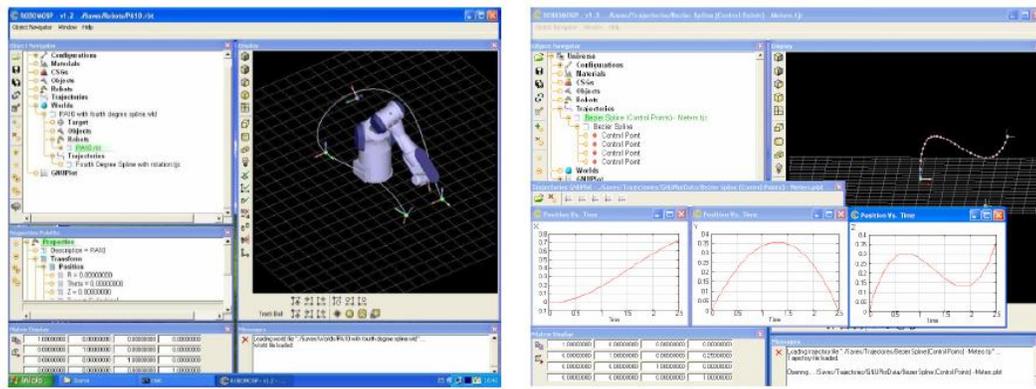


Figura 7 ROBOMOSP, con un Mitsubishi PA-10 y la dinámica de una trayectoria.

2.2 Descripción y Formulación del Problema

Los estudios de la robótica son de gran importancia para los estudiantes de la Universidad de San Buenaventura, y elementos como el Robot Mitsubishi Movemaster RV-M1 son de gran ayuda didáctica para estos entornos académicos. Actualmente el robot se encuentra en condiciones de deterioro y debido a que su modelo es antiguo, no posee el soporte técnico necesario para darle utilidad a esta importante herramienta que tiene la universidad.

En los procesos de manufactura y en algunas industrias es muy común el trasladar elementos y productos que necesitan de mano de obra para las respectivas necesidades reduciendo gastos y ampliando la producción, lo cual hace interesante el estudio de manipuladores para poder restaurar y recuperar el Robot Mitsubishi RV-M1 y lograr entender su funcionamiento desde el punto de vista de la robótica.

El diseño de software de simulación es de gran importancia para muchos procesos y el desarrollo de nuevas tecnologías, lo cual hace que la siguiente pregunta problema cobre toda su importancia, ¿Cómo desarrollar una herramienta que permita manipular el robot Mitsubishi RV-M1 en simulación o tiempo real para aprender a utilizarlo y facilitar su programación, con el fin de ejecutar tareas en entornos académicos?

2.3 Justificación

Para la solución del problema es necesario estudiar de forma precisa las herramientas y variables encontradas, partiendo de lo más básico como la solución de las ecuaciones que permitan describir el movimiento, para ser estudiadas de manera matricial e interactuar con el computador siguiendo los parámetros adecuados.

La programación se puede considerar como una herramienta en la educación que brinda una cantidad de alternativas para simular los comportamientos de procesos industriales de los seres humanos, siendo esto un preámbulo al desarrollo de los algoritmos necesarios para implementar estos requerimientos que son útiles para estudios posteriores.

Además de lo anterior, es interesante aclarar que la robótica se encuentra en una gran etapa de desarrollo y las aplicaciones de estos software son muy grandes y es por ello que se demuestra en este documento lo que la universidad de San Buenaventura pretende investigar para desarrollar cada día los procesos industriales actualmente en nuestro país.

3 MARCO DE REFERENCIA

El desarrollo de conceptos para robótica ha sido y seguirá siendo uno de los estudios principales del ser humano para desarrollar futuras y mejores tecnologías. En donde se debe tener en cuenta un conjunto de leyes como formulaciones matemáticas impresas en los senderos positrónicos del cerebro de los robots. Y que establecen lo siguiente:

- Un robot no puede hacer daño a un ser humano o, por inacción, permitir que un ser humano sufra daño.
- Un robot debe obedecer las órdenes dadas por los seres humanos, excepto si estas órdenes entrasen en conflicto con la anterior ley.
- Un robot debe proteger su propia existencia en la medida en que esta protección no entre en conflicto con las anteriores dos leyes.

Las anteriores leyes se consideraron por Isaac Asimov en 1942. Cuando se desarrolla una herramienta para programar y simular movimientos del manipulador robótico Mitsubishi se utiliza un seguimiento de una trayectoria, es preciso solucionar el problema cinemático asociado al manipulador. Lo anterior implica el cálculos de las diferentes cinemáticas que se toman en cuenta para permitir el desarrollo apropiado del movimiento de las articulaciones y de la estructura mecánica en el espacio, para permitir la representación gráfica del robot en el espacio tridimensional, mientras ejecuta secuencialmente los movimientos articulares requeridos, para seguir la trayectoria definida por el usuario.

El desarrollo de los cálculos cinemáticos se realizará a partir de diferentes matrices de posición relacionadas con una ubicación en el espacio. Para ello se estudian las variables físicas del robot, teniendo en cuenta cada una de sus características de arquitectura para evitar problemas en cuanto a la posición..

La cinemática directa se calculará a partir del desarrollo de la matriz de transformación homogénea algorítmica, que es representada en MATLAB teniendo en cuenta el plano en el que se encuentra desarrollado todo el software.

Para la cinemática inversa se utilizará el método geométrico. Este método se ha escogido por su eficiencia computacional, ya que requiere pocos cálculos, y las características del manipulador, que permiten su uso, como el reducido número de articulaciones del manipulador, la limitación de movimiento en la tercera articulación (de 0° a -110°), que hace innecesario el uso de métodos más complejos, como el Jacobiano.

Desde el punto de vista de la ingeniería, los robots son dispositivos complejos y versátiles que contienen una estructura mecánica, un sistema sensorial, y un sistema de control automático. Fundamentos teóricos de la robótica se basan en los resultados de la investigación en la mecánica, la electrónica, control automático, matemáticas y ciencias de informática. Para ello es importante tener en cuenta definiciones reales para cada uno de los componentes de los robots y lograr representar mediante medios aritméticos y matemáticos las cinemáticas del brazo robot MITSUBISHI RV-M1.

3.1 Componentes y mecanismos de un sistema robótico.

Los manipuladores robóticos están compuestos cinemáticamente de eslabones conectados por articulaciones formando así una cadena cinemática. Un sistema robótico generalmente está compuesto del manipulador, la muñeca, el efector final, actuadores, sensores, controladores, procesadores y software.

Es importante aclarar que los brazos de un Robot, a menudo son categorizados por sus grados de libertad (por lo general más de seis grados de libertad). Este número generalmente se refiere al número de un solo eje de rotación de las articulaciones en el brazo, donde un mayor número indica una mayor flexibilidad en posicionar una herramienta. Es decir que los grados de libertad se refieren al movimiento en un espacio tridimensional, es decir, la capacidad de moverse hacia delante/atrás, arriba/abajo, izquierda/derecha (traslación en tres ejes perpendiculares), combinados con la rotación sobre tres ejes perpendiculares.

De esta manera es importante conocer los componentes que están asociados con el estudio de los manipuladores y que se definen a continuación:

3.1.1 Eslabón

Los cuerpos rígidos individuales que conforman un robot son llamados eslabones. En la robótica ocasionalmente se usa el significado de brazo para denotar un eslabón. Un brazo robot eslabón o brazo robot es un miembro rígido que puede tener movimiento relativo con respecto a los otros eslabones. Desde el punto de vista cinemático, dos o más miembros conectados entre sí, de tal manera que no exista movimiento relativo entre ellos, se consideran un solo eslabón. Ver figura (8).

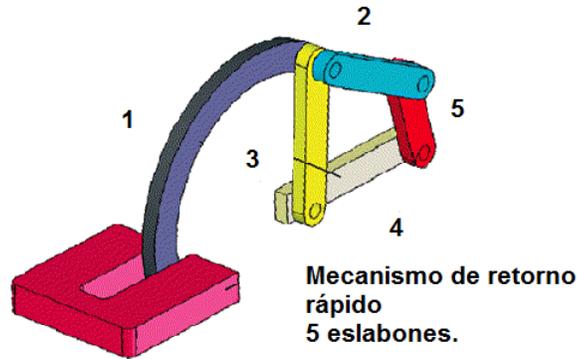


Figura 8 Mecanismo retorno de 5 eslabones

3.1.2 Articulación

Dos líneas unidas por el contacto de una articulación en su movimiento relativo, se puede expresar por una sola coordenada. Las articulaciones suelen ser de revolución (rotatorio) o prismáticas (lineal). En la figura (9) se representa estos dos tipos de articulaciones. Una articulación de revolución (R), es como una bisagra que permite la rotación relativa entre dos eslabones. Una articulación prismática (P), permite el movimiento relativo entre los dos eslabones.

La rotación relativa de eslabones conectados por una articulación de revolución ocurre sobre una línea llamada eje de articulación como se muestra en la figura (9). De la misma manera sucede con el eslabón conectado a partir de una articulación prismática ya que también tiene su eje de articulación. El valor de la única coordenada que describe la posición relativa de dos eslabones en una articulación, es denominada variable de articulación

Las articulaciones activas son usualmente las prismáticas y las de revolución, sin embargo, las articulaciones pasivas pueden ser cualquiera de los pares más bajos que proporcionan superficie de contacto. Hay seis diferentes tipos de de articulaciones de par más bajo; revolución, prismática, cilíndrica, de tornillo, esférica y plana. Ver figura (10).

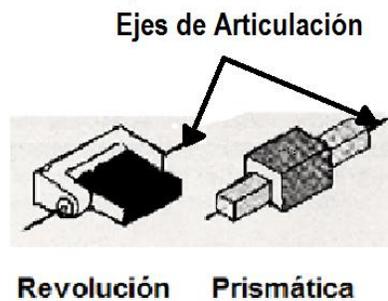


Figura 9 Articulaciones de revolución y prismáticas con sus respectivos ejes de articulación.

Las articulaciones de revolución y las prismáticas son las más comunes y utilizadas en los manipuladores robóticos en serie. Los otros tipos de articulaciones son simplemente implementados para lograr la misma función o proveer más grados de libertad.

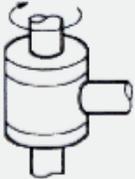
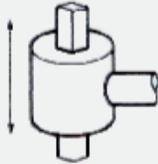
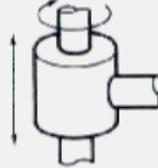
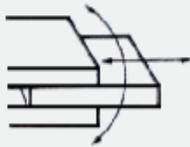
ESQUEMA	ARTICULACIÓN	GRADOS LIBERTAD
	ROTACIÓN	1
	PRISMÁTICA	1
	CILINDRICA	2
	PLANAR	2
	ESFÉRICA (RÓTULA)	3

Figura 10 Tipos de Articulación⁹

⁹ Extraído de <http://www.monografias.com/trabajos16/estacion-robotica/estacion-robotica.shtml>

3.1.3 Manipulador

El cuerpo principal de un robot está compuesto por eslabones, articulaciones, y otros elementos estructurales, es denominado manipulador. Un manipulador llega a ser un robot cuando la muñeca y la pinza están unidas, y el sistema de control es implementado. Sin embargo, en literatura de robots y manipuladores se utilizan de forma equivalente y ambos se refieren a robots.

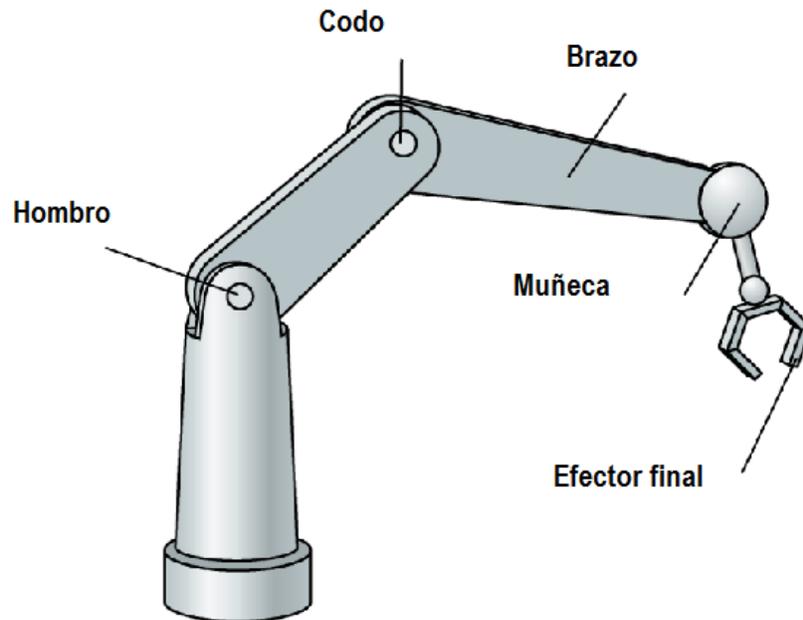


Figura 11 Ejemplo de manipulador

3.1.4 La muñeca

Las articulaciones en la cadena cinemática de un robot entre el antebrazo y el efecto final, se conocen como la muñeca. Esta es comúnmente diseñada para manipuladores con muñeca esférica., por el cual tres ejes de la articulación se cruzan en un punto en común llamado punto de muñeca.

La muñeca esférica simplifica enormemente el análisis cinemático efectivo, lo que permite separar el posicionamiento y la orientación del efecto final. Por tanto, el manipulador tendrá tres grados de libertad para posición, dependiendo de las tres articulaciones del brazo. Por lo anterior se infiere que se debe hacer el diseño de la muñeca teniendo en cuenta uno, dos o tres grados de libertad dependiendo de la aplicación que se requiera.

3.1.5 Actuadores

Los actuadores son los motores que en este caso funcionan como músculos del robot, para cambiar su posición. Los actuadores proporcionan energía para mover la estructura mecánica venciendo la gravedad, inercia y otras fuerzas externas que modifican la geometría de la ubicación de la mano del robot. Los tipos de actuadores pueden ser eléctricos, hidráulicos o neumáticos, y deben ser controlables.

3.1.6 Sensores

Los elementos usados para detectar y recolectar información acerca de los estados internos y ambientales son los sensores. Generalmente, la posición de las articulaciones, la velocidad, la aceleración y la fuerza son las informaciones más importantes que son medidas. Los sensores se integran en el robot, envían información acerca de cada eslabón, cada articulación y cada unidad de control, determinando de esta manera la configuración del robot.

3.1.7 Controladores

Los controladores o unidades de control tienen tres reglas:

- Información de las funciones, consiste en la recolección y el procesamiento de la información entregada por los sensores.
- Decisión de las funciones, el cual consiste en la planeación del movimiento geométrico de la estructura mecánica.
- Funciones de comunicación, que consisten en la organización de la información entre el robot y el medio ambiente. La unidad de control incluye el procesador y el software.

3.2 Clasificación de los Robots

La Asociación Industrial de robótica Japonesa divide los robots en seis diferentes clases:

- Dispositivos de manipulación manual: dispositivo de varios grados de libertad que es manipulado por un operador.
- Robot de secuencia fija: Dispositivo que lleva a cabo las etapas sucesivas de una tarea, de acuerdo a un programa predeterminado y fijo.
- Robot de secuencia variable: Dispositivo que lleva a cabo las etapas sucesivas de una tarea, de acuerdo un predeterminado método de programación.
- Robot de reproducción: Un operario lleva a cabo las tareas manualmente guiando al robot, el cuál archiva los movimientos para una reproducción futura. El robot repite los mismos movimientos de acuerdo a la información recolectada.
- Robot de control numérico: El operador supe al robot con un programa de movimiento, más que enseñarle las tareas manualmente.
- Robot inteligente: Un robot con la habilidad de entender el medio ambiente y además cuenta con la habilidad de completar tareas exitosamente, a pesar de cambios bajo condiciones circundantes, las cuales pueden ser realizadas.

Además de esta clasificación oficial, los robots también se clasifican de acuerdo a otros aspectos tales como su campo de trabajo, su geometría, su control y su aplicación dependiendo del uso al que está destinado, a continuación se presentan algunas otras categorías

3.2.1 Por su geometría

Un robot se denomina un manipulador serial o de lazo abierto, si en su estructura cinemática no hace una cadena de bucle. Es denominado manipulador paralelo o de lazo cerrado si en su estructura hace una cadena de bucle. Es llamado híbrido si su estructura se compone de dos cadenas, lazo abierto y lazo cerrado.

Como un sistema mecánico, se puede pensar en un robot como un conjunto de cuerpos rígidos conectados juntos en algunas articulaciones.

La mayoría de los manipuladores industriales son de seis grados de libertad. Los manipuladores de circuito abierto se pueden clasificar basándose en sus tres primeras articulaciones empezando desde la articulación a tierra. De los dos tipos de articulaciones matemáticamente hay 72 configuraciones diferentes de manipuladores, simplemente porque cada articulación pueda ser P o R, y los ejes de las articulaciones pueden ser paralelos (\parallel), ortogonales (\perp), o perpendiculares (\perp). Dos ejes ortogonales de las articulaciones se cruzan en ángulo recto, sin embargo, dos ejes perpendiculares de las articulaciones están en ángulo recto con respecto al eje normal u original. Dos ejes perpendiculares de las articulaciones son paralelos si un eje se mueve 90 grados sobre el eje normal. Dos ejes perpendiculares de las articulaciones son ortogonales si la medida de su eje normal tiende a cero.

Fuera de las 72 configuraciones diferentes de manipuladores, las más importantes son: R \parallel R \parallel P (SCARA), R \perp R \perp R (Articulado), R \perp R \perp P (Esférico), R \parallel P \perp P (Cilíndrico), y P \perp P \perp P (Cartesiano).

- SCARA

El brazo SCARA (Robot selectivo y articulado compatible para el ensamble) es un manipulador popular, el cual, como su nombre lo indica está diseñado para operaciones de ensamble.

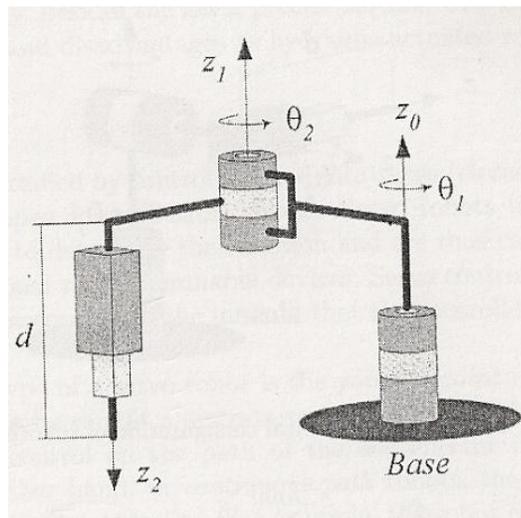


Figura 12 Funcionamiento de robot SCARA

- Articulado

La configuración del brazo articulado, es llamado codo, de revolución o antropomórfico. Es una configuración adecuada para robots industriales. Casi el 25% de los robots industriales, PUMA por ejemplo, son hechos y diseñados de este tipo.

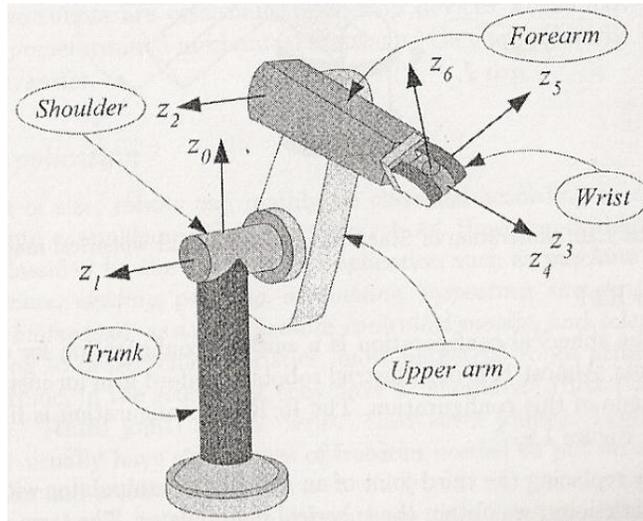


Figura 13 Robot Puma

- Esférico

La configuración del robot esférico es adecuada para pequeños robots. Casi el 15% de las industrias de robots, por ejemplo Brazo Stanford, son fabricados con esta configuración.

Mediante la sustitución de la articulación de la tercera parte de un manipulador articulado por una articulación prismática, se obtiene el manipulador esférico. El nombre de esférico, proviene del hecho de que las coordenadas esféricas definen la posición del efector final, respecto a su estructura base.

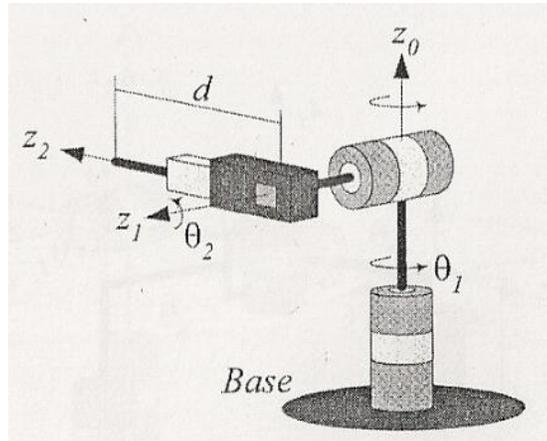


Figura 14 Funcionamiento robot esférico

- Cilíndrico

Es una configuración adecuada para robots con mediana carga de capacidad. Casi el 45% de las industrias de robots fabrican de este tipo. La primera articulación del robot cilíndrico es una articulación por revolución, y produce una rotación sobre su base, mientras la segunda y tercera articulación, son prismáticas. Como su nombre lo indica, las variables de articulación son coordenadas cilíndricas del efector final con respecto a la base.

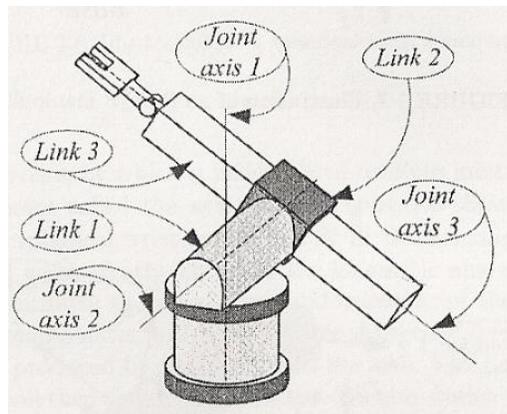


Figura 15 Funcionamiento de un robot cilíndrico

- Cartesiano

La configuración cartesiana está diseñada o adecuada para robots de alta carga de capacidad o robots largos en cuanto a dimensiones. Casi el 15% de las industrias de robots utilizan esta configuración.

Para un robot cartesiano, las variables de la articulación son coordenadas cartesianas del efector final con respecto a la base. Como es de esperar, la

descripción cinemática de este manipulador es la más simple de todos los manipuladores. Los manipuladores cartesianos son generalmente usados en aplicaciones de ensamble en mesa, o transporte de carga.

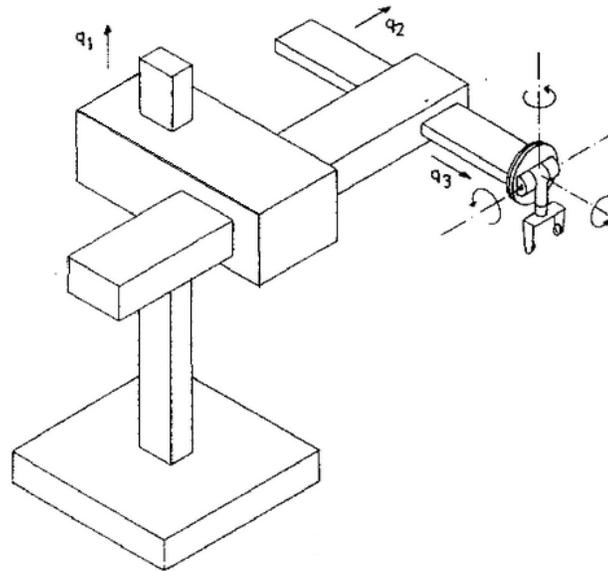


Figura 16 Robot cartesiano

3.2.2 Por sus Actuadores

Los actuadores transforman el poder en movimiento. Los robots son generalmente actuadores eléctricos, neumáticos o hidráulicos. Otro tipo de actuadores son los que se consideran como piezoeléctricos, magnetostricción, aleación con memoria de forma y poliméricos.

Los robots de actuadores eléctricos son energizados por motores AC y DC y son los más considerados aceptables para los robots. Son fáciles de limpiar, silenciosos comparándolos con los actuadores neumáticos e hidráulicos. Los motores eléctricos son eficientes a grandes velocidades por lo que una caja de cambios de relación alta es necesaria para reducir las altas revoluciones por minuto. La facilidad de manipularlos y el auto frenado son ventajas de la caja de cambios en caso de tener pérdidas de potencia. Sin embargo, cuando se necesitan altas velocidades o altas cargas, los conductores eléctricos no son capaces de competir con los actuadores hidráulicos.

Los actuadores hidráulicos son satisfactorios para bajas velocidades, alto torque, altas potencias y masas proporcionales. Por tanto, los robots con actuadores de este tipo son utilizados primordialmente para el levantamiento de cargas pesadas. Entre los aspectos negativos de los actuadores hidráulicos, se encuentra el ruido

que produce, la tendencia a tener pérdidas o fugas, incluyendo una bomba necesaria u otro dispositivo para su correcto funcionamiento.

Los robots con actuadores neumáticos son económicos y simples, pero no se pueden controlar de forma precisa. Junto a la precisión más baja de movimiento, ellos tienen casi las mismas ventajas y desventajas que tienen los robots con actuadores hidráulicos.

3.2.3 Control

Los robots se pueden clasificar por el método de control, robots con servo (control de circuito cerrado) y robot sin servo (control de circuito abierto). Los robots servo utilizan el control de circuito cerrado por computador, para determinar su movimiento y por lo que son capaces de ser dispositivos realmente multifuncionales reprogramables. Los Robots servo controlados son los más clasificados de acuerdo con el método que el controlador utiliza para guiar el efector final.

El manipulador más simple de un robot servo es el manipulador punto a punto, el cuál puede enseñar un discreto conjunto de puntos, llamados puntos de control, pero no hay control en el camino de puntos del efector final. Por otra parte, en robots de caminos de puntos continuos, el camino de puntos puede ser controlado. Por ejemplo, el robot de efector final puede enseñar a seguir una línea recta entre dos puntos o incluso a seguir un contorno como una línea de unión de soldadura. Además de que la velocidad y o aceleración del efector final puede ser a menudo controlada. Estos son los robots más avanzados y requieren los computadores controladores más sofisticados y software desarrollado.

Los robots sin servo son esencialmente dispositivos de circuito abierto cuyo movimiento es limitado o predeterminado por topes mecánicos, y ellos son utilizados para transferir materiales.

3.2.4 Por su aplicación

Independiente de su tamaño, los robots pueden ser clasificados principalmente de acuerdo a su aplicación en robots ensamblados y robots no ensamblados. Sin embargo, en las industrias ellos son clasificados por las categorías tales como robots de máquina de carga, de selección y lugar, soldadura, pintura, ensamble, de inspección, manufactura, biomédicos, asistencia, móviles a control remoto y tele robots.

De acuerdo a las características de diseño, la mayoría de las industrias de brazos robots tienen un “hombro” (primeras dos articulaciones), un “codo” (tercera articulación), y una “muñeca” (las últimas tres articulaciones). Por tanto, en total, ellos usualmente tienen seis grados de libertad necesarios para poner un objeto en una posición y orientación.

Los manipuladores seriales más comercializados tienen solo articulaciones de revolución. Comparados con las articulaciones prismáticas, las articulaciones de revolución cuestan menos y ofrecen amplia destreza por el espacio de trabajo ya que es equivalente al volumen del robot.

Es importante tener en cuenta que los robots seriales son muy pesados comparados con los de máxima carga, ya que se pueden mover sin perder exactitud. Su carga útil en relación al peso es menor que 1/10, teniendo en cuenta que criterios como la simplicidad de la posición de cinemática inversa, cinemática directa y la velocidad han sido siempre uno de los mayores criterios en la industria de los manipuladores para ser utilizados dependiendo de las características de aplicación y es por ello que casi todos ellos tienen una estructura cinemática especial.

3.3 Movimiento cinemático

El desarrollo de los movimientos de cualquier manipulador debe entenderse de acuerdo a los pasos necesarios para ubicar en un sistema de coordenadas la posición de un cuerpo y representar el movimiento de unas coordenadas a otras teniendo en cuenta la traslación y rotación en este mismo sistema. Conforme a lo anterior, el sistema de coordenadas al que está sujeto un cuerpo rígido en el espacio, está representado matricialmente como se explica a continuación en este capítulo.

3.3.1 Movimiento de cuerpo rígido

Considerando un cuerpo rígido con un sistema de coordenadas locales adheridas a B ($oxyz$) moviéndose libremente en un marco o sistema global de coordenadas fijas G ($OXYZ$). El cuerpo rígido puede rotar en un sistema de coordenadas global, mientras el punto o del cuerpo marco B puede traducir en relación al origen O de G como se muestra en la figura:

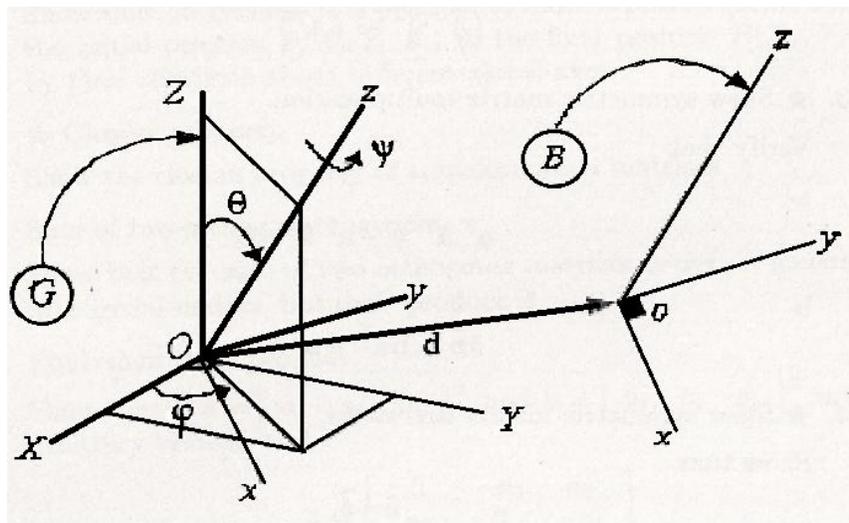


Figura 17 Movimiento de cuerpo rígido en un sistema de coordenadas

Si el vector G_d indica la posición del movimiento de origen o. en relación con el origen fijo O, entonces las coordenadas de un punto del cuerpo P en los marcos locales y globales están relacionados por la siguiente ecuación:

Ecuación 1

$${}^G r_p = {}^G R_B {}^B r_p + {}^G d$$

Dónde:

Ecuación 2

$$\begin{aligned} {}^G r_p &= [X_p \quad Y_p \quad Z_p]^T \\ {}^B r_p &= [x_p \quad y_p \quad z_p]^T \\ {}^G d &= [X_0 \quad Y_0 \quad Z_0]^T \end{aligned}$$

El vector G_d llamado el desplazamiento o translación de B con respecto a G, y G_{R_B} es la matriz de rotación al mapa B_r a G_r cuando G_d es igual a cero. Cada una de las combinaciones de la rotación y translación en la anterior ecuación es llamada movimiento rígido. En otras palabras, la ubicación del cuerpo rígido puede estar descrita por la posición del origen o y la orientación del cuerpo marco, con respecto al marco global. La descomposición del movimiento rígido en una rotación y una translación es el método más simple para representar es desplazamiento espacial. Se muestra la translación por medio de u vector.

3.3.2 Transformación homogénea

Como se muestra en la figura 18 un punto arbitrario P de un cuerpo rígido, está unido a un marco local B y está denotado por B_{r_p} y G_{r_p} en sistemas diferentes. El vector G_d indica la posición de origen o del cuerpo marco en el sistema de coordenadas global. Por tanto, un movimiento general del cuerpo rígido B (oxyz) en el marco global G (OXYZ) es una combinación de rotación G_{R_B} y translación G_d .

Ecuación 3

$${}^G r = {}^G R_B {}^B r + {}^G d$$

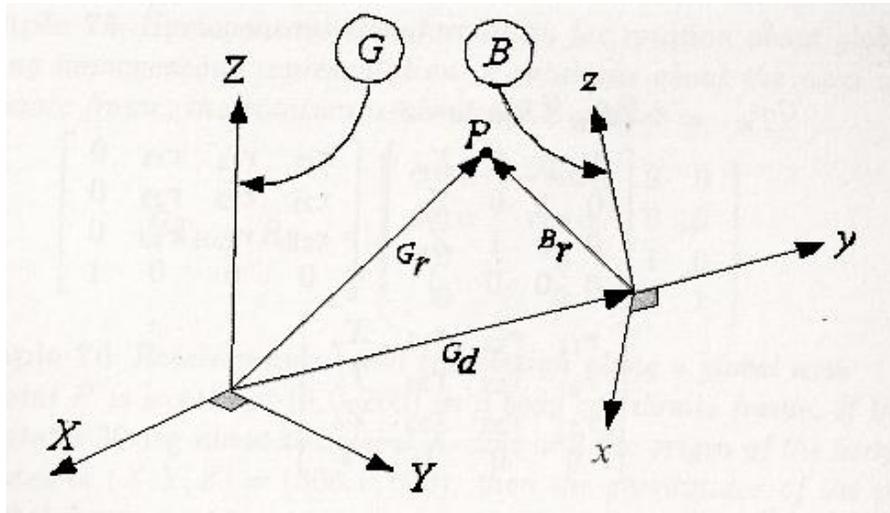


Figura 18 Rotación y translación de un punto arbitrario B

Usando la matriz de rotación más un vector nos lleva a la utilización de las coordenadas homogéneas. Introduciendo una matriz de transformación homogénea denominada G_{T_B} , ayuda a a la explicación del movimiento rígido por una simple matriz de transformación.

Ecuación 4

$${}^G r = {}^G T_B {}^B r$$

Dónde,

Ecuación 5

$${}^G T_B = \begin{bmatrix} r_{11} & r_{12} & r_{13} & X_0 \\ r_{21} & r_{22} & r_{23} & Y_0 \\ r_{31} & r_{32} & r_{33} & Z_0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^G T_B = \begin{bmatrix} {}^G R_B & {}^G d \\ 0 & 1 \end{bmatrix}$$

Y

$${}^G r = \begin{bmatrix} X_p \\ Y_p \\ Z_p \\ 1 \end{bmatrix}$$

$${}^B r = \begin{bmatrix} x_p \\ y_p \\ z_p \\ 1 \end{bmatrix}$$

$${}^G d = \begin{bmatrix} X_0 \\ Y_0 \\ Z_0 \\ 1 \end{bmatrix}$$

Las anteriores ecuaciones, en pocas palabras indican la posición del punto P representado mediante vectores de cada sistema de coordenadas G y B respectivamente. Por esta razón, cada uno de los valores de los vectores G y B son las posiciones con respecto a su sistema de coordenadas, y son multiplicados entre sí vectorialmente con el sistema de coordenadas global para obtener una coordenada final. Ver figura (20).

En conclusión, la matriz de transformación homogénea es una matriz 4 x 4 que transforma un vector de posición expresado en coordenadas homogéneas desde un sistema de coordenadas a otro. Para asimilar más fácilmente la representación se denota con T , donde T es la matriz de transformación:

$$T = \begin{bmatrix} R_{3x3} & P_{3x1} \\ F_{1x3} & 1x1 \end{bmatrix} = \begin{bmatrix} \text{rotación} & \text{translación} \\ \text{perspectiva} & \text{escalado} \end{bmatrix}$$

Ejemplo de Rotación	Ejemplo de traslación
$T_{x,\alpha} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha & 0 \\ 0 & \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$	$T_{tras} = \begin{bmatrix} 1 & 0 & 0 & px \\ 0 & 1 & 0 & py \\ 0 & 0 & 1 & pz \\ 0 & 0 & 0 & 1 \end{bmatrix}$

Figura 19 Ejemplo de Matrices de transformación y rotación

Los elementos de la diagonal principal de una matriz de transformación homogénea producen escalado local y global.

3.3.3 Transformación Homogénea Inversa

Una matriz ortogonal es una matriz cuya inversa coincide con su matriz traspuesta. Geométricamente hablando, las matrices ortogonales representan transformaciones isométricas en espacios vectoriales reales, llamadas justamente transformaciones ortogonales.

La ventaja del trabajo simple con las matrices de transformación homogénea es que ellas pierden por defecto la propiedad de ortogonalidad. Si se muestra ${}^G T_B$ como se muestra en la ecuación 6,

Ecuación 6

$${}^G T_B = \begin{bmatrix} I & {}^G d \\ 0 & 1 \end{bmatrix} \begin{bmatrix} {}^G R_B & 0 \\ 0 & 1 \end{bmatrix}$$

$${}^G T_B = \begin{bmatrix} {}^G R_B & {}^G d \\ 0 & 1 \end{bmatrix}$$

Entonces,

$${}^B T_G = {}^G T_B^{-1}$$

$${}^B T_G = \begin{bmatrix} {}^G R_B & {}^G d \\ 0 & 1 \end{bmatrix}$$

$${}^B T_G = \begin{bmatrix} {}^G R_B & -{}^G R_B^T {}^G d \\ 0 & 1 \end{bmatrix}$$

Ecuación 7

$${}^G T_B^{-1} {}^G T_B = I_4$$

Lo que demuestra que una matriz de transformación no es ortogonal si su inversa no es igual a la transpuesta

Ecuación 8

$${}^G T_B^{-1} \neq {}^G R_B^T$$

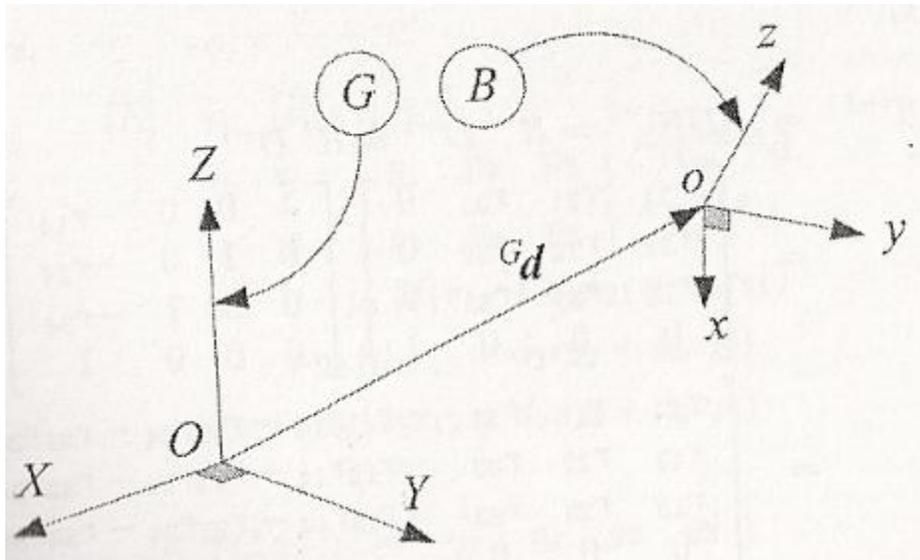


Figura 20 Rotación y translación de un cuerpo en un sistema de coordenadas B con respecto al sistema G

3.4 Cinemática Directa

3.4.1 Notación Denavit-Hartenberg

Un robot o manipulador en serie con n articulaciones tendrá $n+1$ eslabones. Numerando los eslabones desde el inicio empezando desde cero con el eslabón base que va a tierra e incrementándolo hasta el eslabón efector final. Del mismo modo sucede con las articulaciones, numerándose desde uno, iniciando desde la unión del eslabón base a el primer eslabón de movimiento, e incrementando secuencialmente hasta n . Por tal razón, el eslabón (i) está conectado a su eslabón inferior $(i-1)$ en su extremo proximal por la articulación i y está conectado a su eslabón superior $(i+1)$ en su extremo distal de las articulaciones $(i+1)$, como se muestra en la siguiente figura.

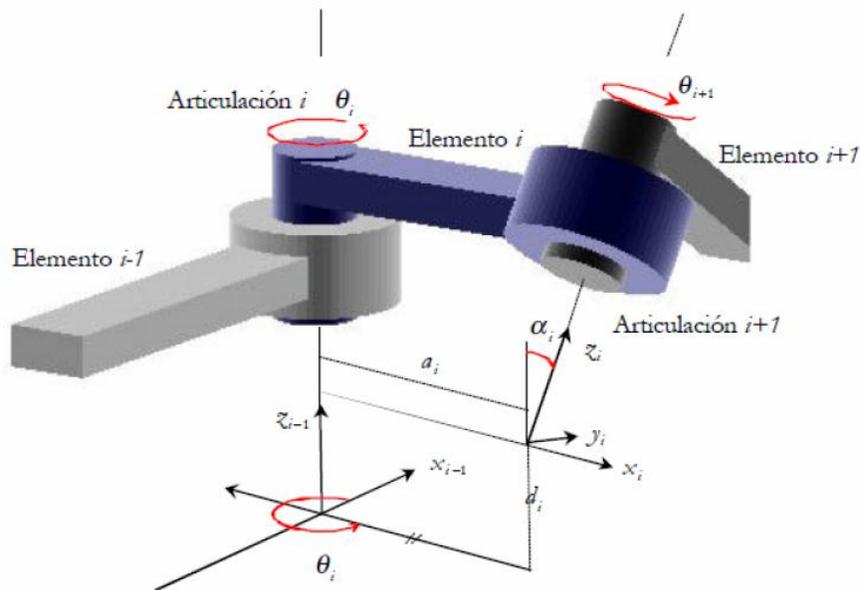


Figura 21 Notación Denavit-Hartenberg parámetros cinemáticos

En la siguiente figura se muestra los eslabones $(i-1)$, (i) y $(i+1)$ de un robot serie a lo largo de las articulaciones $i-1$, i , $i+1$. Todas las articulaciones están indicadas por su eje, el cual puede ser rotacional o de translación. Para relacionar la información cinemática de los componentes del robot, se analiza estáticamente un sistema de coordenadas locales B_i para cada eslabón (i) en la articulación $i+1$ usando un método estándar, conocido como el Método de Denavit-Hartenberg.

De esta manera se procede a desarrollar el método, desarrollando los secuencialmente los siguientes pasos:

- Numerar los eslabones comenzando con 1 (primer eslabón móvil de la cadena) y acabando con n (último eslabón móvil). Se numerará como eslabón 0 a la base fija del robot.
- Numerar cada articulación comenzando por 1 (la correspondiente al primer grado de libertad y acabando en n).
- Localizar el eje de cada articulación, si 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 como se explicó en la figura (9).
- Para i de 0 hasta $n-1$, situar el eje Z_i , sobre el eje de la articulación $i+1$ tal y como se muestra en la figura (22).
- Situar el origen del sistema de la base O en cualquier punto del eje Z_0 . Los ejes X_0 y Y_0 se situarán de modo que formen un sistema dextrógiro con Z_0 . Observe en la figura (22) que en el sistema de coordenadas el origen está situado en la parte inferior del eslabón y la articulación $i+1$.

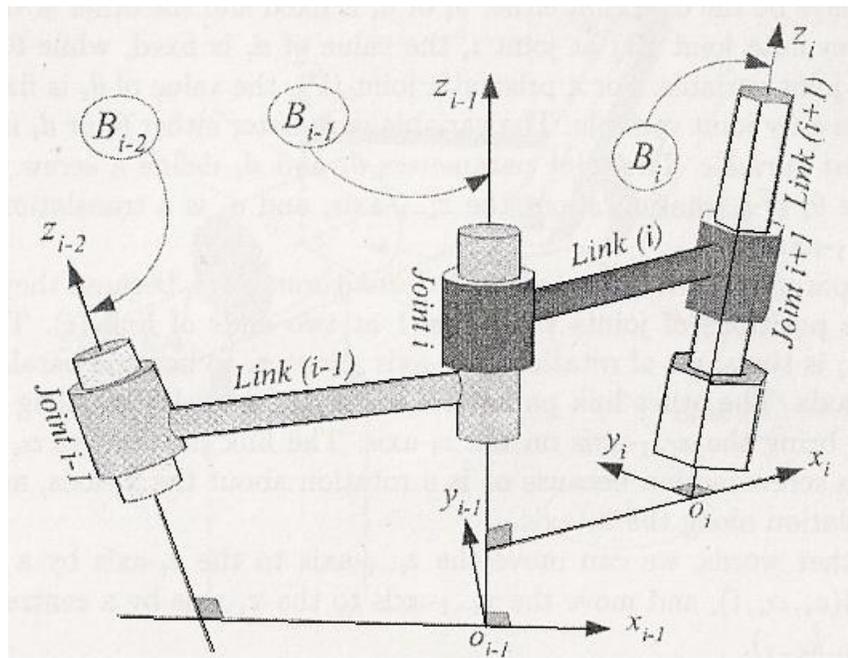


Figura 22 Articulaciones (i-1) , (i) y (i+1) en dos sistemas de coordenadas

- Para i de 1 a $n-1$, situar el sistema (O) (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

- Distancia de articulación d_i es la distancia entre los ejes x_{i-1} y x_i a lo largo del eje z_{i-1} . La distancia de articulación es también llamada eslabón de compensación.
- Ángulo de articulación θ_i es la rotación requerida por el eje x_{i-1} alrededor del eje z_{i-1} para que sea paralelo al eje x_i .

Los parámetros θ_i y d_i son llamados parámetros de articulación, puesto que definen la posición relativa de dos eslabones adyacentes conectados en la articulación i . En un robot de diseño dado, cada articulación es de revolución o prismática. Por lo tanto, por cada articulación, está será siempre para el caso que d_i o θ_i es uno fijo y el otro variable. Para una articulación de revolución (R) en la articulación i , el valor de d_i es fijo, mientras que θ_i es la única articulación variable. Para articulaciones prismáticas (P), el valor de θ_i es fijo y d_i es la única articulación variable. Las variables parámetros θ_i o d_i son también denominadas variables de articulación. Los parámetros de articulación θ_i o d_i definen el movimiento de tornillo porque θ_i es una rotación alrededor del eje z_{i-1} , y d_i es una translación a lo largo del eje z_{i-1} .

Los parámetros a_i y α_i son denominados parámetros de eslabón, porque ellos definen las posiciones relativas de las articulaciones i y $i+1$ en dos extremos del eslabón (i). El eslabón de giro α_i es el ángulo de rotación del eje z_{i-1} alrededor de x_i para llegar a ser paralelo al eje z_i . El otro parámetro de eslabón, a_i es la translación a lo largo del eje z_i para llevar al eje z_{i-1} al eje z_i . Los parámetros de eslabón a_i y α_i definen un movimiento de tornillo porque α_i es una rotación alrededor del eje x_i , y a_i es una translación a lo largo del eje x_i .

3.4.2 Transformación de coordenadas entre dos marcos adyacentes

El sistema de coordenadas B_i está fijo al eslabón (i) y el sistema de coordenadas B_{i-1} está fijo al eslabón ($i-1$). Basándose en la convención del método de Denavit-Hartenberg, la matriz de transformación ${}^{i-1}T_i$ para transformar el sistema de coordenadas B_i a B_{i-1} es representado como un producto de cuatro transformaciones básicas usando el parámetro de eslabón (i) y la articulación i que está definido de acuerdo a la nomenclatura expuesta en la sección anterior de este capítulo.

Ecuación 9

$${}^{i-1}T_i = D_{z_{i-1}, d_i}, R_{z_{i-1}, \theta_i}, D_{x_{i-1}, a_i}, R_{x_{i-1}, \alpha_i}$$

$${}^{i-1}T_i = \begin{bmatrix} \cos \theta_i & -\sin \theta_i \cos \alpha_i & \sin \theta_i \sin \alpha_i & a_i \cos \theta_i \\ \sin \theta_i & \cos \theta_i \cos \alpha_i & -\cos \theta_i \sin \alpha_i & a_i \sin \theta_i \\ 0 & \sin \alpha_i & \cos \alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Donde,

Ecuación 10

$$R_{x_{i-1}, \alpha_i} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \alpha_i & -\sin \alpha_i & 0 \\ 0 & \sin \alpha_i & \cos \alpha_i & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Ecuación 11

$$D_{x_{i-1}, a_i} = \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Ecuación 12

$$R_{z_{i-1}, \theta_i} = \begin{bmatrix} \cos \theta_i & -\sin \theta_i & 0 & 0 \\ \sin \theta_i & \cos \theta_i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Ecuación 13

$$D_{z_{i-1}, d_i} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Por tanto, la ecuación de transformación desde el sistema de coordenadas $B_i(x_i, y_i, z_i)$, a su sistema de coordenadas anterior $B_{i-1}(x_{i-1}, y_{i-1}, z_{i-1})$, es

$$\begin{bmatrix} x_{i-1} \\ y_{i-1} \\ z_{i-1} \\ 1 \end{bmatrix} = {}^{i-1}T_i \begin{bmatrix} x_i \\ y_i \\ z_i \\ 1 \end{bmatrix}$$

Donde,

$${}^{i-1}T_i = \begin{bmatrix} \cos \theta_i & -\sin \theta_i \cos \alpha_i & \sin \theta_i \sin \alpha_i & a_i \cos \theta_i \\ \sin \theta_i & \cos \theta_i \cos \alpha_i & -\cos \theta_i \sin \alpha_i & a_i \sin \theta_i \\ 0 & \sin \alpha_i & \cos \alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Esta matriz de 4x4 puede ser particionada en dos submatrices, las cuales representan una única rotación combinada con una única traslación para producir el mismo movimiento rígido requerido para mover desde B_i hasta B_{i-1} .

Ecuación 14

$${}^{i-1}T_i = \begin{bmatrix} {}^{i-1}R_i & {}^{i-1}d_i \\ 0 & 1 \end{bmatrix}$$

Donde,

$${}^{i-1}R_i = \begin{bmatrix} \cos \theta_i & -\sin \theta_i \cos \alpha_i & \sin \theta_i \sin \alpha_i \\ \sin \theta_i & \cos \theta_i \cos \alpha_i & -\cos \theta_i \sin \alpha_i \\ 0 & \sin \alpha_i & \cos \alpha_i \end{bmatrix}$$

Y

$${}^{i-1}d_i = \begin{bmatrix} a_i \cos \theta_i \\ a_i \sin \theta_i \\ d_i \end{bmatrix}$$

La matriz de transformación inversa homogénea es:

$${}^i T_{i-1} = {}^{i-1} T_i^{-1}$$

$${}^i T_{i-1} = \begin{bmatrix} \cos \theta_i & \sin \theta_i & 0 & -a_i \\ -\sin \theta_i \cos \alpha_i & \cos \theta_i \cos \alpha_i & \sin \alpha_i & -d_i \sin \alpha_i \\ \sin \theta_i \sin \alpha_i & -\cos \theta_i \sin \alpha_i & \cos \alpha_i & -d_i \cos \alpha_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

3.4.3 Posición cinemática directa de los robots

La cinemática directa y la cinemática inversa es la transformación de la información cinemática desde la variable de articulación del robot en el espacio hasta la coordenada final en el espacio del sistema de coordenadas final. En otras palabras, es la información de la articulación en un sistema de coordenadas local hasta el sistema de coordenadas global en el que se presentan los resultados de las coordenadas finales. Luego el problema de encontrar la posición y la orientación del efector final, para un determinado conjunto de variables de articulación, es el problema de la cinemática directa ya que necesitamos conocer la posición final del efector si se ingresan los datos de los ángulos de cada articulación. Este problema se puede resolver determinando las matrices de transformación ${}^0 T_i$ para describir la información cinemática del eslabón (i) en el eslabón base o tierra del sistema de coordenadas. La forma tradicional de generar las ecuaciones de cinemática directa de un robot manipulador es proceder a utilizar la notación de Denavit-Hartenberg y los sistemas de coordenadas de cada eslabón uno por uno. Por tanto, la cinemática directa es la manipulación de la matriz de transformación explicada en la sección anterior.

Para robots de seis grados de libertad, se requieren seis matrices de la notación Denavit-Hartenberg, una por cada eslabón, y son requeridas para transformar las coordenadas finales de las coordenadas base. En el caso del Robot Mitsubishi Movemaster RV-M1 son sólo cinco matrices de transformación debido a sus grados de libertad, donde el último sistema de dichas matrices está unido a la estructura final de coordenadas que generalmente se fijan en el centro de la pinza.

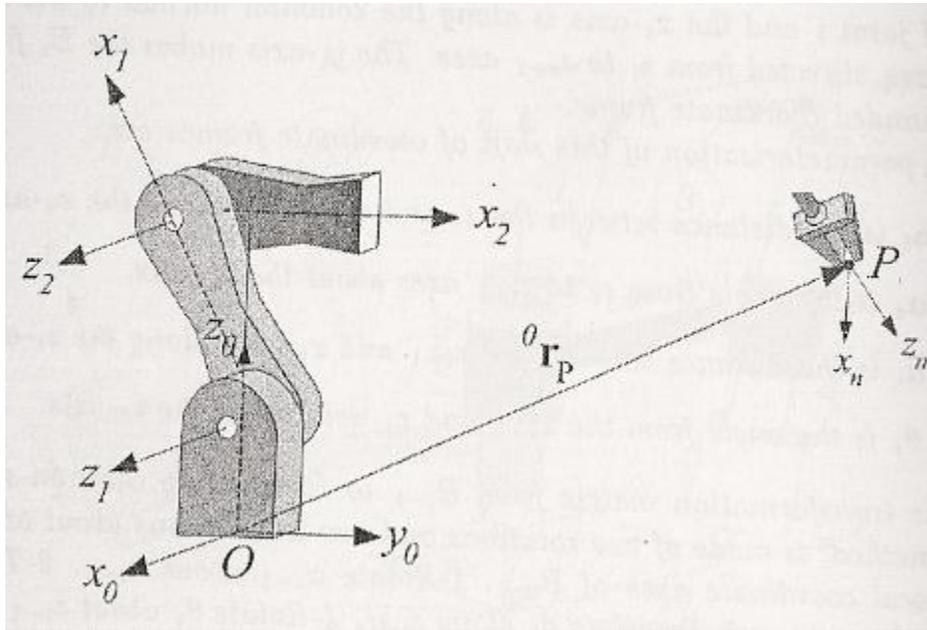


Figura 24 Posición del sistema final en base al sistema base

La información cinemática para el manipulador incluye: posición, velocidad, aceleración y arranque. Sin embargo, la cinemática directa se entiende como el análisis de la posición, luego la cinemática directa es equivalente a determinar la matriz de transformación combinada,

Ecuación 15

$${}^0T_n = {}^0T_1(q_1) {}^1T_2(q_2) {}^2T_3(q_3) {}^3T_4(q_4) \dots {}^{n-1}T_n(q_n)$$

para encontrar las coordenadas de un punto P en el sistema de coordenadas base cuando estas coordenadas están dadas en el sistema final.

$${}^0r_p = {}^0T_n {}^nr_p$$

3.5 Cinemática Inversa

3.5.1 Técnica de Desacoplamiento

La búsqueda de las variables de articulación en términos de la posición y la orientación del efector final es llamada cinemática inversa. Matemáticamente, la cinemática inversa es la búsqueda de los elementos del vector \mathbf{q} cuando una transformación está dada como una función de variables de articulación q_1, q_2, q_3, \dots

$${}^0T_n = {}^0T_1(q_1) {}^1T_2(q_2) {}^2T_3(q_3) {}^3T_4(q_4) \dots {}^{n-1}T_n(q_n)$$

Los robots controlados por computador son usualmente movidos en el espacio de la variable de articulación, sin embargo, los objetos para ser manipulados son expresados en el sistema de coordenadas global. Por tanto, llevar la información cinemática, de ida y de vuelta, entre el espacio de articulación y el espacio cartesiano que se convierte en una necesidad de las aplicaciones del robot. Para controlar la configuración del efector final para alcanzar un objeto, la solución de cinemática inversa debe ser resuelta. Por tanto, se necesita conocer cuáles son los valores requeridos de las variables de articulación, para alcanzar un punto y una orientación deseada.

El resultado de cinemática directa de un robot con seis grados de libertad es una matriz de transformación de 4x4.

Ecuación 16

$${}^0T_6 = {}^0T_1 {}^1T_2 {}^2T_3 {}^3T_4 {}^4T_5 {}^5T_6$$
$${}^0T_6 = \begin{bmatrix} r_{11} & r_{12} & r_{13} & r_{14} \\ r_{21} & r_{22} & r_{23} & r_{24} \\ r_{31} & r_{32} & r_{33} & r_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Donde 12 elementos son funciones trigonométricas de seis variables de articulación desconocidas. Sin embargo, desde la submatriz superior de 3x3, es una matriz de rotación, sólo tres elementos de ellos son independientes. Esto es porque la condición de ortogonalidad. Por tanto seis de las doce ecuaciones son independientes.

Las funciones trigonométricas inherentemente proveen varias soluciones. Por consiguiente, de las múltiples configuraciones del robot se espera que las seis ecuaciones se resuelvan para las variables desconocidas.

Esto es posible para desacoplar el problema de cinemática inversa en dos subproblemas, conocida como posición de cinemática inversa y orientación de cinemática inversa. La consecuencia práctica de cada uno de los desacoplamientos es la asignación para romper el problema en dos problemas independientes, cada uno con tres parámetros desconocidos. Siguiendo el desacoplamiento principal, la transformación de la matriz global de un robot puede ser descompuesta a la translación y a la rotación.

Ecuación 17

$$\begin{aligned}
 {}^0T_6 &= {}^0D_6 {}^0R_6 \\
 {}^0T_6 &= \begin{bmatrix} I & {}^0d_6 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} {}^0R_6 & 0 \\ 0 & 1 \end{bmatrix} \\
 {}^0T_6 &= \begin{bmatrix} {}^0R_6 & {}^0d_6 \\ 0 & 1 \end{bmatrix}
 \end{aligned}$$

La matriz de translación 0D_6 puede ser resuelta mediante las variables de posición de la muñeca, y la matriz de rotación de 0R_6 puede ser resuelta mediante las variables de orientación.

3.5.2 Técnica de transformación inversa

Asumiendo que se tiene la matriz de transformación 0T_6 indicando la posición y la orientación global del efector final de un robot de seis grados de libertad con un sistema de coordenadas base B_0 . Además, se asume que la geometría y las matrices de transformación ${}^0T_1(q_1), {}^1T_2(q_2), {}^2T_3(q_3), {}^3T_4(q_4), {}^4T_5(q_5), {}^5T_6(q_6)$ están dadas en función de las variables de articulación.

De acuerdo con la cinemática directa,

$${}^0T_6 = {}^0T_1 {}^1T_2 {}^2T_3 {}^3T_4 {}^4T_5 {}^5T_6$$

$${}^0T_6 = \begin{bmatrix} r_{11} & r_{12} & r_{13} & r_{14} \\ r_{21} & r_{22} & r_{23} & r_{24} \\ r_{31} & r_{32} & r_{33} & r_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Se puede resolver el problema de cinemática inversa, mediante la resolución de las siguientes ecuaciones de las variables de articulación desconocidas:

$${}^1T_6 = {}^0T_1^{-1} {}^0T_6$$

$${}^2T_6 = {}^1T_2^{-1} {}^1T_1^{-1} {}^0T_6$$

$${}^3T_6 = {}^2T_3^{-1} {}^1T_2^{-1} {}^0T_1^{-1} {}^0T_6$$

$${}^4T_6 = {}^3T_4^{-1} {}^2T_3^{-1} {}^1T_2^{-1} {}^0T_1^{-1} {}^0T_6$$

$${}^5T_6 = {}^4T_5^{-1} {}^3T_4^{-1} {}^2T_3^{-1} {}^1T_2^{-1} {}^0T_1^{-1} {}^0T_6$$

$$I = {}^5T_6^{-1} {}^4T_5^{-1} {}^3T_4^{-1} {}^2T_3^{-1} {}^1T_2^{-1} {}^0T_1^{-1} {}^0T_6$$

3.5.3 Técnica iterativa

El problema de la cinemática inversa puede ser interpretado como la búsqueda de la solución q_k como el conjunto de las ecuaciones algebraicas no lineales.

$${}^0T_n = T(q)$$

$${}^0T_n = {}^0T_1(q_1) {}^1T_2(q_2) {}^2T_3(q_3) {}^3T_4(q_4) \dots {}^{n-1}T_n(q_n)$$

$${}^0T_n = \begin{bmatrix} r_{11} & r_{12} & r_{13} & r_{14} \\ r_{21} & r_{22} & r_{23} & r_{24} \\ r_{31} & r_{32} & r_{33} & r_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

O

$$r_{ij} = r_{ij}(q_k)$$
$$k = 1, 2, 3, 4, \dots n.$$

Donde n es el número de los grados de libertad. Sin embargo, máximo $n=6$ de las doce ecuaciones de 0T_n son independientes y pueden ser utilizadas para resolver las variables de articulación q_k . Las funciones $T(q)$ son trascendentales, que se dan de forma explícita basadas en el análisis de cinemática directa.

Numerosos métodos son aceptados para encontrar el número de ceros de la anterior ecuación, sin embargo, los métodos son en general iterativos. El método más común es conocido como el método de Newton – Raphson.

En el método iterativo, para resolver la ecuación cinemática

Ecuación 18

$$T(q) = 0$$

Para variables q , se empieza con valores iniciales

Ecuación 19

$$q^* = q + \delta q$$

Para variables de articulación. Usando la cinemática directa, se puede determinar la configuración del sistema del efector final y encontrar la variable de articulación.

Ecuación 20

$$T^* = T(q^*)$$

La diferencia entre la configuración calculada con la cinemática directa y el valor deseado representa un error, y es llamado residuo, el cuál puede ser minimizado.

Ecuación 21

$$\delta T = T - T^*$$

Un primer orden de la expansión de Taylor¹² de un set de ecuaciones es

Ecuación 22

$$T = T(q^* + \delta q)$$
$$T = T(q^*) + \frac{\partial T}{\partial q} \delta q + O(\delta q^2)$$

Asumiendo que $\delta q \ll I$ permite trabajar con un conjunto de ecuaciones lineales

Ecuación 23

$$\delta T = J \delta q$$

Donde J es la matriz Jacobiana del conjunto de ecuaciones

Ecuación 24

$$J(q) = \left[\frac{\partial T_i}{\partial q_j} \right]$$

Lo que implica

$$\delta q = J^{-1} \delta T$$

Por lo tanto las variables desconocidas de q son:

$$q = q^* + J^{-1} \delta T$$

Se deben utilizar los valores de la anterior ecuación como una nueva aproximación para repetir los cálculos y encontrar los nuevos valores. Repitiendo el cálculo se puede hacer una sumatoria en la siguiente ecuación iterativa para converger al valor exacto de las variables.

Ecuación 25

$$q^{(i+1)} = q^{(i)} + J^{-1}(q^{(i)}) \delta T(q^{(i)})$$

¹² Serie de Taylor definida en Cálculo conceptos y contextos James Stewart . 4ta edición, editorial Thompson

La técnica de iteración puede ser un conjunto en un algoritmo de fácil cálculo numérico.

1. Encender un contador inicial $i = 0$
2. Encontrar o dar un estimado inicial de $q^{(0)}$.
3. Calcular el residuo de $\delta T(q^{(i)}) = J(q^{(i)})\delta q^{(i)}$

Si todos los elementos de $T(q^{(i)})$ o de su norma $\|T(q^{(i)})\|$ son menores que la tolerancia, $\|T(q^{(i)})\| \ll \epsilon$ entonces termina la iteración. El $q^{(i)}$ es la solución deseada.

4. Calcular $q^{(i+1)} = q^{(i)} + J^{-1}(q^{(i)})\delta T(q^{(i)})$.
5. hacer $i = i + 1$ y retornar al paso 3.

La tolerancia ϵ puede ser el equivalente establecido en las variables

Ecuación 26

$$q^{(i+1)} - q^{(i)} < \epsilon$$

O en Jacobiano¹³

Ecuación 27

$$J - I < \epsilon$$

¹³ Se llama Jacobiano en honor al matemático Alemán Carl Gustav Jacob Jacobi (1804-1851)
Cálculo de Conceptos y Contextos James Stewart , 4ta edición, editorial Thompson

4 METODOLOGÍA

Con el fin de verificar la precisión de las posiciones, en la simulación de movimientos y en el modelo cinemático a implementar, se tomarán distintas posiciones en lugares diferentes del espacio de trabajo; usando el cálculo de cinemática inversa implementado, se obtienen los ángulos para llegar, en teoría, a esa posición. Se mueve el robot por esos valores angulares, desde la posición de origen, y se lee la posición que el robot alcanza. Este valor se compara con el inicial, indicando la precisión del modelo automático. Para estas comparaciones se utilizará la licencia de software comercial Ciros Robotics del que dispone la universidad, en gran parte debido a que sus resultados han sido comprobados directamente con el Robot Mitsubishi, hasta el punto que este es utilizado para la programación del mismo.

Realizando pruebas, se busca verificar la precisión de las posiciones en la simulación de los movimientos y el modelo cinemático utilizado para el cálculo.

Se realiza el cálculo de la cinemática inversa para cada posición, obteniendo los valores angulares necesarios para llegar a ella. A partir de estos valores, se utiliza el simulador comercial, y se lee la posición final, para compararla con la esperada.

Finalmente se estudiarán los elementos que se requieren para el desarrollo de una interfaz que permita la fácil interacción de cualquier interesado y observar los resultados del proceso sin necesidad de tener que realizar el proceso desde cero.

5 ALCANCES Y LIMITACIONES

Los movimientos de robots cada vez con mayor número de grados de libertad han venido evolucionando con el paso del tiempo, gracias a ello, se han implementado en la industria desde hace algunos años. Para iniciar el estudio del robot Mitsubishi Movemaster RV-M1 se analizan las características físicas específicas que se deben tener en cuenta para ciertos aspectos en el desarrollo de una simulación, tales como las especificaciones que se muestran en la tabla (1).

Tabla 1 Espacio de operación y dimensiones del robot Mitsubishi Movemaster RV-M1

ESPACIO DE OPERACIÓN	
Estructura mecánica	Rango
Cintura	300°
Hombro	130°
Codo	110°
Muñeca con palma de la mano	180°
Rodillo de la muñeca	360°
DIMENSIONES	
Base – Cintura	148 mm
Cintura – Hombro	152 mm
Hombro – Codo	250 mm
Codo – Muñeca	160 mm
Muñeca – Palma de la mano	72 mm
Dedos ()	75 mm

De acuerdo a las características del modelo, este software se implementa mediante la herramienta de programación MATLAB que es de vital importancia debido a sus características básicas de programación tales como la manipulación de matrices, la representación de datos y funciones, la implementación de algoritmos, la creación de interfaces de usuario (GUI) y la comunicación con programas en otros lenguajes y con otros dispositivos hardware. Adicionalmente la universidad cuenta con licencias Académicas que permiten el desarrollo de trabajos de este tipo en etapas iniciales.

El paquete MATLAB dispone de dos herramientas adicionales que expanden sus prestaciones, aunque no todas estas prestaciones sean útiles para el desarrollo del software, teniendo en cuenta que las librerías para hacer simulaciones CAD o en 3D no se encuentran en la licencia básica y por el contrario se encuentran en librerías adicionales que no están disponibles, sin embargo la recolección de información también es un importante aspecto, ya que los estudios realizados sobre este tipo de proyectos abarcan información que no es indispensable para el

desarrollo del mismo y se debe hacer una selección adecuada para desarrollar de manera precisa el algoritmo.

El proyecto pretende llegar de una forma sencilla al usuario no experimentado, es por ello que la simulación del proceso debe representarse mediante una interfaz que recree el movimiento del robot Mitsubishi Movemaster RV-M1 con los parámetros de visualización necesarios para programar una secuencia de funciones definiendo las posiciones finales del mismo con sus respectivos valores en las articulaciones.

La representación de secuencias en un proceso son de gran utilidad ya que permite una proyección del proceso a seguir y es por ello que la interfaz permite observar en tiempo real las condiciones de trabajo del manipulador sin dejar a un lado las restricciones físicas del mismo. Sin embargo, tampoco se puede dejar a un lado que la comunicación serial entre el PC y robot, están limitadas a la configuración electrónica del mismo, dependiendo en este caso del sistema de control implementado al manipulador.

Se espera que trabajos de investigación futuros permitan tomar los desarrollos que se mostrarán más adelante y permitan trasladarlos a software de licencia libre para de esta forma hacer una publicación y aplicación a niveles industriales.

6 DESARROLLO INGENIERIL Y ANÁLISIS DE RESULTADOS

El desarrollo de los resultados se presenta de acuerdo al análisis y a las descripciones físicas exactas del robot Mitsubishi Movemaster RV-M1 como manipulador. Para ellos se vale de la información exacta del producto para hacer la implementación adecuada en los cálculos cinemáticos de este sistema.



Figura 25 Fotografía Robot Mitsubishi Movemaster RV-M1

6.1 Características del robot Mitsubishi Movemaster RV-M1

- Control alámbrico
- Unidad de manejo
- Brazo electromecánico

El movimiento en los ejes permite ser controlado en 5 diferentes ejes que se muestran en la siguiente figura:

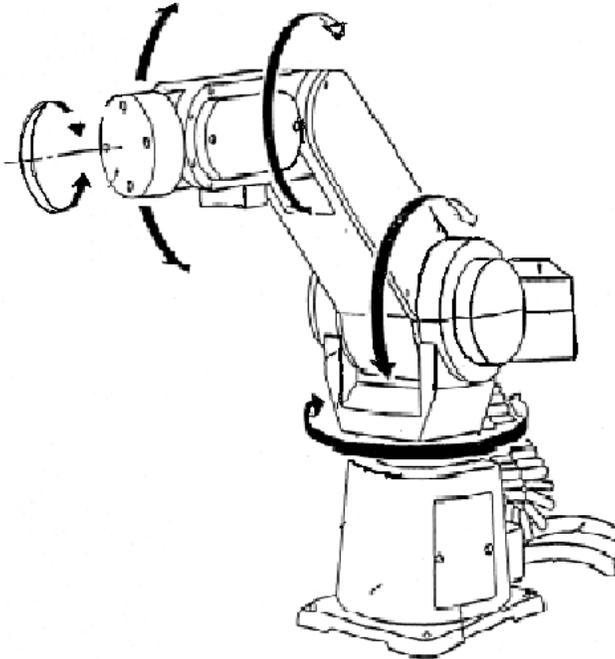


Figura 26 Ejes de las articulaciones del robot Mitsubishi Movemaster RV-M1

El movemaster RV-M1 requiere de un espacio de operación suficiente para realizar todos sus movimientos. Por lo tanto, es necesario que este espacio se encuentre libre de obstáculos para el buen funcionamiento del robot. A continuación se muestra un mapa de los movimientos permitidos dentro del espacio de operación del robot movemaster RV-M1.

El Movemaster RV-M1 es un robot de ensamble para piezas mecánicas, el cual consiste en un brazo asemejanza del brazo humano, como se puede observar. Se consideran cinco ejes (J1,J2,J3,J4,J5), aunque en realidad sólo son cuatro, ya que el eje J5 y J4 son uno mismo, pero se diferencian por los movimientos que éstos pueden realizar. Se les nombra de diferente manera con la finalidad de simplificar al usuario su identificación.

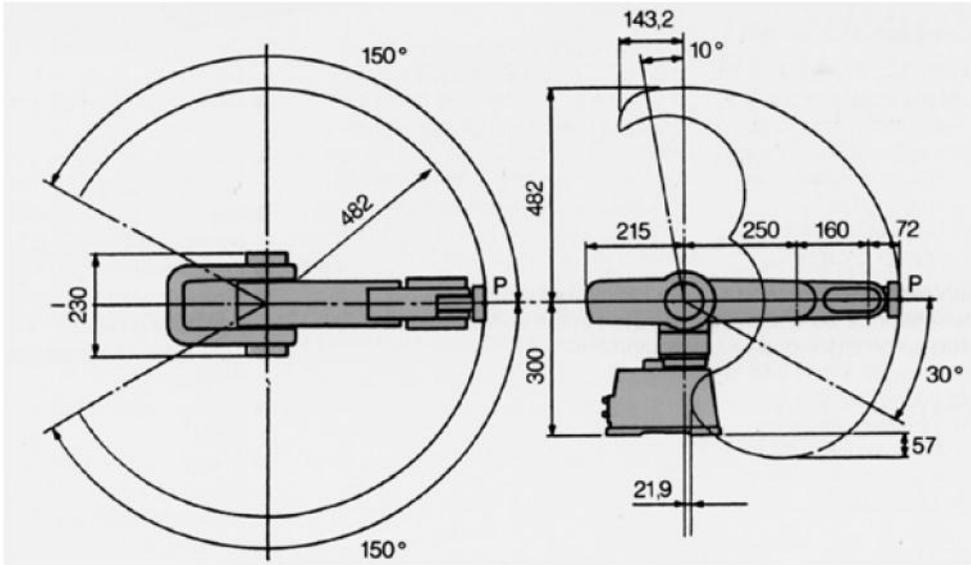


Figura 27 Dimensiones y límites del movimiento articular del robot Mitsubishi Movemaster RV-M1.

En la siguiente figura se muestran los movimientos y la nomenclatura utilizada para cada uno de sus ejes, así como la vista externa del robot sin utilizar una herramienta específica.

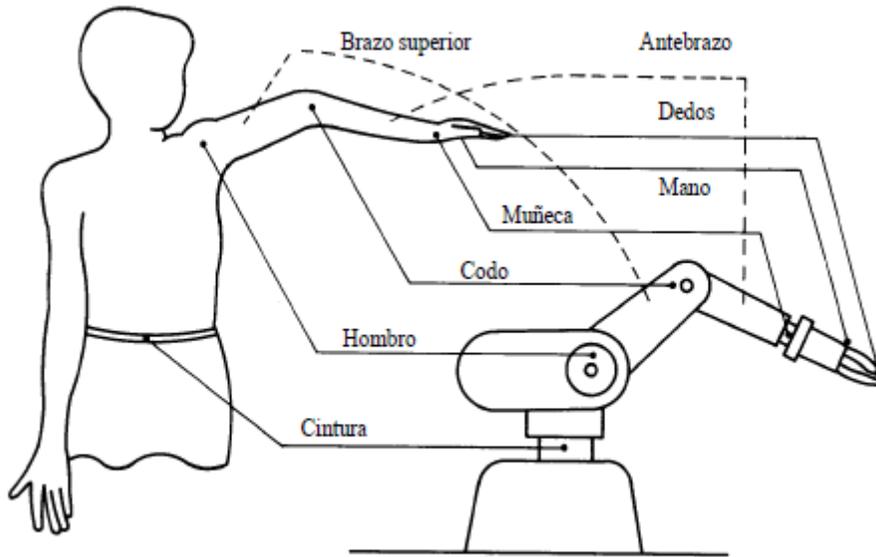


Figura 28 Relación movimientos del robot y nomenclatura

6.1.1 Dimensiones

Las dimensiones del manipulador robot se presentan a continuación:

Tabla 2 Dimensiones del robot Mitsubishi Movemaster RV-M1

DIMENSIONES	
Base – Cintura	148 mm
Cintura – Hombro	152 mm
Hombro – Codo	250 mm
Codo – Muñeca	160 mm
Muñeca – Palma de la mano	72 mm
Dedos ()	75 mm

En la figura siguiente se muestra la operación del robot en el sistema de coordenadas cartesianas. El punto central de la herramienta se mueve siguiendo el sistema de coordenadas cartesianas, dejando dos ejes constantes de cualquiera de los tres del sistema cartesiano. Variando el punto central de la herramienta (en el eje seleccionado como variable) por medio de una interpolación lineal de los ejes coordenados.

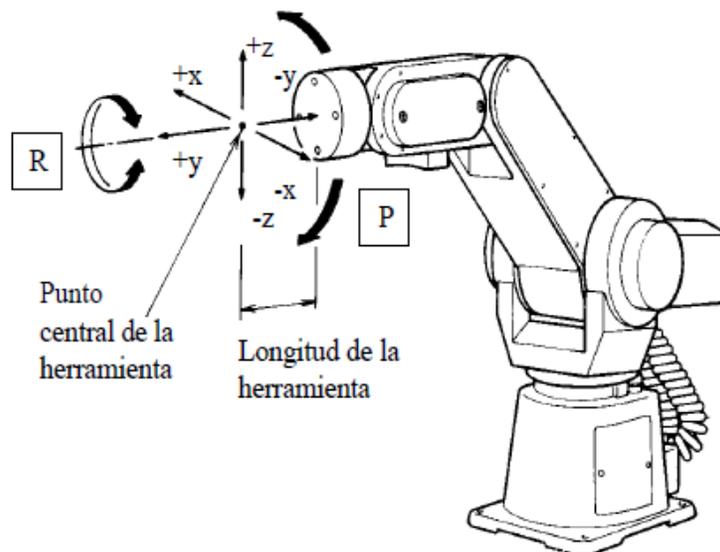


Figura 29 Movimientos del robot en coordenadas cartesianas

Luego de analizar el robot físicamente se procede a desarrollar los cálculos necesarios y propuestos en capítulos anteriores, que desarrollan el movimiento de cinemática directa e inversa en un sistema de coordenadas adecuado para ser representados gráficamente mediante un algoritmo en la herramienta computacional MATLAB.

El desarrollo del algoritmo de cinemática directa que se implementa en MATLAB, es estudiado y analizado de acuerdo al método de Denavit-Hartenberg de acuerdo a sus respectivos parámetros.

6.2 Método Denavit-Hartenberg

El robot Mitsubishi Movemaster RV-M1 es un manipulador articulado de cinco grados de libertad, tal como se observa en la figura. Sus parámetros DH (Denavit-Hartenberg) se establecen, a partir de la representación del robot, tanto en términos de los sistemas de referencia asociados a cada una de sus articulaciones, como de la longitud de los cuerpos rígidos que lo conforman.

En la siguiente ilustración, se representan los eslabones del robot Mitsubishi Movemaster RV-M1, enumerados desde la base hacia el efector final o herramienta como se muestra a continuación:

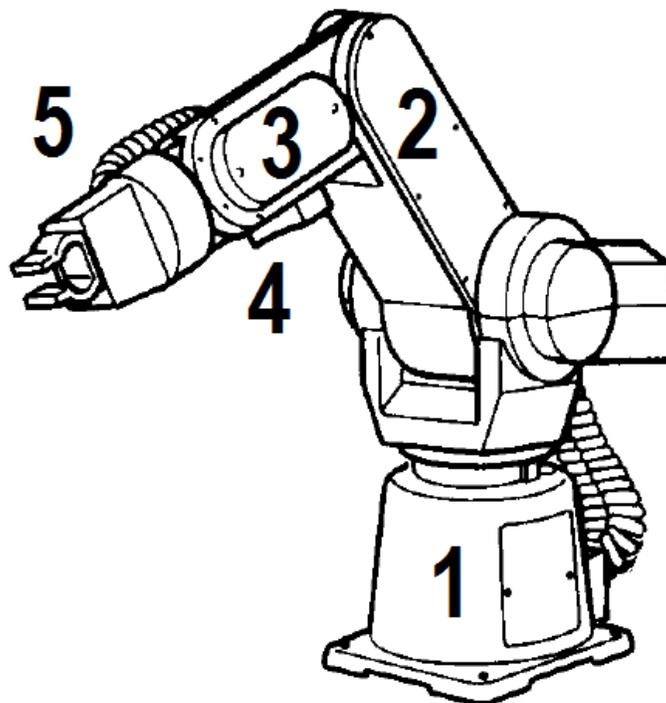


Figura 30 Nomenclatura de los eslabones del robot

De esta manera se puede hacer un análisis directo a la geometría del robot, clasificándolo según las uniones entre las diferentes articulaciones que lo componen y los eslabones del mismo, teniendo en cuenta que todas las articulaciones del manipulador son de revolución (R).

Tabla 3 Clasificación geométrica del robot Mitsubishi Movemaster RV-M1

Clasificación geométrica del manipulador	
Nº DE ESLABÓN	TIPO
1	R⊥R
2	R R
3	R R
4	R⊥R
5	R R

De acuerdo a la anterior información, se representa la tabla del método de Denavit-Hartenberg con notación y marco de coordenadas identificado con sus cuatro parámetros.

Tabla 4 Parámetros DH para el manipulador Mitsubishi RV-M1

ELEMENTO	a_i	α_i	d_i	θ_i
1	0	90	300	θ_1
2	250	0	0	θ_2
3	160	0	0	θ_3
4	0	-90	0	θ_4
5	0	0	187	θ_5

Estos valores describen cada elemento articulado, correspondiendo a las medidas indicadas presentadas anteriormente en la tabla (2) para el robot Mitsubishi Movemaster RV-M1 dadas en milímetros y centígrados respectivamente.

6.3 Matrices de transformación

De acuerdo a la notación Denavit-Hartenberg se pueden interpretar las matrices de acuerdo a la siguiente ecuación, para determinar la cinemática directa del manipulador.

$${}^0T_5 = {}^0T_1 {}^1T_2 {}^2T_3 {}^3T_4 {}^4T_5$$

Donde,

$${}^0T_1 = \begin{bmatrix} \cos \theta_1 & 0 & \sin \theta_1 & 0 \\ \sin \theta_1 & 0 & -\cos \theta_1 & 0 \\ 0 & 1 & 0 & 300 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$${}^1T_2 = \begin{bmatrix} \cos \theta_2 & -\sin \theta_2 & 0 & 250 * \cos \theta_2 \\ \sin \theta_2 & \cos \theta_2 & 0 & 250 * \sin \theta_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$${}^2T_3 = \begin{bmatrix} \cos \theta_3 & -\sin \theta_3 & 0 & 160 * \cos \theta_3 \\ \sin \theta_3 & \cos \theta_3 & 0 & 160 * \sin \theta_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$${}^3T_4 = \begin{bmatrix} \cos \theta_4 & 0 & \sin \theta_4 & 0 \\ \sin \theta_4 & 0 & -\cos \theta_4 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$${}^4T_5 = \begin{bmatrix} 1 & 0 & 0 & 179 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

6.4 Cinemática Directa

El desarrollo de la multiplicación de las anteriores matrices de transformación está representado en la siguiente matriz de 4x4 denominada matriz de transformación 0T_5 , donde se definen cada uno de los términos de forma independiente y su nomenclatura se reduce para mayor comprensión por su extensión y simplificación.

$${}^0T_5 = \begin{bmatrix} r_{11} & r_{12} & r_{13} & r_{14} \\ r_{21} & r_{22} & r_{23} & r_{24} \\ r_{31} & r_{32} & r_{33} & r_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Sea

$C = \text{COSENO}$

$S = \text{SENO}$

$$r_{11} = -C\theta_4(C\theta_1S\theta_2S\theta_3 - C\theta_1C\theta_2C\theta_3) - S\theta_4(C\theta_1C\theta_2S\theta_3 + C\theta_1C\theta_3S\theta_2)$$

$$r_{12} = S\theta_1$$

$$r_{13} = C\theta_4(C\theta_1C\theta_2S\theta_3 + C\theta_1C\theta_3S\theta_2) - S\theta_4(C\theta_1S\theta_2S\theta_3 - C\theta_1C\theta_2C\theta_3)$$

$$r_{21} = C\theta_4(C\theta_2C\theta_3S\theta_1 - S\theta_1S\theta_2S\theta_3) - S\theta_4(C\theta_2S\theta_1S\theta_3 + C\theta_3S\theta_1S\theta_2)$$

$$r_{22} = -C\theta_1$$

$$r_{23} = C\theta_4(C\theta_2S\theta_1S\theta_3 + C\theta_3S\theta_1S\theta_2) + S\theta_4(C\theta_2C\theta_3S\theta_1 - S\theta_1S\theta_2S\theta_3)$$

$$r_{31} = C\theta_4(C\theta_2S\theta_3 + C\theta_3S\theta_2) + S\theta_4(C\theta_2C\theta_3 - S\theta_2S\theta_3)$$

$$r_{32} = 0$$

$$r_{23} = S\theta_4(C\theta_2S\theta_3 + C\theta_3S\theta_2) - C\theta_4(C\theta_2C\theta_3 - S\theta_2S\theta_3)$$

Para el desarrollo de los elementos de la cuarta columna, es decir lo que refiere a r_{14} , r_{24} y r_{34} , se utiliza la abreviación de nomenclatura de las medidas de acuerdo a la geometría física del robot:

Sea:

$$l_0 = 300mm$$

$$l_1 = 250mm$$

$$l_2 = 179mm$$

$$l_3 = 160mm$$

$$r_{14} = l_1 C\theta_1 C\theta_2 - l_2 C\theta_4 (C\theta_1 S\theta_2 S\theta_3 - C\theta_1 C\theta_2 C\theta_3) \\ - l_2 S\theta_4 (C\theta_1 C\theta_2 S\theta_3 + C\theta_1 C\theta_3 S\theta_2) - l_3 C\theta_1 S\theta_2 S\theta_3 + l_3 C\theta_1 C\theta_2 C\theta_3$$

$$r_{24} = l_1 C\theta_2 S\theta_1 + l_2 C\theta_4 (C\theta_2 C\theta_3 S\theta_1 - S\theta_1 S\theta_2 S\theta_3) \\ - l_2 S\theta_4 (C\theta_2 S\theta_1 S\theta_3 + C\theta_3 S\theta_1 S\theta_2) + l_3 C\theta_2 C\theta_3 S\theta_1 - l_3 S\theta_1 S\theta_2 S\theta_3$$

$$r_{34} = l_1 S\theta_2 + l_3 C\theta_2 S\theta_3 + l_3 C\theta_3 S\theta_2 + l_2 C\theta_4 (C\theta_2 S\theta_3 + C\theta_3 S\theta_2) \\ + l_2 S\theta_4 (C\theta_2 C\theta_3 - S\theta_2 S\theta_3) + l_0$$

Los anteriores tres elementos de la matriz global, son evaluados y multiplicados por un vector, para expresarlos como posición en el espacio y lograr obtener la posición final del efector final o herramienta. Lo anterior se realiza ubicando las posiciones de cada articulación, con respecto a la posición de la anterior articulación y al sistema de coordenadas principal en la base del manipulador.

Las siguientes expresiones indican la posición de cada articulación de acuerdo a las matrices de transformación.

POSICIÓN PRIMERA ARTICULACIÓN P_1

$$P_1 = {}^0T_1 * \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

$$P_1 = \begin{bmatrix} 0 \\ 0 \\ 300 \\ 1 \end{bmatrix}$$

POSICIÓN SEGUNDA ARTICULACIÓN P_2

$$P_2 = {}^0T_1 {}^1T_2 * \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

$$P_2 = \begin{bmatrix} l_1 C\theta_1 C\theta_2 \\ l_1 C\theta_2 S\theta_1 \\ l_1 S\theta_2 + 300 \\ 1 \end{bmatrix}$$

POSICIÓN TERCERA ARTICULACIÓN P_3

$$P_3 = {}^0T_1 {}^1T_2 {}^2T_3 * \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

$$P_3 = \begin{bmatrix} l_1 C\theta_1 C\theta_2 - l_3 C\theta_1 S\theta_2 S\theta_3 + l_3 C\theta_1 C\theta_2 C\theta_3 \\ l_1 C\theta_2 S\theta_1 + l_3 C\theta_2 C\theta_3 S\theta_1 - l_3 S\theta_1 S\theta_2 S\theta_3 \\ l_1 S\theta_2 + l_3 C\theta_2 S\theta_3 + l_3 C\theta_3 S\theta_2 + 300 \\ 1 \end{bmatrix}$$

POSICIÓN CUARTA ARTICULACIÓN P_4

$$P_4 = {}^0T_1 {}^1T_2 {}^2T_3 {}^3T_4 * \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

$$P_4 = \begin{bmatrix} l_1 C\theta_1 C\theta_2 - l_3 C\theta_1 S\theta_2 S\theta_3 + l_3 C\theta_1 C\theta_2 C\theta_3 \\ l_1 C\theta_2 S\theta_1 + l_3 C\theta_2 C\theta_3 S\theta_1 - l_3 S\theta_1 S\theta_2 S\theta_3 \\ l_1 S\theta_2 + l_3 C\theta_2 S\theta_3 + l_3 C\theta_3 S\theta_2 + 300 \\ 1 \end{bmatrix}$$

POSICIÓN QUINTA ARTICULACIÓN P_5

$$P_5 = {}^0T_1 {}^1T_2 {}^2T_3 {}^3T_4 {}^4T_5 * \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

Es la representada por los elementos de la matriz r_{14}, r_{24} y r_{34}

Para comprender mejor lo anterior, se presenta a continuación un ejemplo, teniendo en cuenta la anterior información

6.4.1 Ejemplo:

Sea el manipulador Robot Mitsubishi Movemaster RV-M1, se analizan las matrices de transformación de acuerdo a la tabla (4), determinar la posición del efector final con los siguientes valores de ángulo de cada articulación:

- Waist: -45°
- Shoulder: 20°
- Elbow: -30°
- Pitch: 40°
- Roll: 50°

$${}^0T_1 = \begin{bmatrix} \cos(45^\circ) & 0 & \sin(45^\circ) & 0 \\ \sin(45^\circ) & 0 & -\cos(45^\circ) & 0 \\ 0 & 1 & 0 & 300 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$${}^1T_2 = \begin{bmatrix} \cos(20^\circ) & -\sin(20^\circ) & 0 & 250 * \cos(20^\circ) \\ \sin(20^\circ) & \cos(20^\circ) & 0 & 250 * \sin(20^\circ) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$${}^2T_3 = \begin{bmatrix} \cos(-30^\circ) & -\sin(-30^\circ) & 0 & 160 * \cos(-30^\circ) \\ \sin(-30^\circ) & \cos(-30^\circ) & 0 & 160 * \sin(-30^\circ) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$${}^3T_4 = \begin{bmatrix} \cos(50^\circ) & 0 & \sin(50^\circ) & 0 \\ \sin(50^\circ) & 0 & -\cos(50^\circ) & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$${}^4T_5 = \begin{bmatrix} 1 & 0 & 0 & 179 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Desarrollando la multiplicación de las matrices y multiplicando por el vector para obtener los 4 elementos de la última columna, se obtiene:

$$P_5 = {}^0T_1 {}^1T_2 {}^2T_3 {}^3T_4 {}^4T_5 * \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

$$P_5 = \begin{bmatrix} 387.149 \\ -387.149 \\ 447.221 \\ 1 \end{bmatrix} = \begin{bmatrix} \text{Posición X} \\ \text{Posición Y} \\ \text{Posición Z} \\ 1 \end{bmatrix}$$

De esta manera se comprueba que con los ángulos establecidos inicialmente, se tienen las posiciones finales en x, y y z.

6.5 Cinemática Inversa

Para el desarrollo de la cinemática inversa del robot Mitsubishi Movemaster RV-M1 se continúa utilizando el método de las matrices de transformación de acuerdo a la nomenclatura Denavit-Hartenberg. El primer paso es separar los elementos r_{14}, r_{24} y r_{34} para encontrar los ángulos de cada articulación de acuerdo a una posición dada en el espacio.

$${}^0T_5 = \begin{bmatrix} r_{11} & r_{12} & r_{13} & r_{14} \\ r_{21} & r_{22} & r_{23} & r_{24} \\ r_{31} & r_{32} & r_{33} & r_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

A diferencia de la cinemática directa, se utilizan los términos r_{14}, r_{24} y r_{34} despreciando la distancia $l_2 = 0$, debido las descripciones físicas del robot. De acuerdo a lo anterior, se utilizan estos términos igualándose a las posiciones P_x, P_y y P_z respectivamente, de la siguiente manera:

$$r_{14} = l_1 C\theta_1 C\theta_2 - l_3 C\theta_1 S\theta_2 S\theta_3 + l_3 C\theta_1 C\theta_2 C\theta_3$$

$$r_{14} = l_1 C\theta_1 C\theta_2 - l_3 C\theta_1 (S\theta_2 S\theta_3 + C\theta_2 C\theta_3)$$

$$r_{24} = l_1 C\theta_2 S\theta_1 + l_3 C\theta_2 C\theta_3 S\theta_1 - l_3 S\theta_1 S\theta_2 S\theta_3$$

$$r_{24} = l_1 C\theta_2 S\theta_1 + l_3 S\theta_1 (C\theta_2 C\theta_3 - S\theta_2 S\theta_3)$$

$$r_{34} = l_1 S\theta_2 + l_3 C\theta_2 S\theta_3 + l_3 C\theta_3 S\theta_2 + l_0$$

$$r_{34} = l_1 S\theta_2 + l_3 (C\theta_2 S\theta_3 + C\theta_3 S\theta_2) + l_0$$

Donde,

$$\begin{bmatrix} l_1 C\theta_1 C\theta_2 - l_3 C\theta_1 (S\theta_2 S\theta_3 + C\theta_2 C\theta_3) \\ l_1 C\theta_2 S\theta_1 + l_3 S\theta_1 (S\theta_2 S\theta_3 + C\theta_2 C\theta_3) \\ l_1 S\theta_2 + l_3 (C\theta_2 S\theta_3 + C\theta_3 S\theta_2) + l_0 \end{bmatrix} = \begin{bmatrix} P_x \\ P_y \\ P_z \end{bmatrix}$$

De lo anterior se deduce que se tienen tres ecuaciones con tres incógnitas, y se desarrollan algebraicamente dejando los siguientes resultados:

$$\theta_1 = \tan^{-1} \left(\frac{P_y}{P_x} \right),$$

$$\theta_2 = 2ATAN \left[\frac{b \pm \sqrt{b^2 + a^2 - c^2}}{a + c} \right]$$

Donde,

$$a = 2l_1 \sqrt{P_x^2 + P_y^2}$$

$$b = 2l_1 (l_0 - P_z)$$

$$c = P_x^2 + P_y^2 + P_z^2 + l_0^2 + l_1^2 + l_3^2 - 2P_z l_0,$$

$$\theta_3 = 2 \operatorname{ATAN} \left[\frac{b \pm \sqrt{b^2 + a^2 - c^2}}{a + c} \right]$$

Donde,

$$a = 2l_1l_3 + 2l_0l_3S\theta_2$$

$$b = 2l_0l_3C\theta_2$$

$$c = P_x^2 + P_y^2 + P_z^2 - 2l_1l_0S\theta_2$$

Para determinar los ángulos θ_4 y θ_5 se analizan mediante la geometría física del robot de la siguiente manera:

$${}^3T_4 = \begin{bmatrix} \cos \theta_4 & 0 & \sin \theta_4 & 0 \\ \sin \theta_4 & 0 & -\cos \theta_4 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$${}^4T_5 = \begin{bmatrix} 1 & 0 & 0 & 179 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^3T_5 = {}^3T_4 {}^4T_5$$

$${}^3T_5 = \begin{bmatrix} s_{11} & s_{12} & s_{13} & s_{14} \\ s_{21} & s_{22} & s_{23} & s_{24} \\ s_{31} & s_{32} & s_{33} & s_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \cos \theta_4 & 0 & \sin \theta_4 & l_2 \cos \theta_4 \\ \sin \theta_4 & 0 & -\cos \theta_4 & l_2 \sin \theta_4 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Se toman los dos primeros términos de la cuarta columna de la anterior matriz de 4x4 y se procede de la siguiente manera:

Sea,

$$s_{14} = l_2 \cos \theta_4$$

$$s_{24} = l_2 \sin \theta_4$$

Se utiliza la misma expresión trigonométrica para obtener el resultado en términos de θ_4 .

$$\theta_4 = 2\text{ATAN} \left[\frac{s_{24}}{s_{14}} \right]$$

Una vez obtenidos los valores de los ángulos respectivamente, se maneja una estructura secuencial para desarrollar la cinemática inversa recalculando los valores de cada ángulo.

6.5.1 Ejemplo

Al igual que con la cinemática directa, se representa un ejemplo para comprobar los desarrollos matemáticos presentados anteriormente. Sea el manipulador Robot Mitsubishi Movemaster RV-M1, se analizan las matrices de transformación de acuerdo a la tabla (4), determinar la posición de cada uno de los ángulos de las articulaciones, teniendo en cuenta el ejemplo 6.4.1. de la sección anterior, para comprobar los datos partiendo de las posiciones x, y y z.

- Posición en X: 387.149
- Posición en Y: -387.149
- Posición en Z: 447.221

$$\theta_1 = \tan^{-1} \left(\frac{-387.149}{387.149} \right) = -45^\circ$$

$$\theta_2 = 2ATAN \left[\frac{b \pm \sqrt{b^2 + a^2 - c^2}}{a + c} \right]$$

Donde,

$$a = 2 * 250 \sqrt{(387.149)^2 + (-387.149)^2}$$

$$b = 2 * 250 * (300 - 447.221)$$

$$c = (387.149)^2 + (-387.149)^2 + (447.221)^2 + 300^2 + 250^2 + 160^2 - 2(447.221)(300)$$

$$\theta_2 = 20^\circ$$

$$\theta_3 = 2ATAN \left[\frac{b \pm \sqrt{b^2 + a^2 - c^2}}{a + c} \right]$$

Donde,

$$a = 2(250)(160) + 2(300)(160) \text{Sen}(20^\circ)$$

$$b = 2(300)(160) \text{Cos}(20^\circ)$$

$$c = (387.149)^2 + (-387.149)^2 + (447.221)^2 - 2(250)(300) \text{Sen}(20^\circ)$$

$$\theta_3 = -30^\circ$$

Una vez obtenidos los tres primeros ángulos, se procede a descubrir geoméricamente los dos ángulos restantes.

$${}^3T_5 = {}^3T_4 {}^4T_5$$

$${}^3T_5 = \begin{bmatrix} s_{11} & s_{12} & s_{13} & s_{14} \\ s_{21} & s_{22} & s_{23} & s_{24} \\ s_{31} & s_{32} & s_{33} & s_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\theta_4 = 2\text{ATAN} \left[\frac{s_{24}}{s_{14}} \right]$$

$$\theta_4 = 2\text{ATAN} \left[\frac{115.0590}{137.1220} \right]$$

$$\theta_4 = 40^\circ$$

$$\theta_4 = 50^\circ$$

De esta manera se comprueba que la cinemática está desarrollada en su forma inversa y en su forma directa. El ángulo θ_5 se representa directamente por la geometría, ya que por el tipo de movimiento no afecta la posición final.

El desarrollo de los anteriores cálculos matemáticos desarrollan los problemas cinemáticos asociados al Robot Mitsubishi Movemaster MV-R1, lo cual es de gran utilidad para presentar esta información a usuarios no experimentados que desean aprender y conocer más acerca de las posiciones del manipulador. Por esta razón se crea la necesidad de una interfaz capaz de mostrar simultáneamente los pasos y condiciones para programar el robot, en cualquiera de las dos cinemáticas propuestas, además del desarrollo de la visualización de tareas en secuencia para la visualización de los datos.

6.6 Creación de la Interfaz

El desarrollo de interfaz final se hizo mediante etapas, mejorando paso a paso la interfaz agregándole datos de acuerdo a los desarrollos matemáticos estudiados anteriormente:

- Se creó una interfaz amigable al usuario, de fácil manejo y desarrollada en GUIDE de Matlab. En primer lugar se hizo un espacio grafico para simular los movimientos del robot. También se crearon 5 cuadros para el ingreso de valores de ángulos para la cinemática directa, un botón 'CALCULAR' para desarrollar internamente la cinemática y por ultimo un icono estático en la parte superior izquierda que sirve para rotar la grafica.

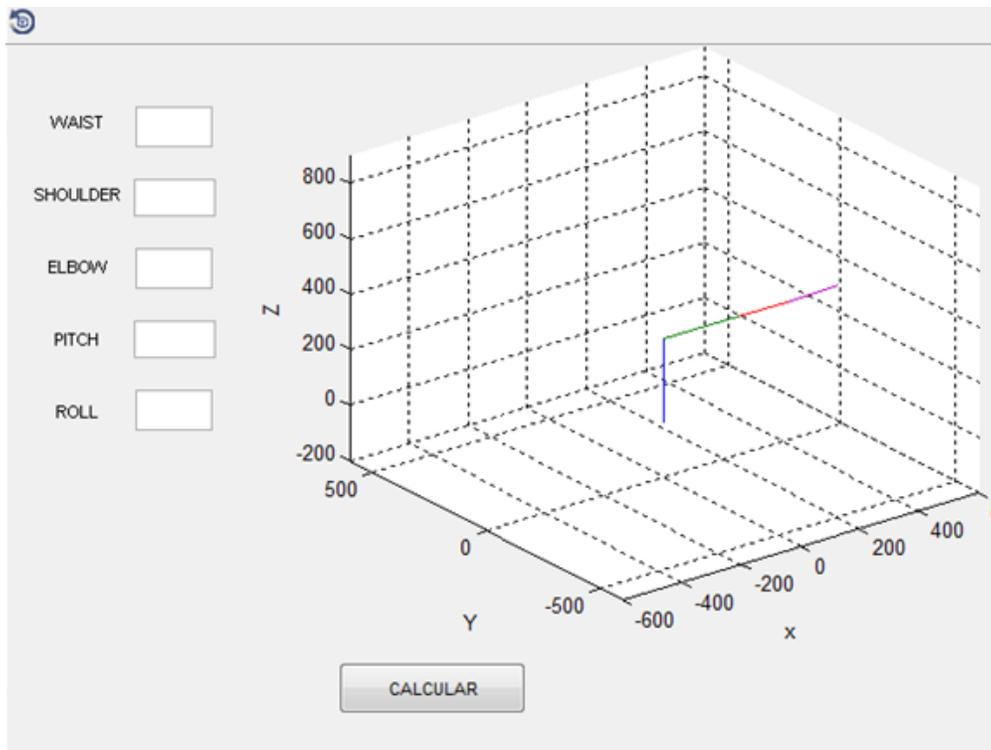


Figura 31 Primera implementación de la interfaz

En la anterior interfaz se podía ver la posición final del robot en una grafica donde se representan las articulaciones con líneas rectas, se podía rotar la grafica para ver mejor la posición del robot las desventajas de esta interfaz era que no se tenía el resultado numérico de la posición final, a su vez, tampoco se podía ver el recorrido que realizaba el robot.

- En la siguiente interfaz se realizó el titulo de la cinemática que se está trabajando y se crearon 5 cuadros para mostrar al usuario la salida de datos finales. En donde se encuentra la posición final de una 4 cuarta articulación que se interpreta como la suma de distancia de la 4 articulación y la 5 articulación debido a que el último ángulo no afecta la posición final; Finalmente se añadieron las restricciones de ángulo.

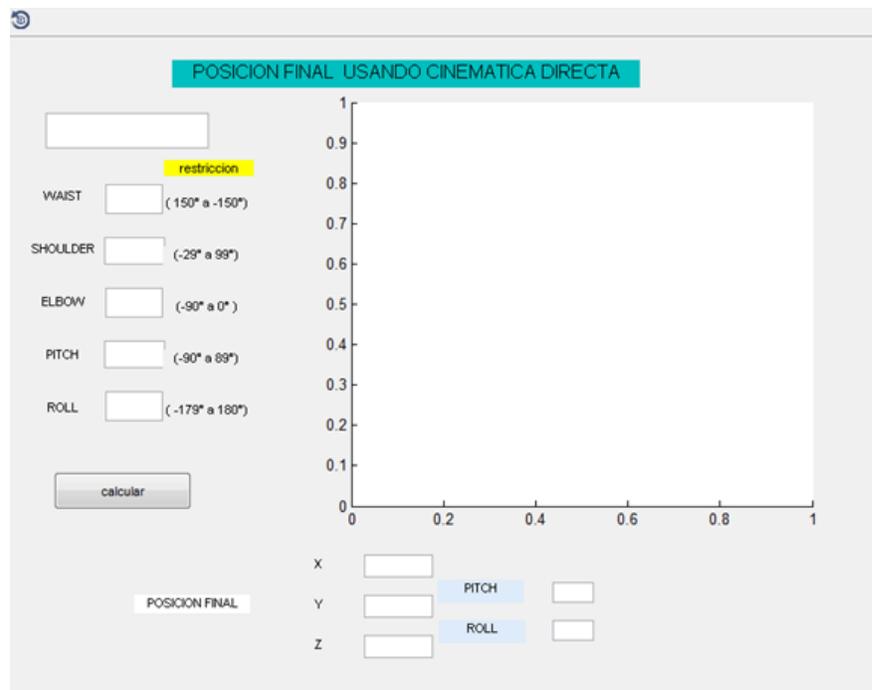


Figura 32 Segunda Implementación de la interfaz

En esta interfaz se incorporó el resultado numérico de la posición final en 5 cuadros, también se colocó las restricciones de ángulo de cada una de las articulaciones para que el programa tuviera las mismas limitantes de movimiento del robot.

Dentro de las desventajas de esta interfaz, se observó que no se podía ver el movimiento que se generaba para llegar a la posición final del robot y no tener como dejar la grafica en una posición inicial; por esta razón, el programa fallaba después de ingresar el primer dato de movimiento.

- En la siguiente interfaz se mejoró el título y en el resultado de la grafica se hizo un intervalo de 10 movimientos entre el ángulo inicial y el final. Cada uno de estos se grafica con un lapso de 0.1 segundos y se hizo un botón de 'borrar' para inicializar los valores iniciales y colocar la grafica en posición donde todos los ángulos tienen el valor de cero.

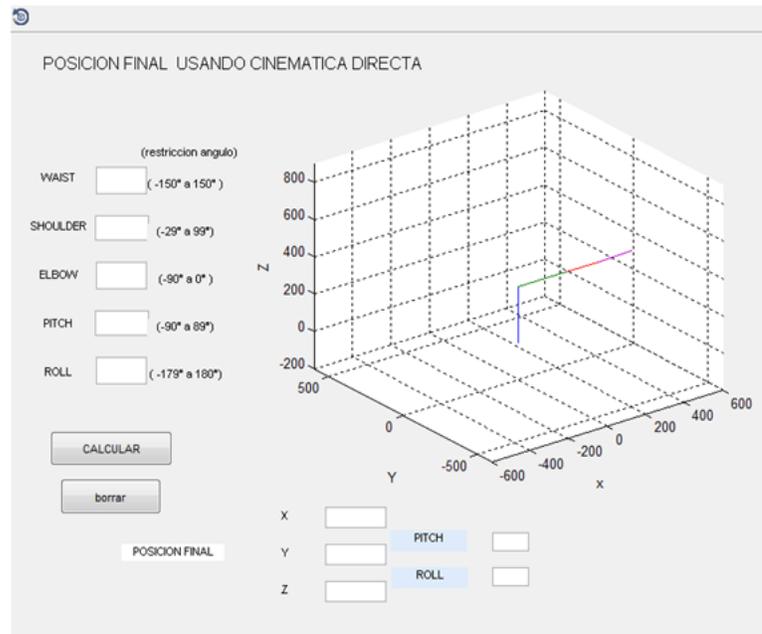


Figura 33 Tercera implementación de la interfaz

En esta interfaz se ordenaron los cuadros y diferentes botones para mejorar la presentación gráfica de la misma, se añadieron los límites fijos en la grafica para que esta no cambiara de tamaño cada vez que se ingresaba un nuevo movimiento y se hizo una secuencia de movimientos para que pareciera que el robot se estuviera moviendo hasta la posición final de una manera constante. Se incorporó un botón de borrado que devuelve el robot a su posición inicial, y se encontró que una de las desventajas era que no se podía re direccionar de una a otra cinemática.

- En la siguiente interfaz se hizo un menú despegable que re direcciona el programa entre la cinemática directa y la cinemática inversa, para que en el resultado de la gráfica, a partir del momento en que el usuario ingrese al software, se encuentren todos los ángulos en la posición inicial (todos los

ángulos en 0). Se le añade que cuando se ingresa el segundo re direccionamiento de ángulo, no comience el movimiento desde la posición inicial si no desde la posición final del primer direccionamiento ingresado en el programa. Se crearon mensajes de advertencia cuando el usuario ingresaba valores errados en el programa que advierten donde se produjo el error, que en cinemática inversa estos mensajes advierten cuando las coordenadas no se pueden hacer desarrollar mediante este medio cinemático.

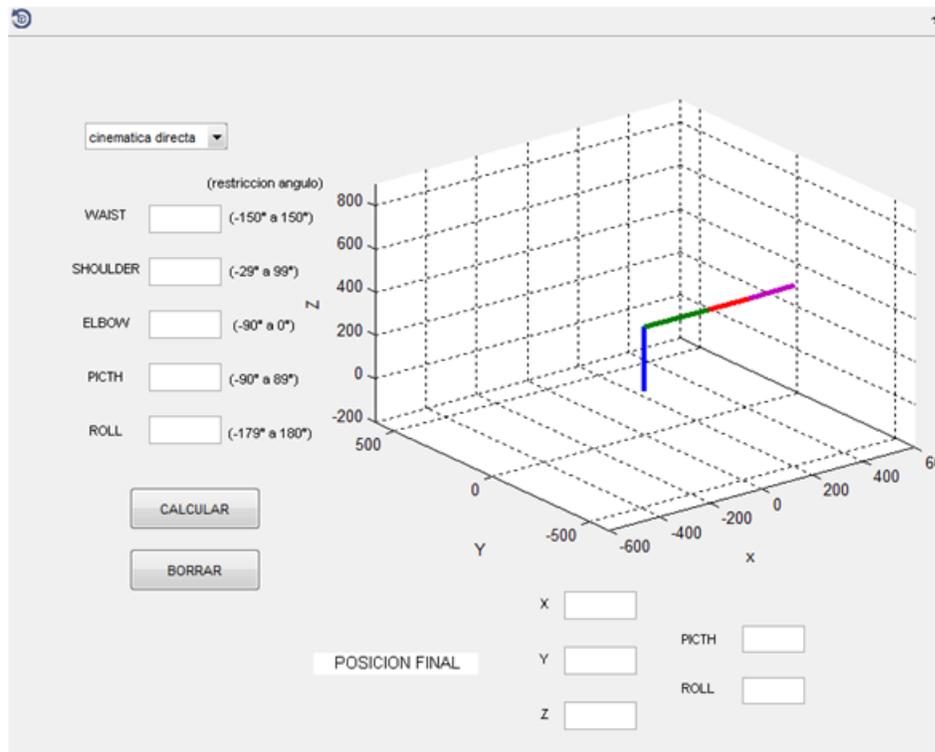


Figura 34 Cuarta implementación de la interfaz

En esta interfaz se mejoró la grafica presentada al usuario, se hizo un re direccionamiento desde un menú despegable para pasar de una a otra cinemática desde la misma interfaz, sin necesidad de activar otro programa en Matlab.

Se hizo posible pasar de un movimiento a otro sin necesidad de oprimir el botón de borrado que coloca el robot en posición inicial, pero como desventaja de esta interfaz se encontró que no se podían guardar las diferentes posiciones del robot en una memoria para realizar una secuencia con las posiciones.

- En esta interfaz final se hizo una tabla de datos para guardar datos de programación; en donde se pueden ejecutar hasta un máximo de 15 movimientos programados con un intervalo de 3 segundos para que se pueda observar el cambio de movimiento. En cada intervalo se muestra la posición final de cada movimiento, se crearon 2 botones 'PROGRAMAR' y 'BORRAR' en la parte inferior derecha que sirven para controlar la tabla de datos. Se le añaden las líneas gruesas en la grafica para comodidad del usuario, además por visibilidad y se reorganizo de una manera proporcional al espacio para que fueran agradables al usuario.

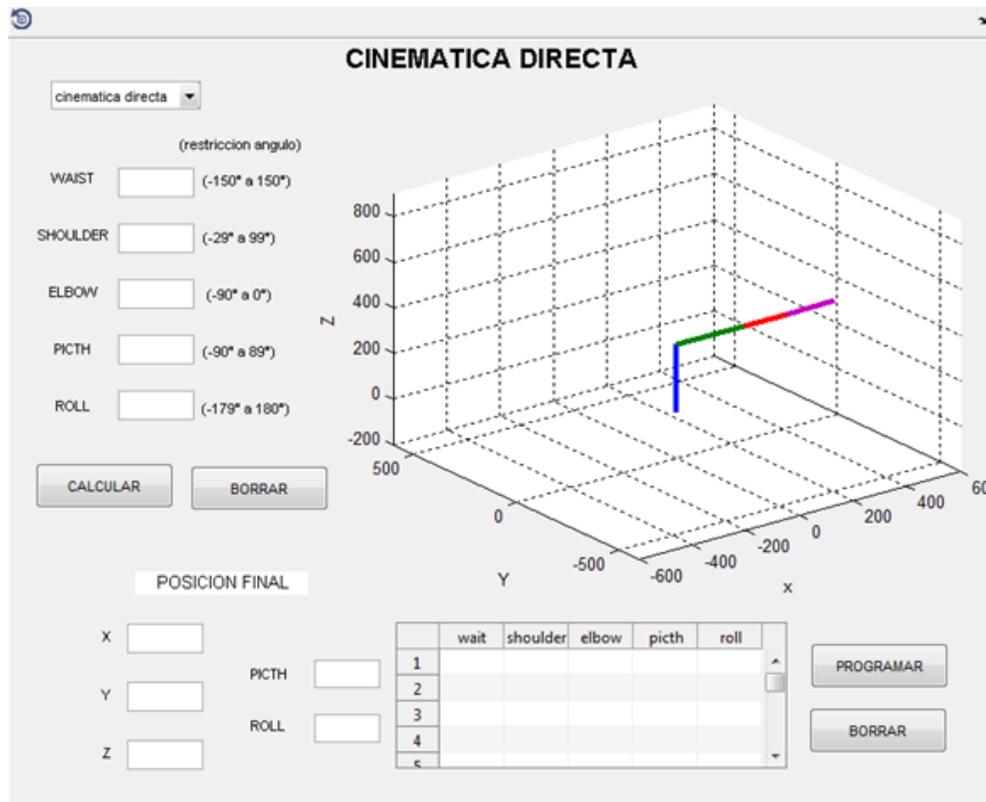


Figura 35 Desarrollo final de la interfaz

En esta interfaz final se logró tener una memoria para un máximo de 15 posiciones que permitiera simular una secuencia de movimientos como en un proceso real del robot, teniendo 2 botones de mando capaces programar la secuencia desde la misma tabla y que realiza los movimientos en un intervalo de 3 segundos entre un movimiento y otro. Con esta interfaz se logró tener en una misma vista los datos ingresados y la posición final, también se pudo interactuar entre los dos métodos de programación (cinemática inversa y cinemática directa) de una manera fácil y cómoda para el usuario no experimentado; Finalmente, se pudo comparar de manera satisfactoria los resultados con el programa Ciro's Festo Automation.

6.7 Comparación software CIROS Festo Automation con el software desarrollado en este proyecto.

Antes de realizar una comparación directa de los resultados con el software comercial, es necesario hacer la siguiente salvedad: el software CIROS Festo Automation tiene una diferencia en su representación grafica de los movimientos de robot. En el software CIROS, en dos de sus tres planos que tienen los ángulos positivos en el sentido contrario a las manecillas del reloj y en el plano donde se mueve el ángulo 'WAIST' los ángulos positivos están en sentido de las manecillas de reloj. El software presentado en este trabajo todos los planos tienen los ángulos positivos en sentido contrario de las manecillas del reloj.

A continuación se mostrará el resultado de 3 pruebas de cinemática directa y 3 pruebas de cinemática inversa comparando los 2 paquetes de software; los primeros son resultados del programa CIROS Festo Automation y a continuación los resultados del programa de este proyecto.

6.7.1 Cinemática directa

En la cinemática directa se ingresan los valores de los 5 ángulos de las articulaciones que se desean simular y como resultado de la posición final se tienen 3 coordenadas (x,y,z) y 2 valores de ángulos (pitch y roll) que indican la posición en el espacio.

CIROS Festo Automation

En este software lo primero que se realiza es desplegar el menú de 'Joint operation (rv-m1)' que aparece en la pantalla al presionar la tecla F8, en este menú se selecciona la opción 'Joint jog', que desplegara otra ventana llamada 'set Joint coordinates' en ella se colocan los valores de cada ángulo de cada articulación. A continuación se presiona el botón 'OK' para que muestre la posición final de la gráfica. Para obtener los valores numéricos de la posición final se selecciona el menú 'Joint operation (rv-m1)' la opción 'xyz jog' que desplegara una ventana llamada 'Set world coordinates' donde aparecen las 3 coordenadas y 2 valores de ángulos finales.

A continuación los valores ingresados en la ventana 'Set joint coordinates':

CIROS Festo Automation

- Waist: 0 shoulder:90 elbow:-90 pitch:0 roll:0

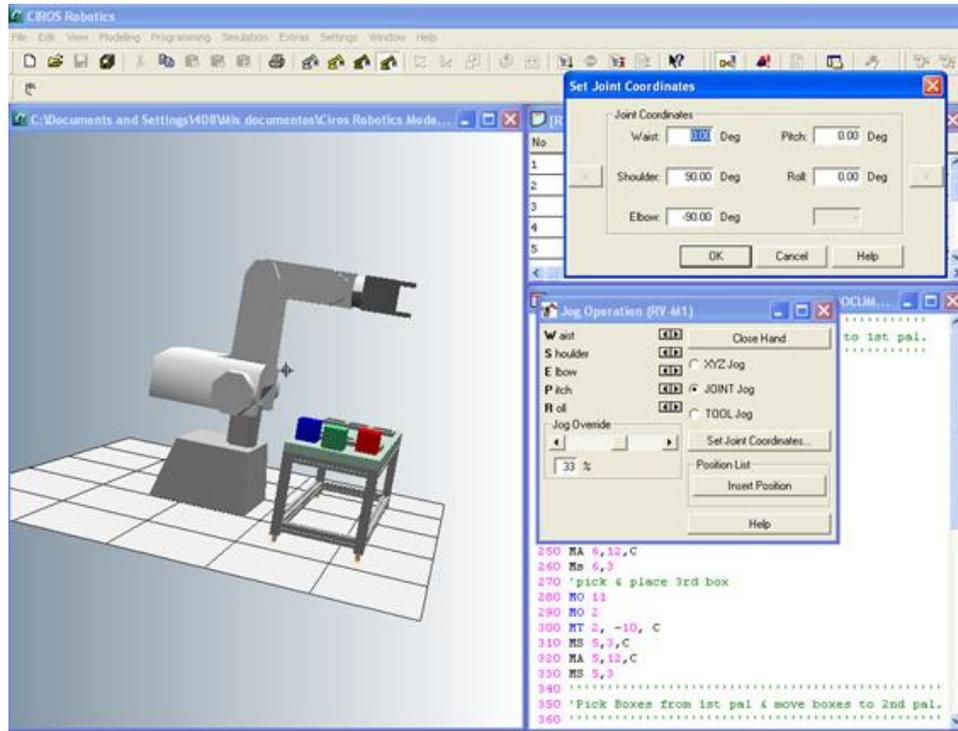


Figura 36 Posición del manipulador simulada en CIROS Festo Automation

Valores ingresados en la ventana 'set joint coordinates':



Figura 37 valores ingresados a CIROS Festo Automation

Resultado de la posición final en valores numericos:

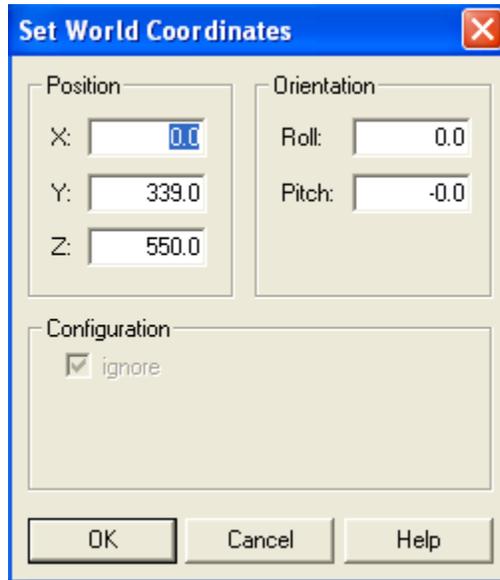


Figura 38 Resultado de las coordenadas finales en CIROS Festo Automation

Software Del Proyecto

En este software lo primero que se realiza es seleccionar en el menú desplegable la opción 'cinemática directa'. Una vez seleccionado se tendrá en la parte superior izquierda cinco cuadros para ingresar los ángulos de las articulaciones, una vez ingresado estos datos se oprime el botón 'calcular' y aparece en la grafica el movimiento del robot hasta la posición. Además de ello, en la parte inferior izquierda se tienen 5 cuadros donde aparecen las 3 coordenadas y 2 valores de ángulos finales.

A continuación los valores ingresados en la interfaz:

- Waist: 0 shoulder: 90 elbow: -90 picth: 0 roll: 0

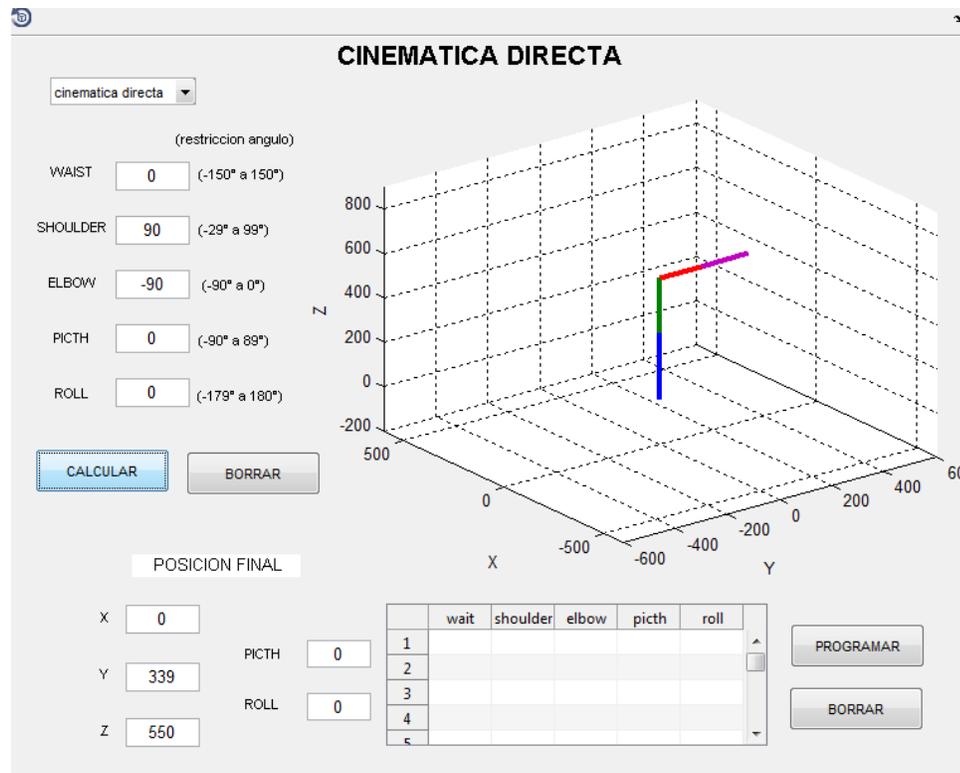


Figura 39 Posición del manipulador simulada en el software del proyecto

Valores ingresados en la interfaz:



Figura 40 valores ingresados a el software del proyecto

Resultado de la posición final en valores numericos:

POSICION FINAL			
X	0	PICTH	0
Y	339	ROLL	0
Z	550		

Figura 41 Resultado de las coordenadas finales en el software del proyecto

A continuación el segundo ejemplo de la comparación de la cinemática directa:

CIROS Festo Automation

Valores ingresados en la ventana 'set joint coordinates':

- Waist: 150 shoulder: 60 elbow: -45 picth: 60 roll: 120

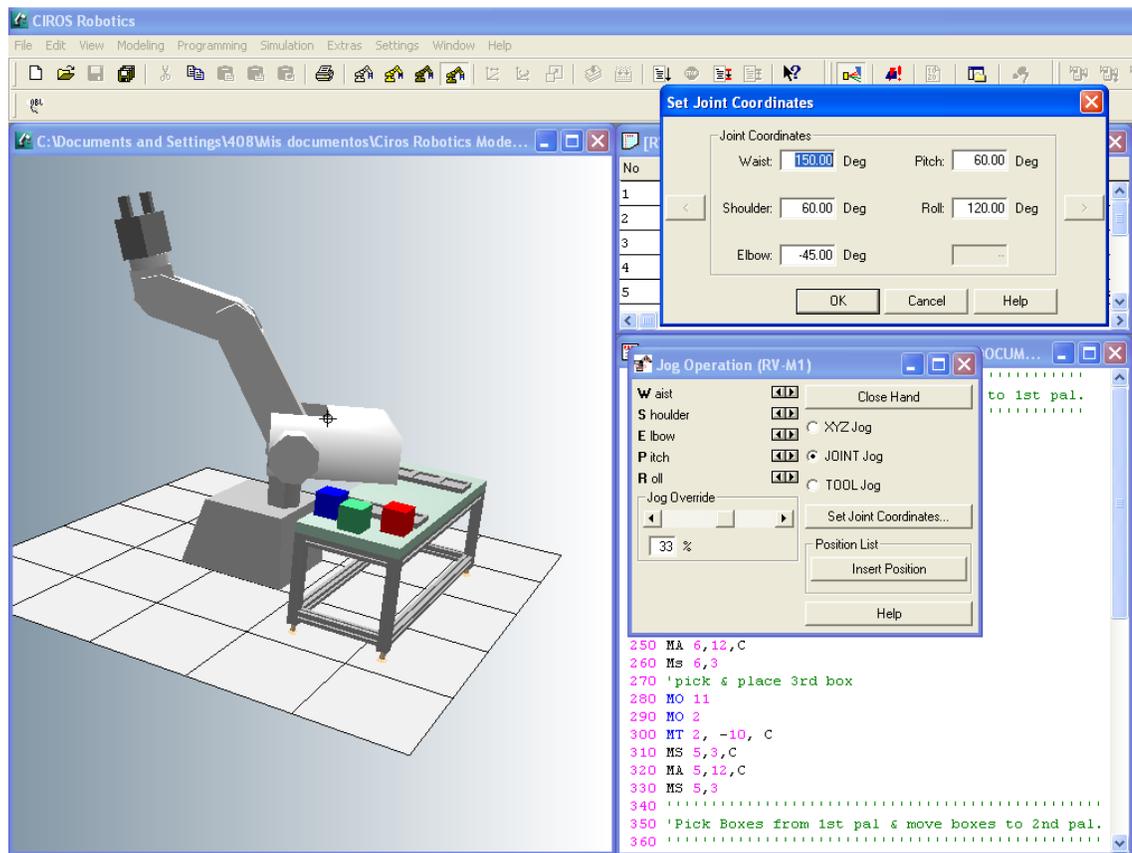


Figura 42 Posición del manipulador simulada en CIROS Festo Automation

Valores ingresados en la ventana 'set joint coordinates':



Figura 43 Valores ingresados a CIROS Festo Automation

Resultado de la posición final en valores numericos:

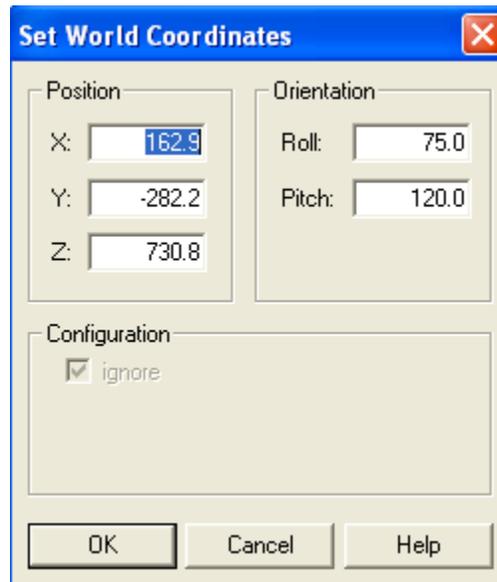


Figura 44 Resultado de las coordenadas finales en CIROS Festo Automation

Software Del Proyecto

Valores ingresados en la interfaz:

- Waist: 150 shoulder: 60 elbow: -45 picth: 60 roll: 120

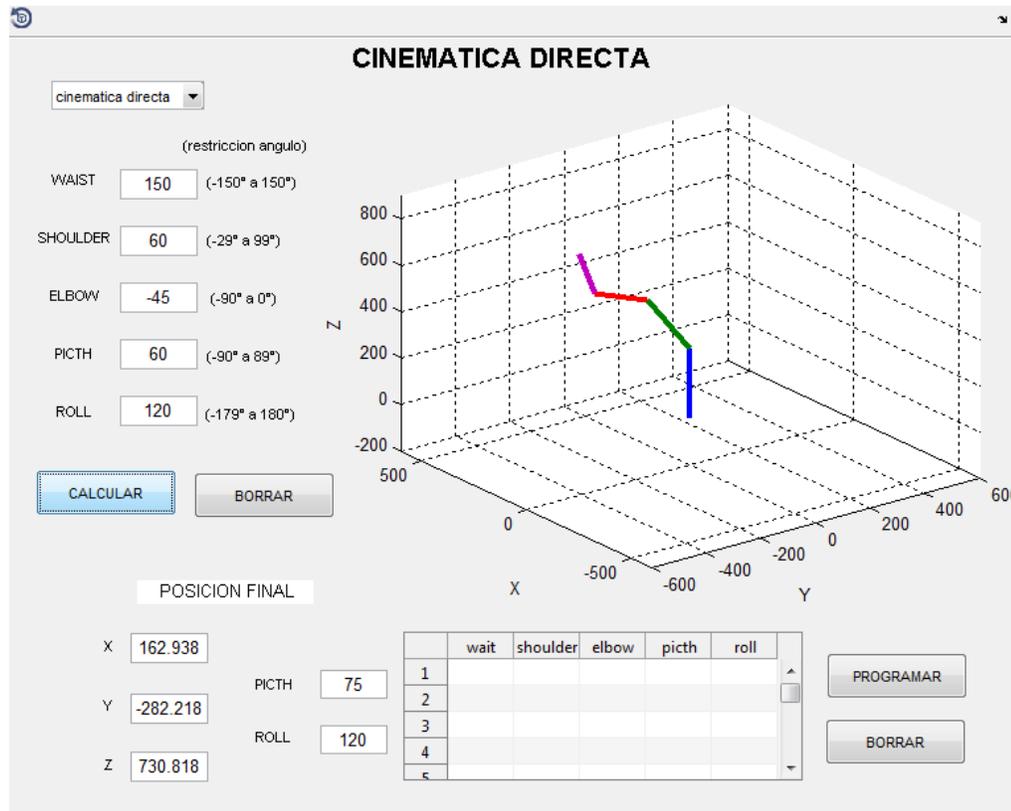


Figura 45 Posición del manipulador simulada en el software del proyecto

Valores ingresados en la interfaz:

(restriccion angulo)

WAIST (-150° a 150°)

SHOULDER (-29° a 99°)

ELBOW (-90° a 0°)

PICTH (-90° a 89°)

ROLL (-179° a 180°)

Figura 46 Valores ingresados a el software del proyecto

Resultado de la posición final en valores numericos:

POSICION FINAL	
X	162.938
Y	-282.218
Z	730.818
PICHT	75
ROLL	120

Figura 47 Resultado de las coordenadas finales en el software del proyecto

A continuación el Tercer ejemplo de la comparación de la cinemática directa:

CIROS Festo Automation

Valores ingresado en la ventana 'set joint coordinates':

- Waist: -75 shoulder: 34 elbow: -30 picth: 45 roll: 34

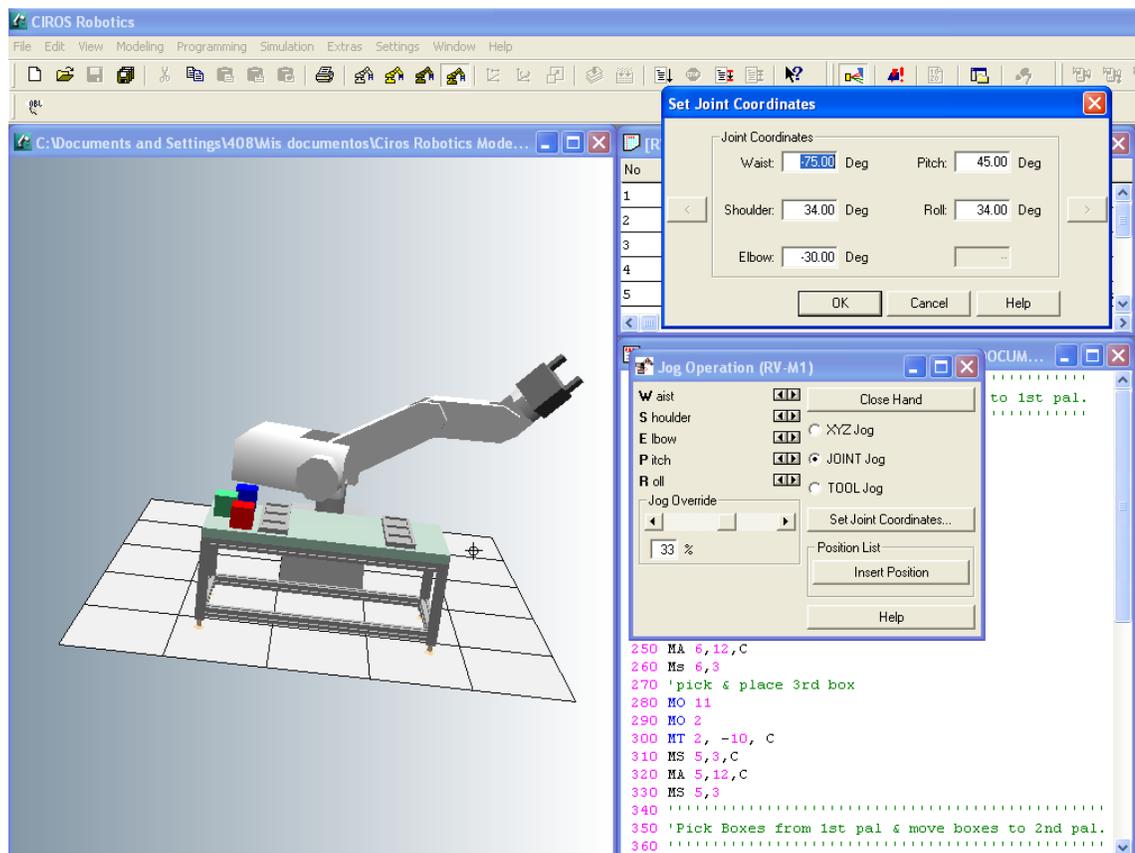


Figura 48 Posición del manipulador simulada en CIROS Festo Automation

Valores ingresados en la ventana 'set joint coordinates':



Figura 49 Valores ingresados a CIROS Festo Automation

Resultado de la posición final en valores numericos:

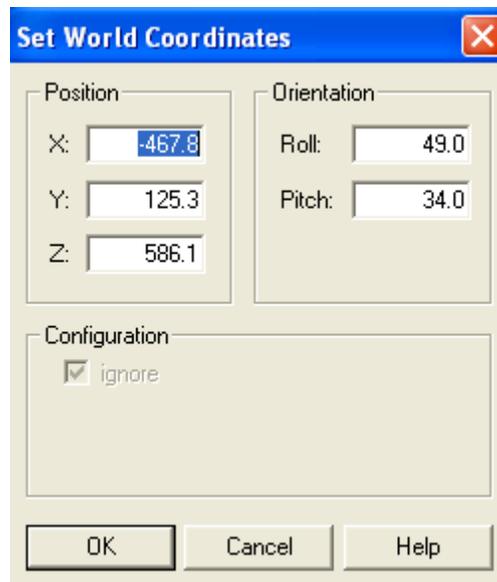


Figura 50 Resultado de las coordenadas finales en CIROS Festo Automation

Software Del Proyecto

Valores ingresados en la interfaz:

- Waist: -75 shoulder: 34 elbow: -30 picth: 45 roll: 34

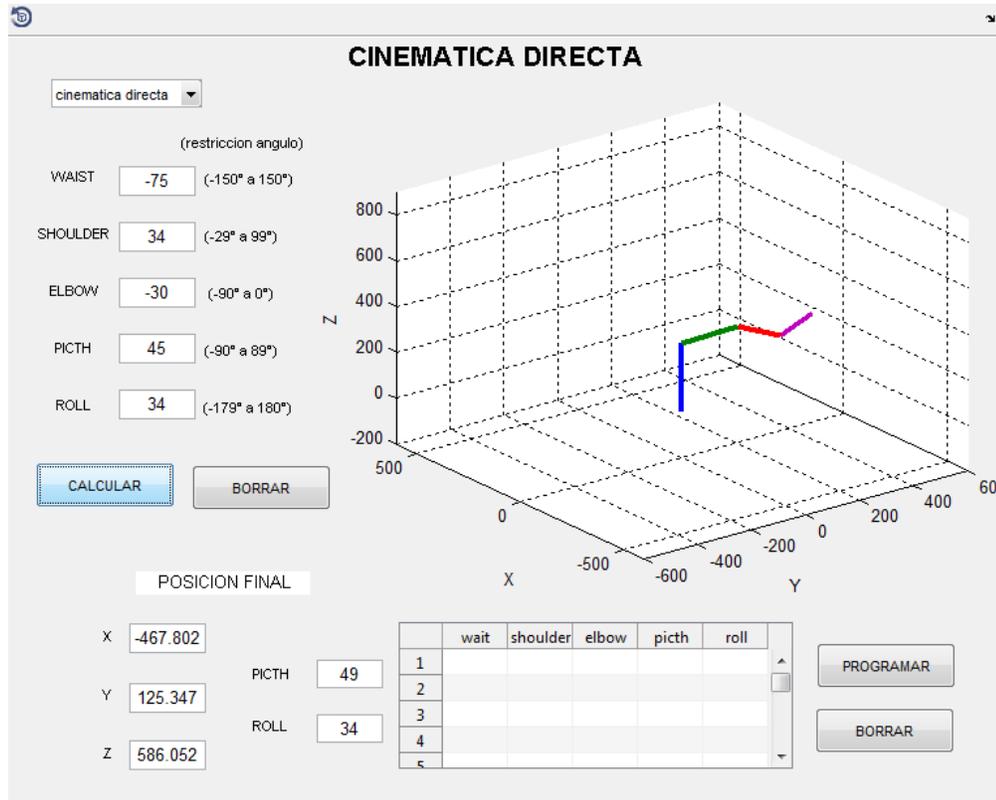


Figura 51 Posición del manipulador simulada en el software del proyecto

Valores ingresados en la interfaz:

(restriccion angulo)

WAIST (-150° a 150°)

SHOULDER (-29° a 99°)

ELBOW (-90° a 0°)

PICTH (-90° a 89°)

ROLL (-179° a 180°)

Figura 52 Valores ingresados a el software del proyecto

Resultado de la posición final en valores numericos:



The image shows a software window titled "POSICION FINAL" with a light gray background. It contains five input fields with numerical values: X is -467.802, Y is 125.347, Z is 586.052, PICTH is 49, and ROLL is 34. The labels X, Y, Z, PICTH, and ROLL are positioned to the left of their respective input boxes.

Variable	Value
X	-467.802
Y	125.347
Z	586.052
PICTH	49
ROLL	34

Figura 53 Resultado de las coordenadas finales en el software del proyecto

6.7.2 Cinemática inversa

En la cinemática inversa se ingresan 3 coordenadas (x,y,z) y 2 valores de ángulos (pictH y roll) que indican la posición en el espacio y como resultado de esta cinemática se obtienen los ángulos de cada una de las 5 articulaciones.

CIROS Festo Automation

En este software lo primero que se realiza es desplegar el menú de 'joint operation (rv-m1)' que aparece en la pantalla al presionar la tecla F8. En este menú se selecciona la opción 'xyz jog', que desplegará otra ventana llamada 'set world coordinates'; en ella se colocan los valores de cada coordenada y los ángulos pictH y roll. A continuación se presiona el botón 'OK' para que muestre la posición final de la gráfica.

Para obtener los valores numéricos de la posición final se selecciona el menú 'joint operation (rv-m1)' la opción 'joint jog' que desplegará una ventana llamada 'set joint coordinates' donde aparece los 5 valores finales de los ángulos de las articulaciones.

A continuación los valores ingresados en la ventana 'set world coordinates':

- Posicion en X: -81 Posicion en Y: 300 Posicion en Z: 742 pitch:-68 roll: 89

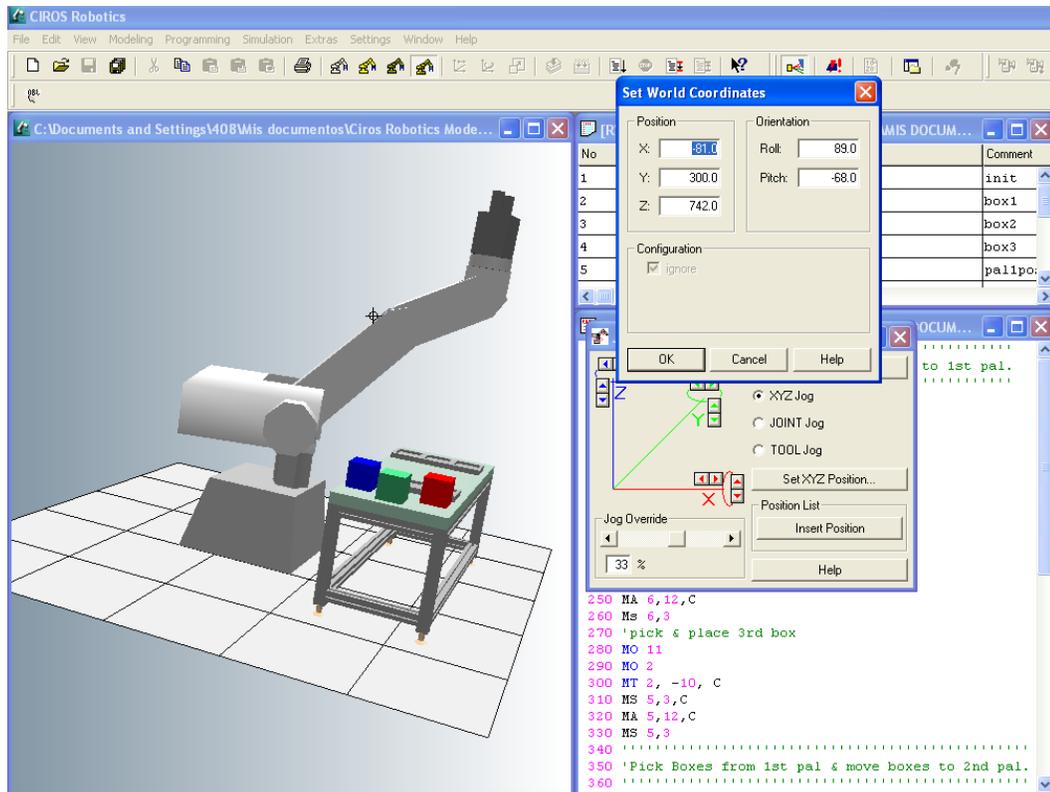


Figura 54 Posición del manipulador simulada en CIROS Festo Automation

Valores ingresados en la ventana 'set world coordinates'

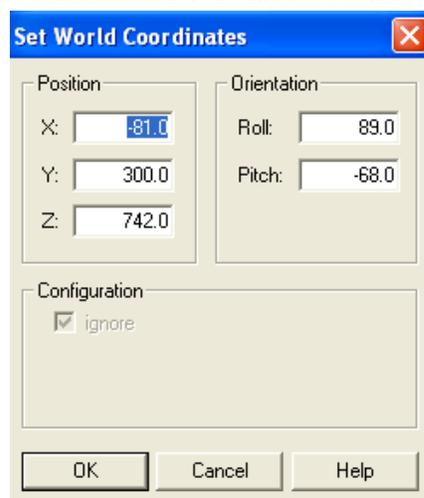


Figura 55 Valores ingresados a CIROS Festo Automation

Resultado numérico de los Angulos:



Figura 56 Resultado de las coordenadas finales en CIROS Festo Automation

Software Del Proyecto

En este software lo primero que se realiza es seleccionar en el menú desplegable la opción 'cinemática inversa' una vez hecho se tendrá en la parte superior izquierda cinco cuadros para ingresar las 3 coordenadas (X,Y,Z) y los ángulos pitch y roll. Una vez ingresados estos datos se oprime el botón 'calcular' y aparece en la grafica el movimiento del robot hasta la posición final y en la parte inferior izquierda se encuentran 5 cuadros donde aparece el valor de los 5 ángulos finales de las articulaciones.

A continuación los valores ingresados en la interfaz:

- Posicion en X: -81 Posicion en Y: 300 Posicion en Z: 742 pitch:-68
roll: 89

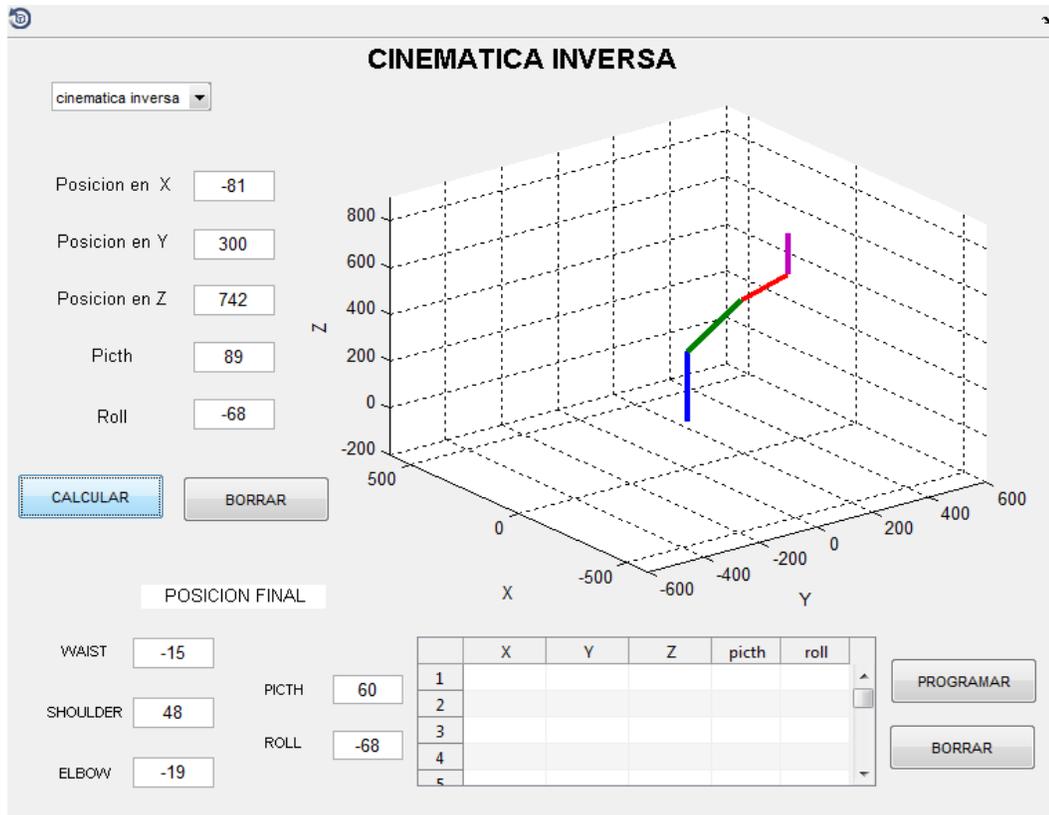


Figura 57 Posición del manipulador simulada en el software del proyecto

Valores ingresados en la interfaz:

Posicion en X: -81

Posicion en Y: 300

Posicion en Z: 742

Picth: 89

Roll: -68

Figura 58 Valores ingresados a el software del proyecto

Resultado numérico de los Angulos:

POSICION FINAL	
WAIST	-15
SHOULDER	48
ELBOW	-19
PICTH	60
ROLL	-68

Figura 59 Resultado de las coordenadas finales en el software del proyecto

A continuación el segundo ejemplo de la comparación de la cinemática inversa.

CIROS Festo Automation

Valores ingresados en la ventana 'set world coordinates':

- Posicion en X: 335 Posicion en Y: 335 Posicion en Z: 378 pitch:66 roll: 12

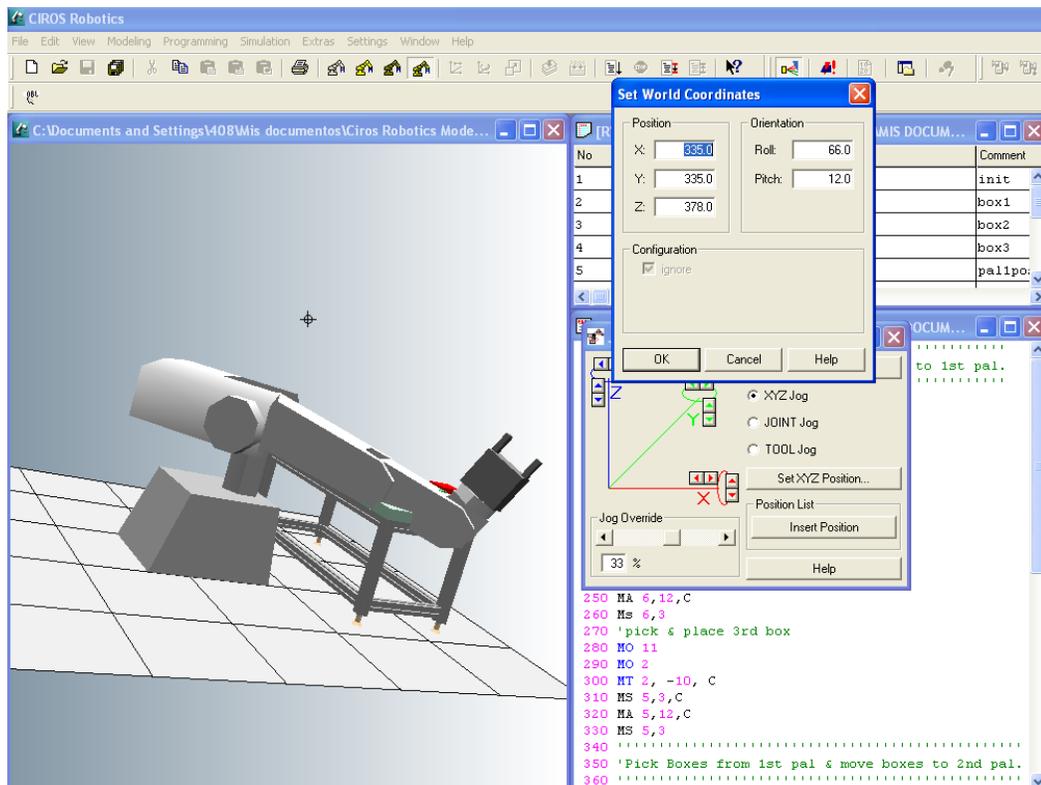


Figura 60 Posición del manipulador simulada en CIROS Festo Automation

Valores ingresados en la ventana 'set world coordinates':

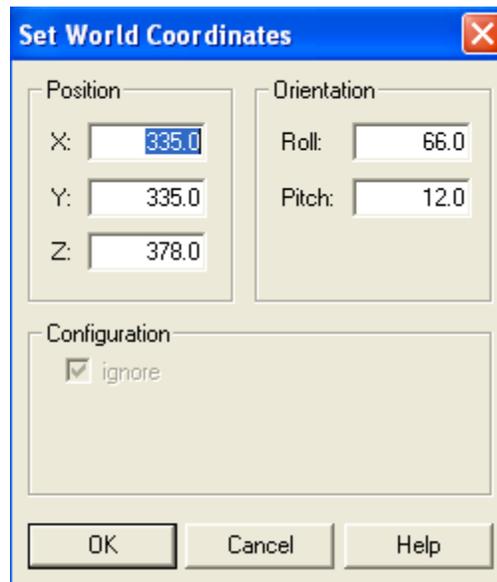


Figura 61 Valores ingresados a CIROS Festo Automation

Resultado numérico de los Ángulos:



Figura 62 Resultado de las coordenadas finales en CIROS Festo Automation

Software Del Proyecto

Valores ingresados en la interfaz:

- Posicion en X: 335 Posicion en Y: 335 Posicion en Z: 378 picth:66 roll: 12

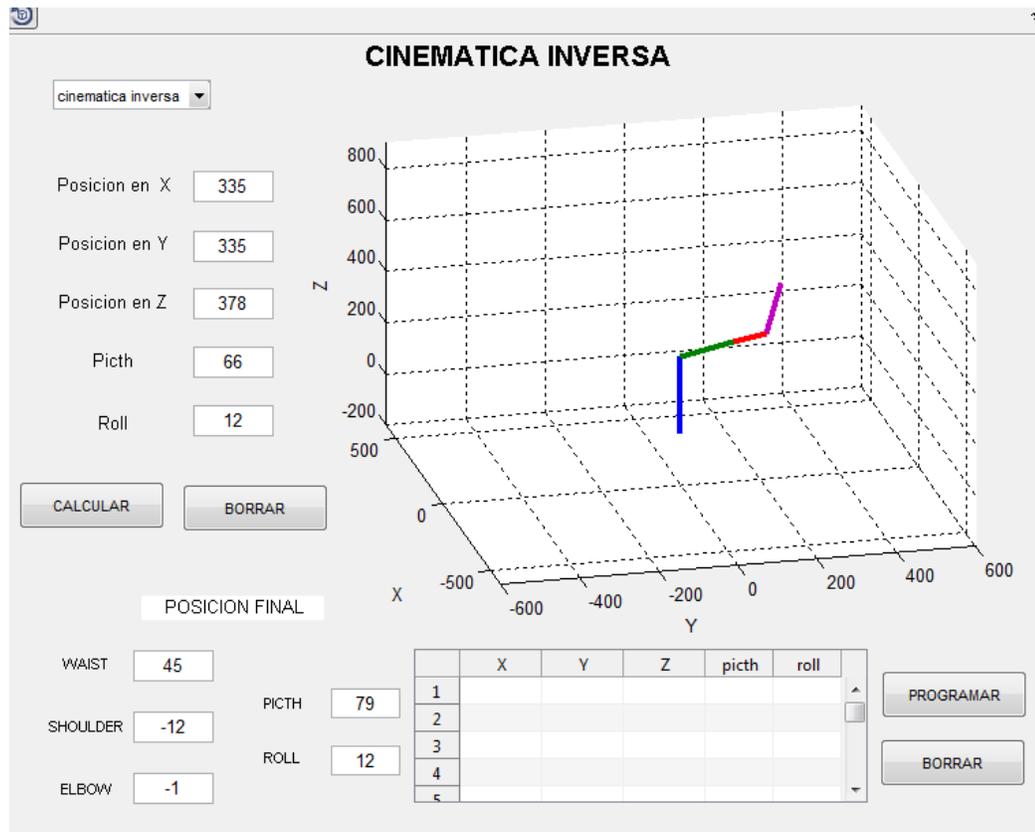


Figura 63 Posición del manipulador simulada en el software del proyecto

Valores ingresados en la interfaz:

Posicion en X	<input type="text" value="335"/>
Posicion en Y	<input type="text" value="335"/>
Posicion en Z	<input type="text" value="378"/>
Picth	<input type="text" value="66"/>
Roll	<input type="text" value="12"/>

Figura 64 Valores ingresados a el software del proyecto

Resultado numérico de los Ángulos:

POSICION FINAL	
WAIST	45
SHOULDER	-12
ELBOW	-1
PICTH	79
ROLL	12

Figura 65 Resultado de las coordenadas finales en el software del proyecto

A continuación el segundo ejemplo de la comparación de la cinemática inversa.

CIROS Festo Automation

Valores ingresados en la ventana 'set world coordinates':

- Posicion en X: 467 Posicion en Y: 0 Posicion en Z: 291 pitch: -56
roll: -135

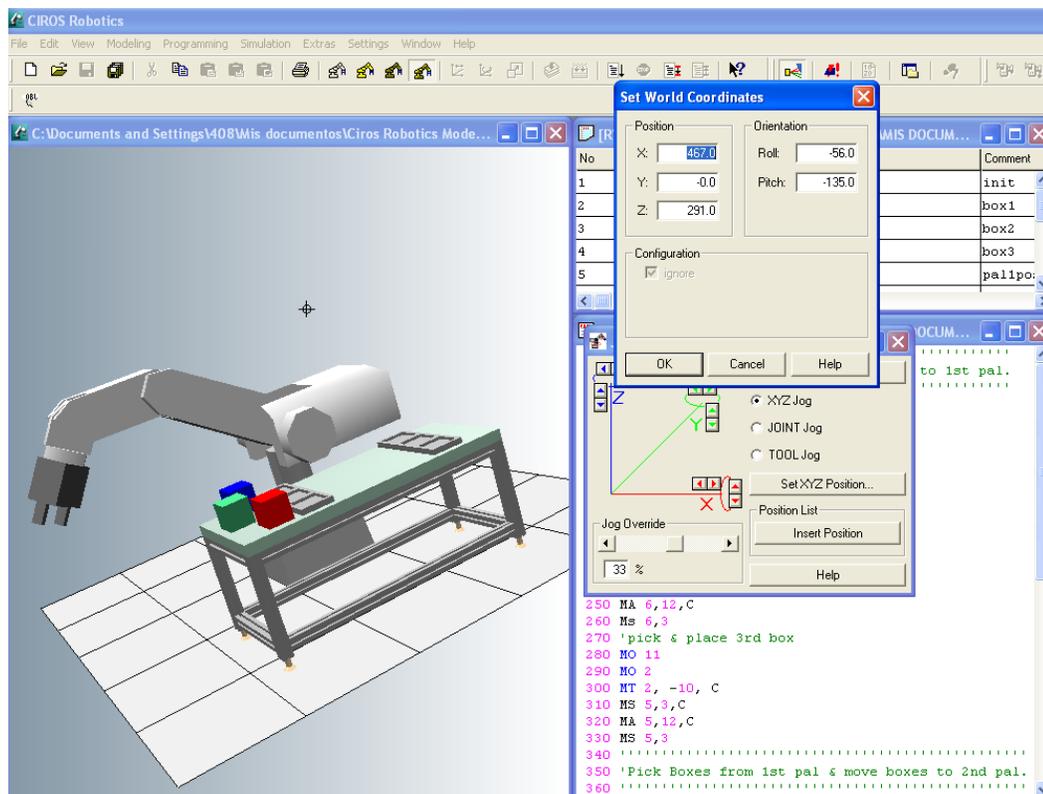


Figura 66 Posición del manipulador simulada en CIROS Festo Automation

Valores ingresados en la ventana 'set world coordinates':

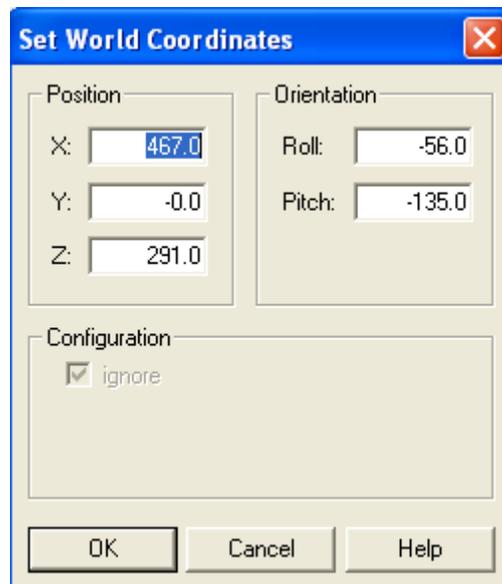


Figura 67 Valores ingresados a CIROS Festo Automation

Resultado Numérico de losÁngulos:



Figura 68 Resultado de las coordenadas finales en CIROS Festo Automation

Software Del Proyecto

Valores ingresados en la interfaz:

- Posicion en X: 467 Posicion en Y: 0 Posicion en Z: 291 picth: -56
roll: -135

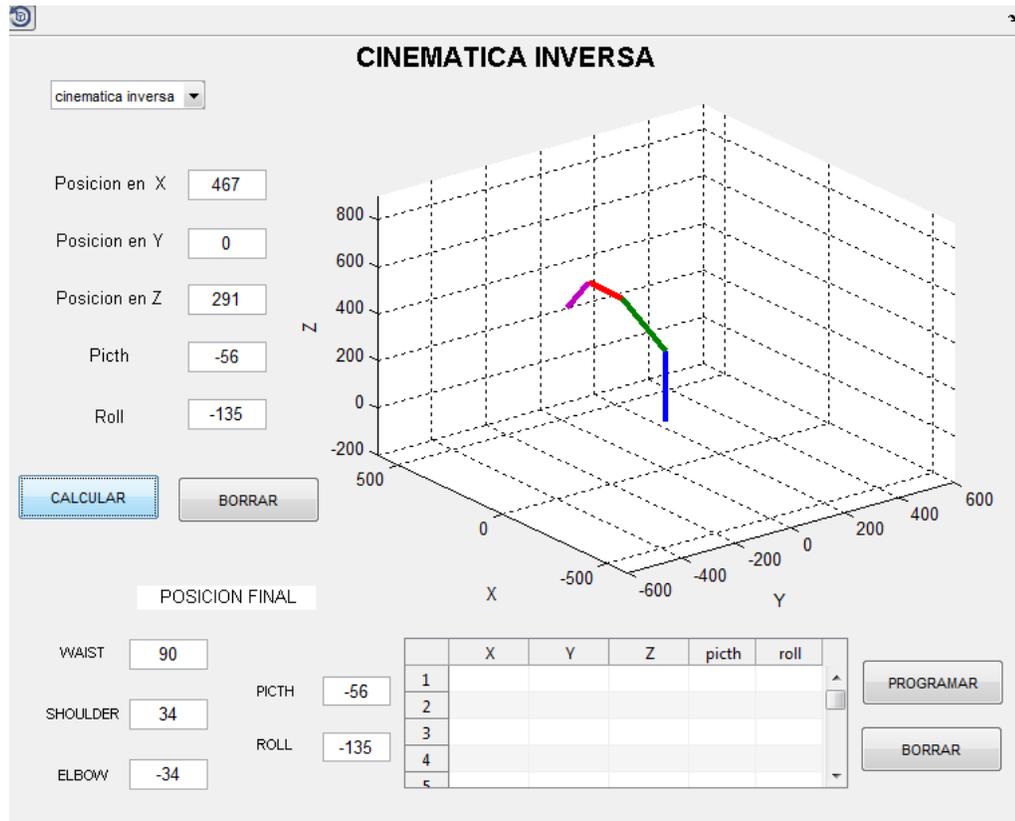


Figura 69 Posición del manipulador simulada en el software del proyecto

Valores ingresados

Posicion en X	<input type="text" value="467"/>
Posicion en Y	<input type="text" value="0"/>
Posicion en Z	<input type="text" value="291"/>
Picth	<input type="text" value="-56"/>
Roll	<input type="text" value="-135"/>

Figura 70 Valores ingresados a el software del proyecto

Resultado Numérico de los Ángulos:

POSICION FINAL	
WAIST	90
SHOULDER	34
ELBOW	-34
PICTH	-56
ROLL	-135

Figura 71 Resultado de las coordenadas finales en el software del proyecto

De esta manera se culmina satisfactoriamente el proceso de modelado y prueba del software presentado en este proyecto, se realizó un software que facilita la interacción del usuario con el programa y se obtuvo una solución actualizada al problema de programación del robot Mitsubishi RV-M1.

7 CONCLUSIONES

- Gracias al análisis realizado de la cinemática de cuerpos rígidos se pudo llegar a determinar que todos los movimientos para determinar la posición de cuerpos rígidos pertenecientes al robot Mitsubishi RV-M1, se pudieron representar vectorialmente en los ejes x, y y z definiendo apropiadamente los sistemas de coordenadas al que estarían incorporados. Esto permitió avanzar en el desarrollo de los cálculos que habilitarían posteriormente el desarrollo de la interfaz.
- A través del planteamiento de las ecuaciones de posición se pudo establecer que los robots manipuladores están ligados a una configuración geométrica de la cual depende el análisis cinemático, adicionalmente los aspectos de sus dimensiones y a la estructura física, que varía generalmente en cuanto a la posición de las articulaciones ya sean en serie o en paralelo, tiene que tenerse muy en cuenta para evitar inconvenientes a la hora de realizar cálculos apropiados.
- A pesar de que en la actualidad existen paquetes de software con características parecidas a las de este proyecto, el desarrollo de los cálculos y la plataforma productos de este se trabajó, se realizó de forma didáctica, con un código y cálculos abiertos para permitir que todos los estudiantes interesados en aprender algo más acerca de las consideraciones para cinemática inversa y directa de un robot manipulador lo puedan hacer sin necesidad de empezar desde cero, sino comenzando con una plataforma cuyos resultados podrán comparar con los movimientos de un robot real que se espera pueda recuperarse en proyectos posteriores.
- El desarrollo de cinemática inversa y cinemática directa se implementó haciendo uso de los parámetros del método de Denavit-Hartenberg y sus matrices de transformación homogénea. Los resultados obtenidos y mostrados aquí una vez más corroboran la vigencia de estos métodos por la

precisión de sus resultados, su relativamente sencilla implementación y la facilidad con la que se puede llegar a dominar los mismos.

- Las pruebas mostradas en Matlab, a pesar de que no es un programa muy eficiente para la visualización, generan la respuesta de forma inmediata y proporciona una gran velocidad a la hora de realizar cualquier tipo de cálculos matemáticos. Un elemento que sin duda alguna se convertirá en una gran ventaja a la hora de conectar de manera directa el software con el sistema de control del robot real.
- Tal y como se estableció al comienzo de este proyecto, y luego de observar la presentación de resultados, es posible concluir que el proyecto permite la creación de una aplicación que simular visualiza un brazo robot Mitsubishi RV M1 de 5 grados de libertad, permitiendo configurar el tipo de trayectoria, y calculando de manera apropiada las posiciones de cada una de las articulaciones. Esto hará de la herramienta un elemento apropiado para la enseñanza de temas de Robótica a nivel de pregrado y muy probablemente permita el avance de nuevos proyectos investigativos en el tema. Además la herramienta resultante es bastante liviana, podrá ser ejecutada en un ordenador de capacidades de cómputo normales, facilita la creación de laboratorios virtuales económicamente más accesibles a las instituciones educativas y matemáticamente confiables.

8 BIBLIOGRAFÍA

- JAZAR, REZA. Theory of Applied Robotics: kinematics, dynamics, and control 4ª edición. Australia, Melbourne. Springer. 685 p.
- SANTOS, VICTOR. Robótica Industrial. 1ª edición. Portugal, Aveiro. Noviembre 2002.
- VENTURELLO MEJIA, JUAN MIGUEL. Herramienta para la programación y simulación de movimientos del manipulador robótico Mitsubishi movemaster RV-M1. 1ª edición. Colombia, Cali. Agosto 2006. 226 pag.
- GUTIERREZ SALAZAR, URIEL. Manual de operación básico. 1ª edición. 1987.
- Cosimir MP Overview [online]. EFR GmbH. Disponible en Internet: URL<http://www.cosimir.com/COSIMIR_MP/CD/English/index.htm>
- Deitel, Harvey M., Deitel, Paul J. Como Programar en C/C++. Naucalpan de Juárez, México. Prentice Hall, 1998.
- Delgado Fernández, Beatriz. Manual Básico Manipulador Robot Mitsubishi RV-M1.
- Dennis, Crain. Win32: Hit Testing Lines and Curves [online]. Microsoft Developer Network, 1994. Disponible en Internet: URL<http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dngdi/html/msdn_hittest2.asp>
- Dennis, Crain. Use of Two-Dimensional Vectors with Windows NT [online]. Microsoft Developer Network, 1993. Disponible en Internet: URL <
http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dngdi/html/msdn_vector.asp>
- Dion, Picco. How To Find Memory Leaks [online]. 2000. Disponible en Internet: URL<http://www.flipcode.com/articles/article_memoryleaks.shtml>
- Jaramillo, Andrés. Robótica de Manipuladores: teoría y algoritmos para modelado, simulación y control. Manuscrito, 1999-2006.
- Mitsubishi Micro-Robot System RV-M1 MoveMaster EX Technical Manual. Mitsubishi Corporation, Japón, 1996.

9 ANEXOS

9.1 Manual del software.

MANUAL DE USO DEL SOFTWARE

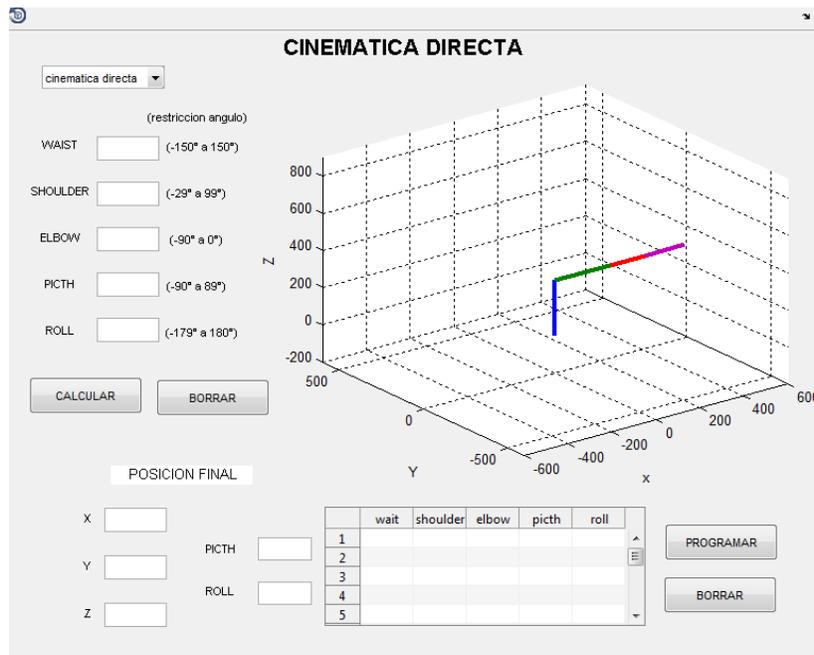
Este software fue hecho en Matlab (abreviatura de MATrix LABoratory, "laboratorio de matrices") es un software matemático que ofrece un entorno de desarrollo integrado(IDE) con un lenguaje de programación propio (lenguaje M). Está disponible para las plataformas Unix, Windows y Apple Mac OS X.

Entre sus prestaciones básicas se hallan: la manipulación de matrices, la representación de datos y funciones, la implementación de algoritmos, la creación de interfaces de usuario (GUI) y la comunicación con programas en otros lenguajes y con otros dispositivos hardware. El paquete MATLAB dispone de dos herramientas adicionales que expanden sus prestaciones, a saber, Simulink (plataforma de simulación multidominio) y GUIDE (editor de interfaces de usuario - GUI).

La presentación final está hecha en GUIDE de Matlab r2010a y esta compilado en un archivo (.exe) para ejecutarlo en cualquier computadora sin necesidad de tener instalado Matlab r2010a.

RECONOCIMIENTO DE LA INTERFAZ

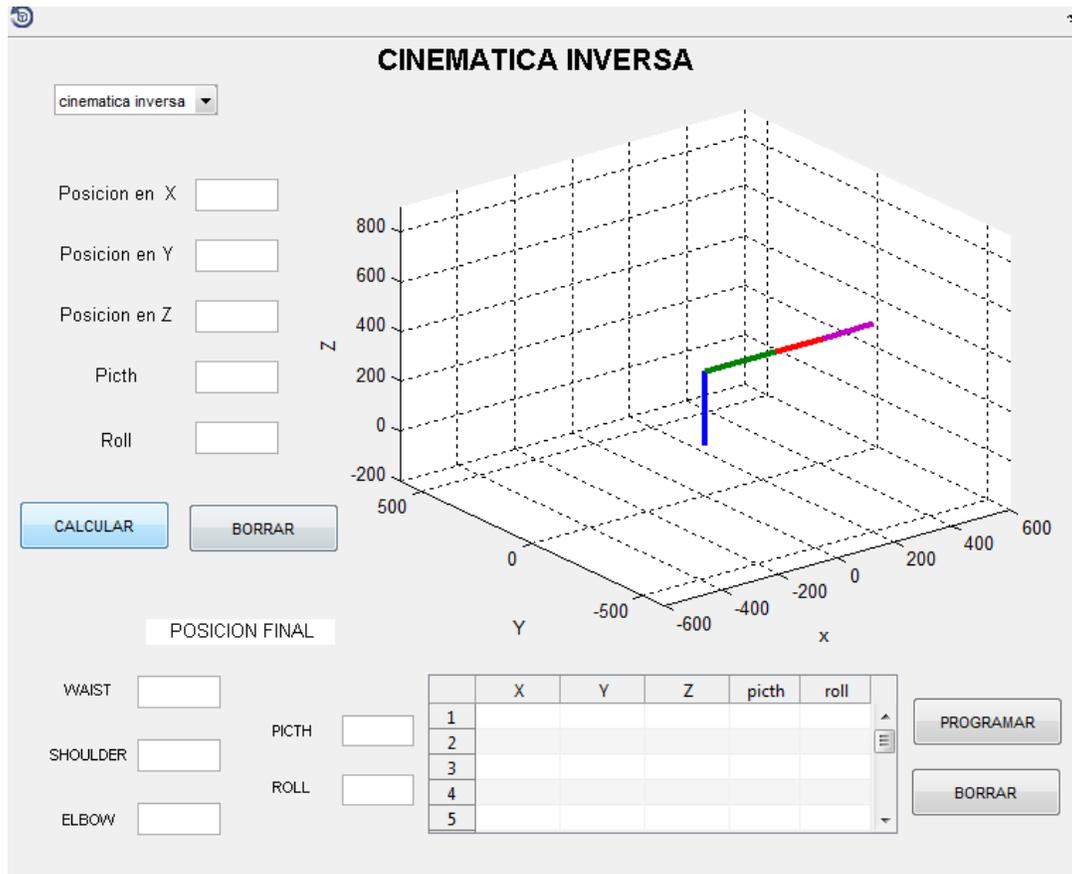
Primero llamamos la ventana de la interfaz. Esto se hace picando sobre el archivo tesis que al abrirse nos mostrara esta ventana.



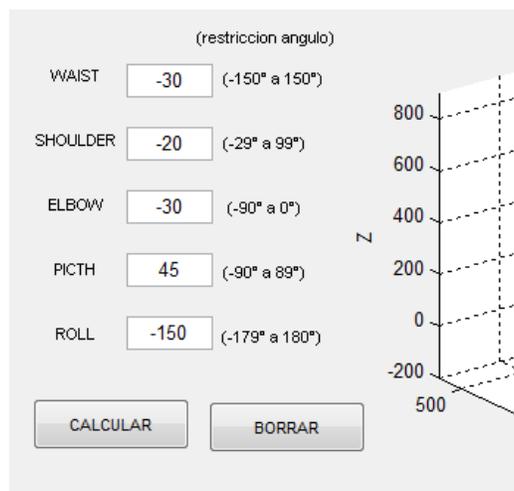
En esta interfaz se podrá ver un menú despegable en la parte superior izquierda que nos re direccionara a la cinemática que se va a trabajar.



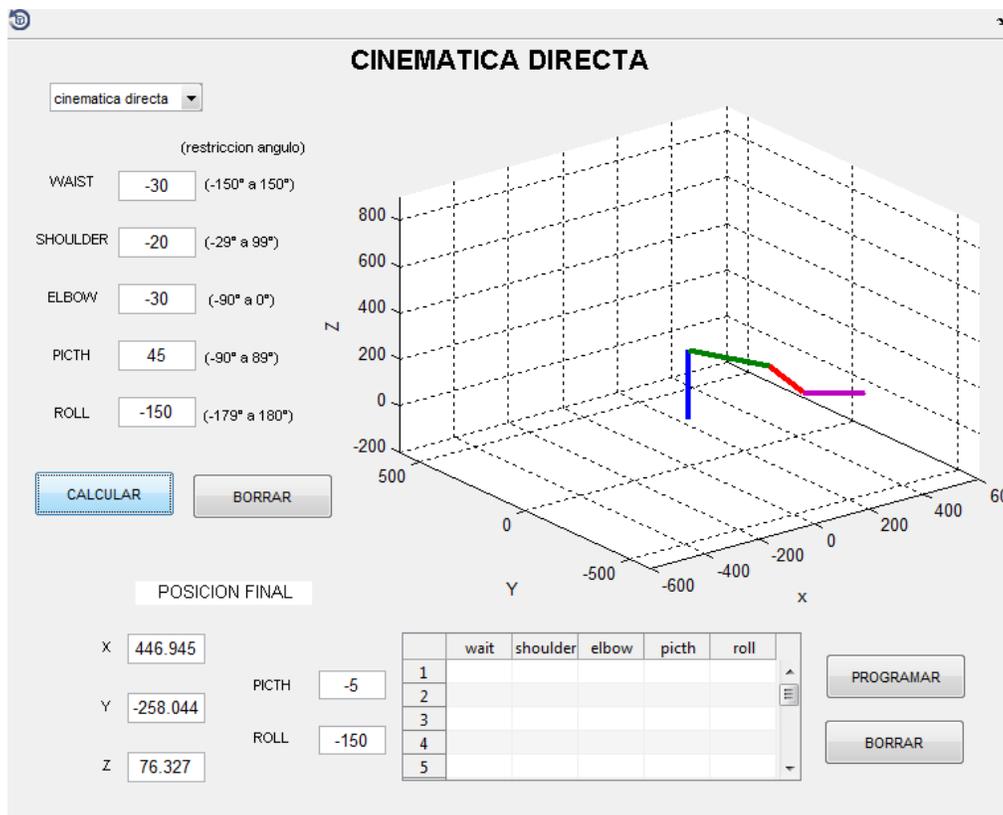
En el anterior menú despegable vemos las 2 opciones de cinemática y la escoge picando con el puntero, que al escoger aparece una nueva ventana.



Para **CINEMATICA DIRECTA** picamos en el menú despegable la opción “cinemática directa”. Después introducimos los ángulos de cada articulación del robot en la parte superior izquierda.



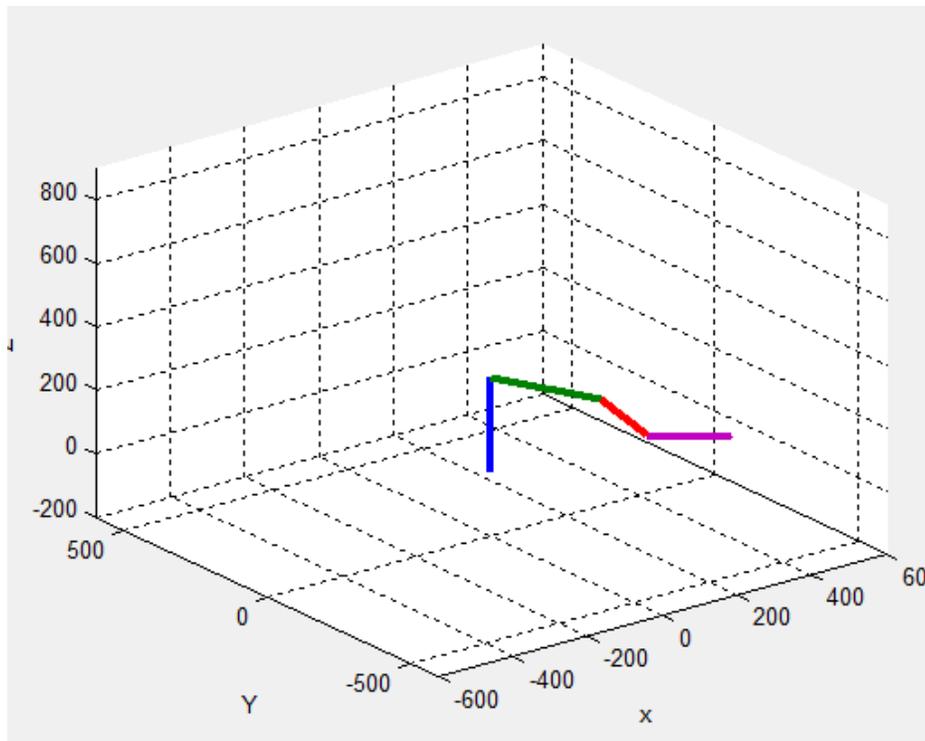
Después de introducir los ángulos deseados picamos el botón “ CALCULAR” e inmediatamente después la gráfica que está en su estado inicial, comenzara a hacer sus movimientos hasta llegar a los ángulos introducidos.



En esta interfaz muestra la posición final en coordenadas X, Y, Z con su correspondiente ángulo picth y roll. En la parte inferior izquierda.



También muestra en la parte superior derecha como se comporta el robot en cada instante del movimiento y termina mostrando su posición final.



Para volver a la posición inicial de la gráfica picamos el botón “BORRAR” e inmediatamente aparecerá reseteada la posición final y la gráfica en su estado inicial.

CINEMATICA DIRECTA

cinematica directa

(restriccion angulo)

WAIST (-150° a 150°)

SHOULDER (-29° a 99°)

ELBOW (-90° a 0°)

PICHT (-90° a 89°)

ROLL (-179° a 180°)

POSICION FINAL

X

Y

Z

PICHT

ROLL

	wait	shoulder	elbow	picth	roll
1					
2					
3					
4					
5					

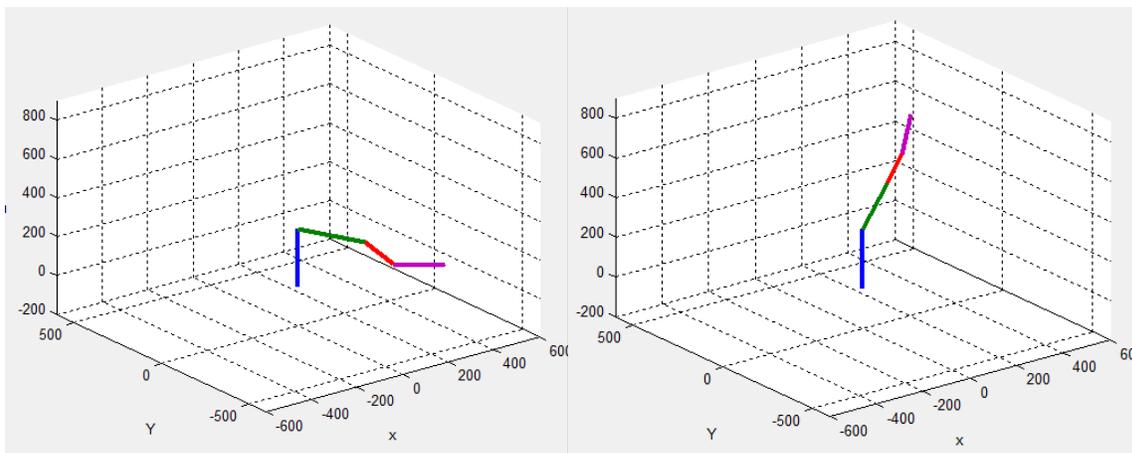
En la parte inferior derecha tenemos un cuadro Excel que sirve para ingresar hasta 15 posiciones determinadas para hacer secuencias de movimientos.

	wait	shoulder	elbow	picth	roll
1					
2					
3					
4					
5					

Se ingresan los valores de los ángulos y picamos el botón “PROGRAMAR”.

	X	Y	Z	pitch	roll
1	-30	-20	-30	45	-150
2	30	30	0	34	-10
3					
4					
5					

Esto nos muestra la gráfica de la primera posición y luego se moverá a la posición de la segunda posición.



En la parte izquierda de la anterior figura, se observa la gráfica final de la primera posición y en la parte derecha se ve la gráfica final de la segunda posición.

Para volver a la posición inicial de la gráfica picamos el botón “BORRAR” e inmediatamente aparecerá reseteada la posición final, la gráfica en su estado inicial y en la tabla Excel las 2 posiciones para modificarlas.

	X	Y	Z	pitch	roll
1	-30	-20	-30	45	-150
2	30	30	0	34	-10
3					
4					
5					

Para **CINEMATICA INVERSA** picamos en el menú despegable la opción “cinemática inversa”. Después introducimos las coordenadas finales del robot en la parte superior izquierda.

Posicion en X

Posicion en Y

Posicion en Z

Picth

Roll

Z

800

600

400

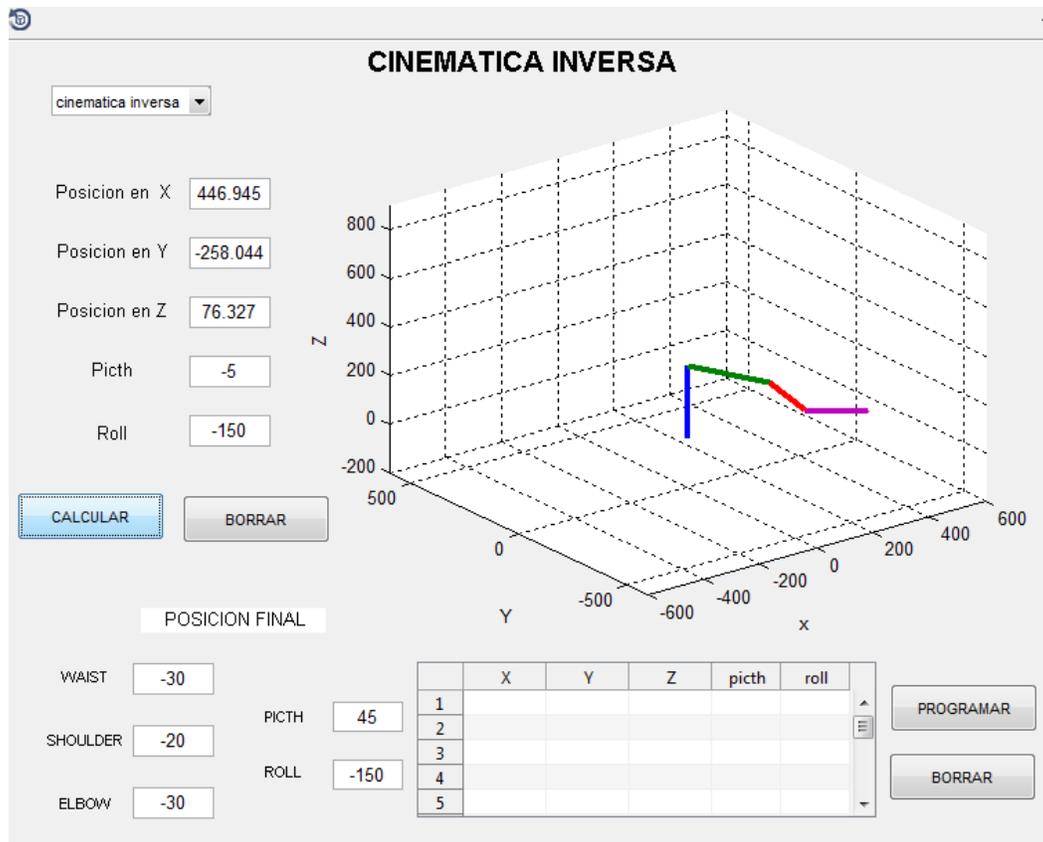
200

0

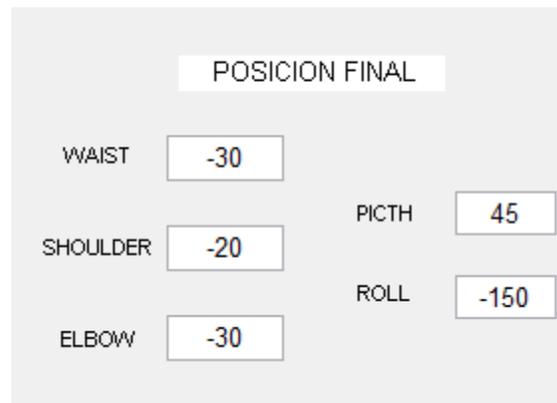
-200

500

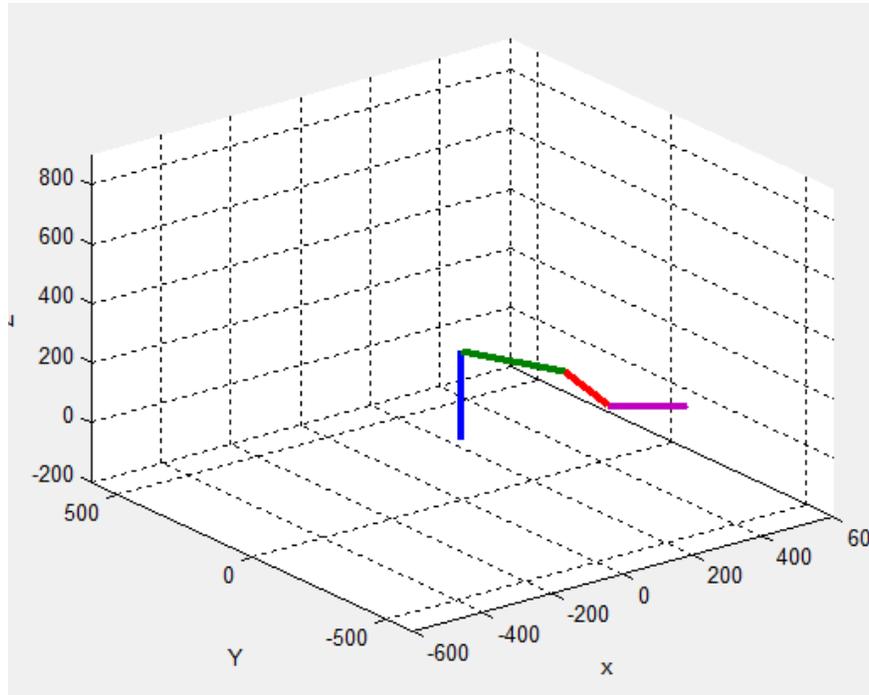
Después de introducir las coordenadas deseadas picamos el botón “CALCULAR” e inmediatamente después la gráfica que está en su estado inicial, comenzara a hacer sus movimientos hasta llegar a las coordenadas introducidas.



En esta interfaz muestra la posición final de los ángulos waist, shoulder, elbow, pitch y roll. En la parte inferior izquierda.



También muestra en la parte superior derecha como se comporta el robot en cada instante del movimiento y termina mostrando su posición final.



Para volver a la posición inicial de la gráfica picamos el botón “BORRAR” e inmediatamente aparecerá reseteada la posición final y la gráfica en su estado inicial.

CINEMATICA INVERSA

cinematica inversa

Posicion en X

Posicion en Y

Posicion en Z

Picth

Roll

POSICION FINAL

WAIST

SHOULDER

ELBOW

PICth

ROLL

	X	Y	Z	picth	roll
1					
2					
3					
4					
5					

En la parte inferior derecha tenemos un cuadro Excel que sirve para ingresar hasta 15 posiciones determinadas para hacer secuencias de movimientos.

	X	Y	Z	pitch	roll
1					
2					
3					
4					
5					

PROGRAMAR

BORRAR

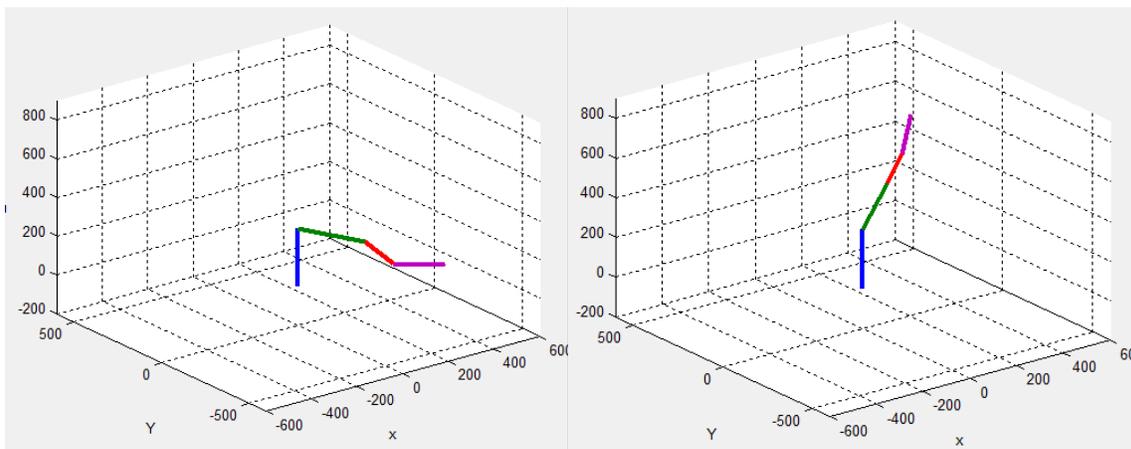
Ingresamos los valores de las coordenadas y picamos el botón “PROGRAMAR”.

	X	Y	Z	pitch	roll
1	446.9450	-258.0440	76.3270	-5	-150
2	375.4560	216.7690	665.8840	64	-10
3					
4					
5					

PROGRAMAR

BORRAR

Esto nos muestra la gráfica de la primera posición y luego se moverá a la posición de la segunda posición.



En la parte izquierda de la figura, se observa la gráfica final de la primera posición y en la parte derecha se ve la gráfica final de la segunda posición.

Para volver a la posición inicial de la gráfica picamos el botón “BORRAR” e inmediatamente aparecerá reseteada la posición final, la gráfica en su estado inicial y en la tabla Excel las 2 posiciones para modificarlas.

	X	Y	Z	pitch	roll	
1	446.9450	-258.0440	76.3270	-5	-150	▲
2	375.4560	216.7690	665.8840	64	-10	☰
3						
4						
5						▼

PROGRAMAR

BORRAR

9.2 Desarrollo algebraico cinemática inversa

$$1. l_1 C\theta_1 C\theta_2 - l_3 C\theta_1 (S\theta_2 S\theta_3 + C\theta_2 C\theta_3) = P_x$$

$$2. l_1 C\theta_2 S\theta_1 + l_3 S\theta_1 (C\theta_2 C\theta_3 - S\theta_2 S\theta_3) = P_y$$

$$3. l_1 S\theta_2 + l_3 (C\theta_2 S\theta_3 + C\theta_3 S\theta_2) + l_0 = P_z$$

Desarrollo del primer ángulo θ_1

Utilizando la ecuación 1 y 2, se saca el término de igualdad denominado A

$$A = (C\theta_2 C\theta_3 - S\theta_2 S\theta_3)$$

$$A = S(\theta_2 + \theta_3)$$

Luego,

$$1. l_1 C\theta_1 C\theta_2 - l_3 C\theta_1 A = P_x$$

$$2. l_1 C\theta_2 S\theta_1 + l_3 S\theta_1 A = P_y$$

De 1,

$$4. C\theta_2 = \frac{P_x - l_3 C\theta_1 A}{l_1 C\theta_1}$$

4 en 1,

$$l_1 S\theta_1 \left(\frac{P_x - l_3 C\theta_1 A}{l_1 C\theta_1} \right) + l_3 S\theta_1 A = P_y$$

$$\tan \theta_1 (P_x - l_3 C\theta_1 A) + l_3 S\theta_1 A = P_y$$

$$\tan \theta_1 P_x - l_3 S\theta_1 A + l_3 S\theta_1 A = P_y$$

$$\tan \theta_1 P_x = P_y$$

$$\tan \theta_1 = \frac{P_y}{P_x}$$

$$\theta_1 = \tan^{-1} \left(\frac{P_y}{P_x} \right)$$

Desarrollo del segundo ángulo θ_2

De la ecuación 3

$$B = (C\theta_2 S\theta_3 + C\theta_3 S\theta_2)$$

$$B = C(\theta_2 + \theta_3)$$

De la ecuación 1

De la ecuación 2

$$C\theta_1 = \frac{P_x}{l_1 C\theta_2 + l_2 A} \quad S\theta_1 = \frac{P_y}{l_1 C\theta_1 + l_2 A}$$

Por la identidad trigonométrica se hace igualación

$$\text{Sen}^2 \theta_1 + \text{Cos}^2 \theta_1 = 1$$

$$\left(\frac{P_x}{l_1 C\theta_2 + l_2 A} \right)^2 + \left(\frac{P_y}{l_1 C\theta_2 + l_2 A} \right)^2 = 1$$

$$(P_x + P_y)^2 = (l_1 C\theta_2 + l_2 A)^2$$

$$\sqrt{P_x + P_y} = l_1 C\theta_2 + l_2 A$$

$$A = \frac{\sqrt{P_x + P_y} - l_1 C\theta_2}{l_2}$$

De la ecuación 3,

$$B = \frac{P_z - l_0 - l_1 S\theta_2}{l_2}$$

$$A^2 + B^2 = 1$$

$$\left(\sqrt{P_x + P_y} - l_1 C\theta_2\right)^2 + \left(P_z - l_0 - l_1 S\theta_2\right)^2 = l_2^2$$

$$P_x^2 + P_y^2 + P_z^2 + l_0^2 + l_1^2 + l_3^2 - 2P_z l_0 + 2S\theta_2 l_1 (l_0 - P_z) - 2l_1 C\theta_2 \sqrt{P_x^2 + P_y^2} = 0$$

Para llevarlo a la relación trigonométrica:

$$aC\theta_2 + bS\theta_2 = c$$

Dónde,

$$a = 2l_1 \sqrt{P_x^2 + P_y^2}$$

$$b = 2l_1 (l_0 - P_z)$$

$$c = P_x^2 + P_y^2 + P_z^2 + l_0^2 + l_1^2 + l_3^2 - 2P_z l_0$$

Para realizar un correcto análisis al despejar las ecuaciones anteriores en términos del ángulo θ_2 , ayudándose de las siguientes identidades trigonométricas de ángulos dobles:

$$S\theta_2 = \frac{2 \tan\left(\frac{\theta_2}{2}\right)}{1 + \tan^2\left(\frac{\theta_2}{2}\right)} \quad C\theta_2 = \frac{1 - \tan^2\left(\frac{\theta_2}{2}\right)}{1 + \tan^2\left(\frac{\theta_2}{2}\right)}$$

Se utiliza la relación trigonométrica de Pitágoras reemplazando

Sea $\left(\frac{\theta_2}{2}\right) = x$

$$a \left[\frac{2 \tan(x)}{1 + \tan^2(x)} \right] + b \left[\frac{1 - \tan^2(x)}{1 + \tan^2(x)} \right] = c$$

$$\tan^2(x)(a + c) - 2b \tan(x) - (a - c) = 0$$

$$a' = (a + c)$$

$$b' = 2b$$

$$c' = -a + c$$

$$x = \frac{-b' \pm \sqrt{(b')^2 - 4a'c'}}{2a'}$$

$$x = \frac{b \pm \sqrt{b^2 + a^2 - c^2}}{a + c}$$

Finalmente,

$$\theta_2 = 2 \text{ATAN} \left[\frac{b \pm \sqrt{b^2 + a^2 - c^2}}{a + c} \right]$$

Desarrollo del segundo ángulo θ_3

Recordando que

$$\begin{aligned} A &= (C\theta_2 C\theta_3 - S\theta_2 S\theta_3) & B &= (C\theta_2 S\theta_3 + C\theta_3 S\theta_2) \\ A &= S(\theta_2 + \theta_3) & B &= C(\theta_2 + \theta_3) \end{aligned}$$

Luego,
$$\text{Sen}^2\theta_1 + \text{Cos}^2\theta_1 = 1$$

$$A^2 + B^2 = 1$$

Sumando los términos de las ecuaciones 1, 2, y 3 así:

$$1. \quad l_1 C\theta_1 C\theta_2 - l_3 C\theta_1 (S\theta_2 S\theta_3 + C\theta_2 C\theta_3) = P_x$$

$$2. \quad l_1 C\theta_2 S\theta_1 + l_3 S\theta_1 (C\theta_2 C\theta_3 - S\theta_2 S\theta_3) = P_y$$

$$3. \quad l_1 S\theta_2 + l_3 (C\theta_2 S\theta_3 + C\theta_3 S\theta_2) + l_0 = P_z$$

$$*[l_1 C\theta_1 C\theta_2 - l_3 C\theta_1 A] + [l_1 C\theta_2 S\theta_1 + l_3 S\theta_1 A]$$

$$+ [l_1 S\theta_2 + l_3 B + l_0] = P_x + P_y + P_z$$

$$*[l_1 C\theta_1 C\theta_2 - l_3 C\theta_1 A]^2 + [l_1 C\theta_2 S\theta_1 + l_3 S\theta_1 A]^2$$

$$+ [l_1 S\theta_2 + l_3 B + l_0]^2 = P_x^2 + P_y^2 + P_z^2$$

$$*l_1^2 C^2\theta_1 C^2\theta_2 + 2l_1 l_3 C^2\theta_1 C\theta_2 A + l_3^2 C^2\theta_1 A^2 + l_1^2 C^2\theta_2 S^2\theta_1 + 2l_1 l_3 S^2\theta_1 C\theta_2 A$$

$$+ l_3^2 S^2\theta_1 A^2 + l_1^2 S^2\theta_2 + 2l_1 l_3 S\theta_2 B + l_2^2 B^2 + 2l_1 l_0 S\theta_2 + 2l_0 l_3 B + l_0^2 = P_x^2 + P_y^2 + P_z^2$$

$$*l_1^2 [C^2\theta_2 + S^2\theta_2] + l_3^2 [A^2 + B^2] + 2l_1 l_3 C\theta_2 A + 2l_1 l_3 S\theta_2 B$$

$$+ 2l_1 l_0 S\theta_2 + 2l_0 l_3 B + l_0^2 = P_x^2 + P_y^2 + P_z^2$$

$$\begin{aligned}
& *l_1^2 [C^2\theta_2 + S^2\theta_2] + l_3^2 [A^2 + B^2] + 2l_1l_3C\theta_2A + 2l_1l_3S\theta_2B \\
& + 2l_1l_0S\theta_2 + 2l_0l_3B + l_0^2 = P_x^2 + P_y^2 + P_z^2 \\
& *l_0^2 + l_1^2 + l_3^2 + 2l_1l_3C\theta_2 [C\theta_2C\theta_3 - S\theta_2S\theta_3] + 2l_1l_3S\theta_2 [C\theta_2S\theta_3 - S\theta_2C\theta_3] \\
& + 2l_1l_0S\theta_2 + 2l_0l_3 [C\theta_2S\theta_3 - S\theta_2C\theta_3] = P_x^2 + P_y^2 + P_z^2 \\
& *l_0^2 + l_1^2 + l_3^2 + 2l_1l_3C\theta_2^2C\theta_3 - 2l_1l_3C\theta_2S\theta_2S\theta_3 + 2l_1l_3C\theta_2S\theta_2S\theta_3 + 2l_1l_3S\theta_2^2C\theta_3 \\
& + 2l_1l_0S\theta_2 + 2l_0l_3C\theta_2S\theta_3 + 2l_0l_3C\theta_3S\theta_2 = P_x^2 + P_y^2 + P_z^2 \\
& *2l_1l_3C\theta_3 [C^2\theta_2 + S^2\theta_2] + 2l_1l_0S\theta_2 + 2l_0l_3C\theta_2S\theta_3 + 2l_0l_3C\theta_3S\theta_2 = P_x^2 + P_y^2 + P_z^2 \\
& *2l_1l_3C\theta_3 + 2l_1l_0S\theta_2 + 2l_0l_3C\theta_2S\theta_3 + 2l_0l_3C\theta_3S\theta_2 = P_x^2 + P_y^2 + P_z^2
\end{aligned}$$

Para llevarlo a la relación trigonométrica:

$$aC\theta_2 + bS\theta_2 = c$$

Dónde,

$$*C\theta_3 [2l_1l_3 + 2l_0l_3S\theta_2] + S\theta_3 [2l_0l_3C\theta_2] = P_x^2 + P_y^2 + P_z^2 - 2l_1l_0S\theta_2$$

y

$$a = 2l_1l_3 + 2l_0l_3S\theta_2$$

$$b = 2l_0l_3C\theta_2$$

$$c = P_x^2 + P_y^2 + P_z^2 - 2l_1l_0S\theta_2$$

Finalmente,

$$\theta_3 = 2ATAN \left[\frac{b \pm \sqrt{b^2 + a^2 - c^2}}{a + c} \right]$$