



SECRETARÍA DE EDUCACIÓN PÚBLICA
TECNOLÓGICO NACIONAL DE MÉXICO
INSTITUTO TECNOLÓGICO DE MÉRIDA

ITM

TESIS
“SIMULADOR DE UN BRAZO ROBÓTICO
MITSUBISHI RV-M1”

PARA OPTAR AL GRADO DE:
MAESTRO EN INGENIERÍA

PRESENTA:
I.S.C. JONATHAN H. APONTE

ASESOR:
M.C. MARIO RENÁN MORENO SABIDO

MÉRIDA, YUCATÁN, MÉXICO
28 DE AGOSTO DE 2018



DEPENDENCIA: DIV. DE EST. DE POSG. E INV.
No. DE OFICIO: X-048/2018

ASUNTO: AUTORIZACIÓN DE IMPRESIÓN

MÉRIDA, YUCATÁN A 06 DE FEBRERO DE 2018

C. HERRERA APONTE JONATHAN.
Pasante de Maestría en Ingeniería
Presente.

De acuerdo al fallo emitido por su asesor el **M.C. Mario Renán Moreno Sabido**, y la comisión revisora integrada por el Dr. Jesús Sandoval Gío, el Dr. José Ramón Atoche Enseñat, el Dr. Carlos Alberto Luján Ramírez y el Dr. Agustín Flores Novelo considerando que cubre los requisitos establecidos en el Reglamento de Titulación de los Institutos Tecnológicos le autorizamos la impresión de su trabajo profesional con la TESIS:

"SIMULADOR DE UN BRAZO ROBÓTICO MITSUBISHI RV-M1"

ATENTAMENTE
IN HOC SIGNO VINCES

M.C. DANIEL ARCÁNGEL LÓPEZ SAURI
JEFE DE LA DIVISIÓN DE ESTUDIOS
DE POSGRADO E INVESTIGACIÓN

C.p. Archivo
DALS/fja



S. E. P.
**INSTITUTO TECNOLÓGICO
DE MÉRIDA**
**DIVISIÓN DE ESTUDIOS DE
POSGRADO E INVESTIGACIÓN**



DEDICATORIAS

Dios, por darme la oportunidad de vivir y por estar conmigo en cada paso que doy, por fortalecer mi corazón e iluminar mi mente y por haber puesto en mi camino a aquellas personas que han sido mi soporte y compañía durante todo el periodo de estudio.

A mi esposa Angélica el amor de mi vida que día a día y en todo momento ha dedicado su vida y su tiempo en apoyarme en los momentos más difíciles de mi vida incondicionalmente, gracias por todo.

Para mi abuela Paulina Bautista por apoyarme todos los días su vida, cuidarme, quererme y aconsejarme incondicionalmente gracias por una vida dedicada a mí.

Para mis Padres Estela y Adolfo que siempre me apoyaron incondicionalmente, guiaron y aconsejaron toda la vida sin importar los momentos y las circunstancias.

A mi madre Maria que siempre me lleva en sus pensamientos, ora y pide por mí todos los días con tanto amor y cariño sin importar la distancia y las circunstancias.

AGRADECIMIENTOS

- A todo el comité y maestros pertenecientes a la Maestría en Ingeniería del Instituto Tecnológico de Mérida, en especial al Dr. Jesús Sandoval Guío, al Dr. José Ramón Atoche Enseñat y al DR. Carlos Lujan Ramírez por su comprensión y apoyo en todo momento para poder culminar con mis estudios. Muchas gracias por todo.
- Al M.C Mario Renán Moreno Sabido por estar siempre pendiente, apoyarme y alentarme a terminar mis estudios con esa calidez humana y profesional única que lo caracteriza. Te agradezco de corazón.
- Al M.C Silvia Ruiz Casanova por su apoyo y guía para poder llevar acabo todos los trámites necesarios muy importantes para poder terminar con mis estudios. Le agradezco muchas gracias.
- A todas las personas en general que estuvieron apoyándome directamente o indirectamente durante todos estos años para poder culminar esta etapa de mi vida muchas gracias a todos.

RESUMEN

En esta tesis se presenta un modelo cinemático completo del manipulador Mitsubishi RV-M1 de 5 grados de libertad. El modelo cinemático directo se basa en la notación Denavit-Hartenberg, y el modelo cinemático inverso se deriva a partir de un modelo de matrices inversas para fijar la orientación de la herramienta. Se desarrolla una interfaz gráfica usando el software MATLAB para ilustrar la cinemática directa e inversa, permitiendo al estudiante o al investigador tener un modelo gráfico virtual que describa completamente, tanto la geometría del robot, como el movimiento del robot en su área de trabajo antes de abordar cualquier tarea real.

ABSTRACT

A complete kinematic model of the 5 DOF-Mitsubishi RV-M1 manipulator is presented in this thesis. The forward kinematic model is based on the Modified Denavit-Hartenberg notation, and the inverse one is derived in closed form by fixing the orientation of the tool by a modeling of inverse matrices. A graphical interface is developed using MATLAB software to illustrate the forward and inverse kinematics, allowing student or researcher to have hands-on of virtual graphical model that fully describe both the robot's geometry and the robot's motion in its workspace before to tackle any real task.

ÍNDICE

CAPITULO 1: INTRODUCCIÓN.....	1
1.1. ANTECEDENTES.....	1
1.2. PLANTEAMIENTO DEL PROBLEMA.....	1
1.3. OBJETIVOS.....	2
1.3.1. OBJETIVO GENERAL.....	2
1.3.2. OBJETIVOS ESPECÍFICOS.....	2
1.4. ALCANCES DEL PROYECTO.....	2
1.5. JUSTIFICACIÓN.....	2
CAPÍTULO 2: MARCO TEÓRICO.....	3
2.1. INTRODUCCIÓN.....	3
2.2. ANTECEDENTES.....	3
2.3. BRAZO ROBÓTICO.....	3
2.4. CONCEPTOS BÁSICOS.....	3
2.4.1. CONTROLADOR.....	3
2.4.2. GRADOS DE LIBERTAD (GDL).....	4
2.4.3. ESPACIO DE TRABAJO.....	4
2.4.4. EXACTITUD DE LOS MOVIMIENTOS.....	4
2.5. TIPOS DE BRAZOS ROBÓTICOS.....	5
2.5.1. SCARA.....	6
2.5.2. CARTESIANO.....	6
2.5.3. CILÍNDRICOS.....	6
2.5.4. ESFÉRICO.....	6
2.6. SOFTWARE PARA MITSUBISHI RV-M1.....	7
2.6.1. COSIMIR.....	7
2.7. METODOLOGÍA.....	13
2.7.1. MODELO EN CASCADA.....	14
2.7.2. INGENIERÍA Y ANÁLISIS DEL SISTEMA.....	14
2.7.3. ANÁLISIS DE LOS REQUISITOS DEL SOFTWARE.....	14
2.7.4. DISEÑO.....	14
2.7.5. CODIFICACIÓN.....	14
2.7.6. PRUEBAS.....	14
2.7.7. MANTENIMIENTO.....	14
2.8. HERRAMIENTAS PARA EL DESARROLLO DE LA APLICACIÓN.....	15

2.8.1.	MATLAB.	15
2.8.2.	SIMULINK.	15
CAPÍTULO 3: DESARROLLO.		16
3.1.	INTRODUCCIÓN.	16
3.2.	CINEMÁTICA ROBÓTICA.	16
3.2.1.	ARTICULACIONES.	16
3.2.2.	REPRESENTACIÓN DE TRANSFORMACIONES.	18
3.2.3.	MATRICES DE ROTACIÓN.	19
3.2.4.	MATRIZ DE TRANSFORMACIÓN HOMOGÉNEA.	22
3.2.5.	PARÁMETROS DENAVIT-HARTENBERG.	23
3.2.6.	LINKS, JOINTS Y SUS PARÁMETROS.	24
3.2.7.	ALGORITMO DENAVIT-HARTENBERG.	25
3.2.8.	CINEMÁTICA DIRECTA.	26
3.2.9.	CINEMÁTICA INVERSA.	28
3.3.	CARACTERÍSTICAS PRINCIPALES DEL ROBOT.	28
3.3.1.	DESCRIPCIÓN Y CARACTERÍSTICAS DEL ROBOT “RVM1”.	28
3.3.2.	ESPACIO DE TRABAJO.	29
3.3.3.	SOLUCIÓN DE CINEMÁTICA DIRECTA.	30
3.3.4.	SOLUCION DE CINEMÁTICA INVERSA.	34
3.4.	SIMULACIÓN Y RESULTADOS.	44
3.4.1	FUNCIÓN LOADDATA.	44
3.4.2.	FUNCIÓN INITHOME.	45
3.4.3.	FUNCIÓN PARA EL CÁLCULO CINEMÁTICA DIRECTA.	48
3.4.4.	DESCRIPCIÓN DE LOS BOTONES DE CINEMÁTICA DIRECTA.	49
3.4.5.	PANEL DE POSICIÓN CINEMÁTICA DIRECTA.	50
3.4.6.	PANEL CINEMÁTICA INVERSA.	51
3.4.7.	FUNCIÓN DE CINEMÁTICA INVERSA RVM1_IK3.	51
3.4.8.	BOTONES DE PROPÓSITO GENERAL.	52
CAPÍTULO 4: CONCLUSIONES.		56
4.1	CONCLUSIONES.	56
4.2	TRABAJO A FUTURO.	56

ÍNDICE DE ILUSTRACIONES

Ilustración 2. 1 Software COSIMIR.	8
Ilustración 2. 2 Simulación en COSIMIR.	8
Ilustración 2. 3 Movimiento por articulación.	9
Ilustración 2. 4 Control de pinza de COSIMIR.	10
Ilustración 2. 5 Área de trabajo para rutinas en COSIMIR.	10
Ilustración 2. 6 Menú opciones de programación en COSIMIR.	11
Ilustración 2. 7 Lectura de ficheros en COSIMIR.	11
Ilustración 2. 8 Menú de lectura de ficheros en COSIMIR.	11
Ilustración 2. 9 Menú de manipulación en COSIMIR.	12
Ilustración 2. 10 Menú para crear archivo en COSIMIR.	12
Ilustración 2. 11 Consola de salida de ejecución del fichero en COSIMIR.	13
Ilustración 2. 12 Metodología de desarrollo de software.	13
Ilustración 3. 1 Relación cinemáticas.	17
Ilustración 3. 2 Vector en el espacio.	17
Ilustración 3. 3 Marco de referencia fijo.	18
Ilustración 3. 4 Sistema de coordenadas.	19
Ilustración 3. 5 Sistema de dos coordenadas.	20
Ilustración 3. 6 Producto escalar.	20
Ilustración 3. 7 Representación matricial.	20
Ilustración 3. 8 Matriz de rotación.	21
Ilustración 3. 9 Composición matriz homogénea.	22
Ilustración 3. 10 Representación de un vector.	22
Ilustración 3. 11 Matriz de rotación ampliada.	22
Ilustración 3. 12 Matriz de traslación homogénea.	23
Ilustración 3. 13 Matriz homogénea.	23
Ilustración 3. 14 Representación de una articulación y un enlace.	24
Ilustración 3. 15 Representación de sistemas.	27
Ilustración 3. 16 Representación y tabla denavit-hartenberg.	27
Ilustración 3. 17 Robot Mitsubishi RV-M1.	28
Ilustración 3. 18 Espacio de operación.	30

Ilustración 3. 19 Parámetros RV-M1.....	30
Ilustración 3. 20 Medidas del robot.....	31
Ilustración 3. 21 Robot RV-M1.....	32
Ilustración 3. 22 Medidas de referencia del robot.....	32
Ilustración 3. 23 Función loaddata.....	45
Ilustración 3. 24 Función inithome.....	46
Ilustración 3. 25 Resultados de las funciones loaddata e inithome.....	46
Ilustración 3. 26 Teaching box.....	47
Ilustración 3. 27 Carga final del programa.....	48
Ilustración 3. 28 Función rvm1_ani.....	49
Ilustración 3. 29 Función para creación de botones.....	49
Ilustración 3. 30 Botones barra deslizamiento.....	50
Ilustración 3. 31 Panel de posición cinemática directa.....	50
Ilustración 3. 32 Código para el panel de posición cinemática directa.....	50
Ilustración 3. 33 Panel cinemática inversa.....	51
Ilustración 3. 34 Código panel cinemática inversa.....	51
Ilustración 3. 35 Código cinemática inversa.....	52
Ilustración 3. 36 Código para el movimiento aleatorio.....	53
Ilustración 3. 37 Código para el posicionamiento en home.....	53
Ilustración 3. 38 Código para botones de teaching box.....	55
Ilustración 3. 39 Botones funcionales.....	55

ÍNDICE DE TABLAS

Tabla 3.1 Movimiento de los ejes.....	29
Tabla 3.2 Características generales.....	29
Tabla 3.3 Parámetros denavit-hartenberg.....	31

CAPITULO 1: INTRODUCCIÓN.

1.1. ANTECEDENTES.

La historia de la automatización industrial está caracterizada por periodos de cambios bruscos en los métodos populares. Ya sea como causa o, tal vez, como un efecto, dichos periodos de cambio en las técnicas de automatización parecen estar estrechamente ligados con la economía mundial. El uso del robot industrial, que se identificó como dispositivo único en la década de 1960, junto con los sistemas de diseño asistido por computadora (CAD) y manufactura asistida por computadora, caracteriza las tendencias más recientes en la automatización del proceso de manufactura. Estas tecnologías están llevando a la automatización industrial hacia otra transición, cuyo alcance se desconoce aún (Craig, 2006).

Una de las principales razones del crecimiento en el uso de robots industriales es la reducción en el costo. Durante la década de 1990, los precios de los robots disminuyeron al mismo tiempo que aumentaron los costos de mano de obra humana. Además, los robots no sólo se están volviendo más baratos, sino también más efectivos: más rápidos, precisos y flexibles. Si se factorizan en los números estos ajustes de calidad, el costo por utilizar robots disminuye mucho más rápido que su etiqueta de precio. A medida que los robots se vuelvan más efectivos en cuanto al costo por sus tareas, y a medida que aumente el costo de la mano de obra humana, habrá más trabajos que se conviertan en candidatos para la automatización robótica. Ésta es la tendencia individual más importante que impulsa el crecimiento del mercado de los robots. Una tendencia secundaria es que, haciendo a un lado la economía, a medida que los robots se vuelvan más capaces, podrán hacer más y más tareas que serían peligrosas o imposibles de realizar por los trabajadores humanos. Las aplicaciones que llevan a cabo los robots industriales se están volviendo cada vez más sofisticadas pero aún se da el caso de que, como en el año 2000, aproximadamente el 78% de los robots instalados en los EE.UU. fueron robots para soldadura o manejo de materiales (Craig, 2006).

1.2. PLANTEAMIENTO DEL PROBLEMA.

El Instituto Tecnológico de Mérida cuenta con un brazo robótico de tipo antropomórfico de cinco grados de libertad (RV-M1 Mitsubishi), el cual actualmente se encuentra fuera de operación e inhabilitado, carece de software especializado de interfaz gráfica de fábrica o hecho a la medida para su control y/o manipulación de rutinas de movimiento ejecutadas por comandos o de rutinas complejas de trabajo programadas para

elaborar movimientos específicos. Se puede agregar que actualmente no existe ninguna base de conocimientos, documentación o investigación con respecto a la manipulación y programación del brazo RV-M1 Mitsubishi ni para su uso industrial o educativo. Su estatus actual es inoperable y obsoleto.

1.3. OBJETIVOS.

1.3.1. OBJETIVO GENERAL.

Desarrollar un simulador en Matlab de un brazo robot de cinco grados de libertad que emule la cinemática directa y cinemática inversa del brazo robot RV-M1 Mitsubishi.

1.3.2. OBJETIVOS ESPECÍFICOS.

- Investigar las tecnologías, materias y disciplinas relacionadas para el desarrollo y control del brazo.
- Desarrollar la interfaz gráfica.
- Desarrollar la cinemática directa.
- Desarrollar la cinemática inversa.
- Realizar pruebas de funcionalidad y verificación.

1.4. ALCANCES DEL PROYECTO.

- Desarrollar una interfaz gráfica.
- Implementar botones para manipulación y simulación de la cinemática directa.
- Implementar cajas de texto para la manipulación y simulación de la cinemática inversa.
- Implementar la cinemática directa.
- Implementación de cinemática inversa.
- Implementar un panel de impresión de resultados.

1.5. JUSTIFICACIÓN.

Con este trabajo se busca desarrollar una plataforma de un robot industrial, por una parte con fines académicos, y por otra apuntando a constituirse como apoyo a las áreas involucradas.

CAPÍTULO 2: MARCO TEÓRICO.

2.1. INTRODUCCIÓN.

Dentro de la categoría de Robots, se destacan los Brazos Robóticos ya que a nivel tecnológico son los proyectos más utilizados debido a sus funcionabilidades y practicidad en sus procesos de manufactura. A través de los años se han incrementado a nivel mundial principalmente debido al nivel de trabajo y tiempo de producción que alcanzan, permitiendo al ser humano preocuparse de labores con más raciocinio y de gestionamiento de la producción o proyección de ventas; debido a estas razones se tiene una amplia gama de categorías y funcionalidades a nivel mundial, los cuales se presentarán más adelante.

2.2. ANTECEDENTES.

Los brazos robóticos se han estado desarrollando desde la década de los años sesentas y se utilizan ampliamente en las fábricas industriales las cuales automatizan generalmente procesos de soldadura, pintura, montaje, etc. Hoy en día, los brazos robóticos son indispensables para la automatización de fábricas. Sin embargo, sus aplicaciones no solamente se limitan a la industria, por lo contrario, se han expandido a otras áreas de uso de la vida cotidiana. Los brazos robóticos son una tecnología integrada que consta de diversos elementos tecnológicos como actuadores, sensores, mecanismos y sistemas de control, etc.

2.3. BRAZO ROBÓTICO.

Un brazo robótico es un robot con forma de un brazo humano, diseñado para cumplir funciones repetitivas en las cuales un brazo normal desempeñaría con cierta dificultad.

2.4. CONCEPTOS BÁSICOS.

2.4.1. CONTROLADOR.

Registra cada uno de los movimientos del manipulador, las operaciones, cálculos y procesado de la codificación. El controlador recibe y envía señales a otros procesos (por medio de señales de entrada/salida) y almacena los programas. Se registran varios grados de control que están en función del tipo de cuantificaciones que se regulan a partir de una variable o petición del ordenador. A continuación se presentan algunos tipos de controladores:

- De posición: el controlador ejecuta las acciones cuando el control determina la posición final del elemento.
- Cinemático: el control se determina considerando la posición y la velocidad.

- Dinámico: regula la velocidad, la posición y además controla las características solicitadas del manipulador y los elementos vinculados a él.
- Adaptativo: determina todas las medidas anteriores y además, permite controlar la variación de las propiedades del manipulador al cambiar de posición (PEREZ, 2010).

2.4.2. GRADOS DE LIBERTAD (GDL).

Considerando cada uno de los movimientos autónomos (torques y desplazamientos) que determina una articulación en función de la anterior, esto determina las variables que se necesitan para establecer la posición y la dirección del mecanismo final del manipulador. El valor de grados de libertad (GDL) del robot se determina por el conteo de los GDL de las juntas que lo conforman. Debido a que las articulaciones usadas suelen ser exclusivamente de traslación y prismáticas, con solo un grado de libertad cada una, el número de GDL del robot se determina en relación al número total de articulaciones. Para ubicar y dar dirección a un cuerpo en cualquier lugar en el entorno de trabajo son requeridas seis variables, tres para relacionar la posición y tres para la dirección; si se desea que un robot posicione y oriente su efector final en cualquier dirección en el espacio, precisará de seis grados de libertad o más (PEREZ, 2010).

2.4.3. ESPACIO DE TRABAJO.

Las proporciones de las articulaciones del robot, junto a los grados de libertad, precisan el entorno de trabajo del robot, las cuales definen las propiedades principales en las etapas de clasificación y creación del patrón apropiado. El espacio de trabajo se puede establecer en diferentes superficies, determinados por el alcance característico del efector final, es diferente a la que permite colocar verticalmente o con un determinado sentido de movimiento (PEREZ, 2010).

2.4.4. EXACTITUD DE LOS MOVIMIENTOS.

La precisión en un robot industrial obedece algunos elementos que se deben considerar de la siguiente manera: La resolución del entorno de trabajo, se puntualiza como la extensión del movimiento en la que el robot puede dividir la cantidad de trabajo. Depende también de dos factores: del control de los sistemas que definen la resolución y los errores mecánicos.

Se define como de control, ya que éste, es el entorno para reconocer todos los aumentos en los movimientos de cada articulación. La controladora que genera la segmentación del intervalo total del movimiento para una junta específica en aumentos independientes (control de mando). La flexibilidad de dividir el rango de la junta en diferentes lapsos, obedece a la capacidad de almacenamiento en la memoria del dispositivo. La cantidad de incrementos deducibles para un eje específico es: $2n$. Si se tiene, un autómeta con un valor de incremento en $n = 8$ el controlador puede fraccionar el momento de la transición en 256 estados discretos. De esta manera la resolución del control son intervalos de número (transiciones o movimientos)/256. Los aumentos casi siempre son constantes.

Las fallas mecánicas se encuentran altamente afectadas por la disposición de los materiales que integran las juntas y las articulaciones. Como ejemplos de fallas mecánicas se puede encontrar la amplitud de los engranajes, la rigidez en los motores, los escapes de fluidos, etc.

La precisión se refiere al volumen de un robot para ubicar el extremo de su articulación en el lugar indicado dentro del entorno de trabajo la cual determina el desplazamiento entre el estado especificado y la colocación real de la articulación final del manipulador.

Sostiene una correspondencia directa con el valor espacial, es decir, con el volumen de control del manipulador de fraccionar el volumen de trabajo. El manipulador se puede ubicar en el punto de retorno (PR) o en un punto de retorno que posea una distancia equivalente hacia el Primer Punto (PP). Dependiendo del trabajo que se vaya a ejecutar, es la exactitud en la recursividad de los desplazamientos sean estos mayor o menor al proceso inicial, por ejemplo, en trabajos de ensamblaje de segmentos, dicha propiedad será bajo 0.1 ms. En soldadura, pintura y manipulación de fragmentos, la igualdad en la repetición se entiende entre 1 y 3 mm y en los procedimientos de mecanizado, una precisión bajo el 1mm. La repetición de un punto es usualmente menor que la precisión (PEREZ, 2010) (Bonilla, 2013).

2.5. TIPOS DE BRAZOS ROBÓTICOS.

Existe una amplia clasificación de los brazos robóticos, de los cuales se mencionan a continuación:

2.5.1. SCARA.

El uso de algunas disposiciones en las partes del robot, nota diferentes clasificaciones, con particularidades a tener en cuenta en función del diseño y estructura del robot como en su uso. Las composiciones más notorias son con tres articulaciones, siendo las más importantes al tomar en cuenta el posicionamiento de su extremo en un lugar en el área de trabajo (Bonilla, 2013).

2.5.2. CARTESIANO.

La ubicación se hace en el lugar de trabajo con el brazo establecido. Esta configuración se utiliza cuando el lugar de labor es extenso y tiene que ser cubierto, o debido a la precisión que se fundamenta en la perspectiva del robot. Contiene tres desplazamientos lineales, de los tres GDL, que tienen correspondencia con los desplazamientos focalizados en los ejes X, Y y Z. Los desplazamientos que genera este robot tomando desde un punto a otro, tienen su fundamento en sus composiciones lineales. Interpolación, se define como, el tipo de recorrido que ejecuta el manipulador en función del desplazamiento entre un sitio y otro. Al desplazamiento realizado en un traslado rectilíneo se lo define como inserción lineal y teniendo un recorrido realizado en relación al tipo de trayectoria que poseen sus articulaciones se conoce como interpolación por junta (Bonilla, 2013).

2.5.3. CILÍNDRICOS.

El robot posee un desplazamiento de giro en su propio eje, una articulación helicoidal para la altura, y una prismática para soporte del radio. El robot se ajusta bien a los lugares de trabajo cónico o cilíndrico. Logra ejecutar los movimientos, rectilíneo y rotacional, o sea, que exhibe tres GDL. El robot fue construido para elaborar trayectorias definidas como lineales y por articulación. La definida por articulación está en función de la primera junta, debido a que puede realizar un traslado rotacional.

2.5.4. ESFÉRICO.

Conocidos los robots esféricos capaces de desplazarse según diferentes trayectorias, pero suelen adolecer, en muchas ocasiones, de determinados inconvenientes, tales como que su movimiento no es continuo sino oscilatorio, complejo funcionamiento y control o necesidad de una precisión de fabricación del interior de la esfera que encarecerían su fabricación. Así, la presente invención se refiere a un robot esférico, en el que los medios de

accionamiento son de constitución y funcionamiento sencillos, con un número reducido de componentes y que además permiten un desplazamiento tanto rectilíneo como curvilíneo, es decir, es capaz de desplazarse en diferentes sentidos y según trayectorias distintas, estando constituido por una carcasa esférica en la que se alojan medios de accionamiento que actúan sobre la superficie interna de dicha carcasa para provocar su desplazamiento. Además, este robot esférico asegura una actuación correcta de los medios de accionamiento, aunque la superficie interna de la carcasa esférica presente irregularidades (Bonilla, 2013).

2.6. SOFTWARE PARA MITSUBISHI RV-M1.

No es raro oír sobre la implementación de robots en las diversas aplicaciones en la industria mundial. Existe una gran variedad de modelos y sistemas robotizados como también una gran cantidad de empresas que los proveen. Estas empresas tienen a su disposición software de modelamiento que les permite simular la manera en que una aplicación robotizada funcionará dentro de una planta de producción automatizada, para facilitar de esta manera la tarea de toma de decisiones y elegir o no la adecuación o creación de un sistema robotizado. Para tal efecto, el software de simulación permite mostrar de manera gráfica y simulada como quedará un sistema de producción, incluyendo obviamente dentro de líneas de producción brazos robóticos que ejecutarán operaciones de ensamble, soldadura, pintura, moldeado, transferencia de materiales, clasificación de piezas, etc. Es importante mencionar que en internet hay posibilidad de descargar, a modo de evaluación y de manera gratuita, software de simulación y modelamiento de robots.

2.6.1. COSIMIR.

Una de las tantas aplicaciones que permiten este tipo de simulaciones es el software COSIMIR Educational con el que se puede modelar, diseñar y simular sistemas de producción robotizados.

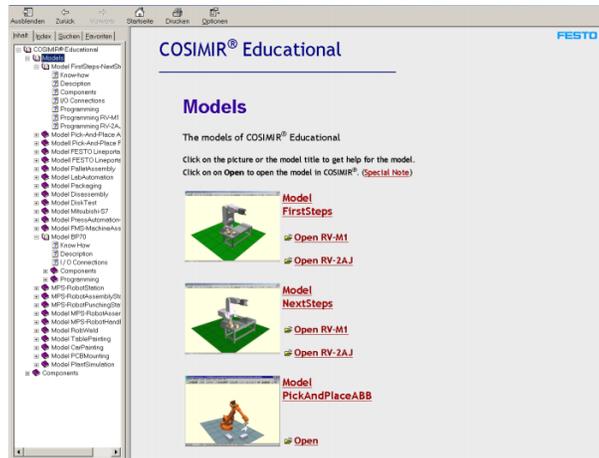


Ilustración 2. 1 Software COSIMIR.

Este software está disponible en la web para que toda persona interesada en el tema lo pueda evaluar por un determinado periodo de tiempo. Posee una amigable interface de diseño y programación que permite un fácil manejo. Asimismo, cuenta con una librería de elementos que contiene modelos de robots, pinzas, herramientas, alimentadores neumáticos, sensores, PLC (controlador lógico programable), al abrir el software aparecen varias ventanas, entre las principales están la ventana de modelamiento, de posiciones y de programación (FESTO COSIMIR).

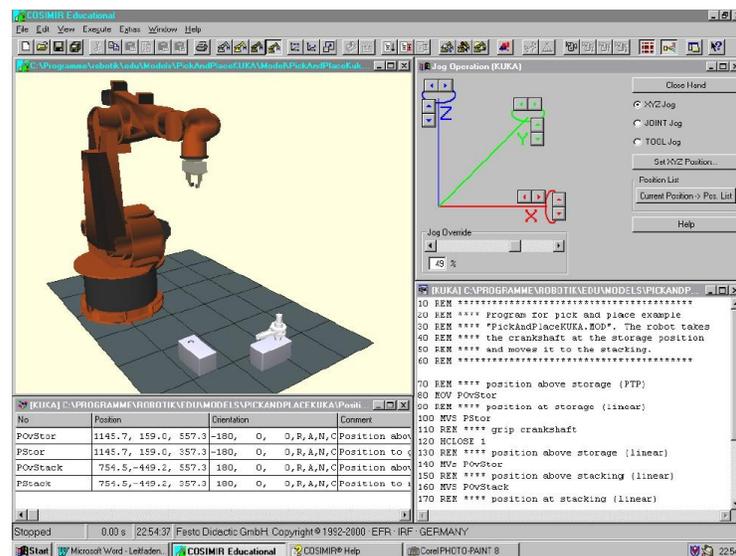


Ilustración 2. 2 Simulación en COSIMIR.

Los siguientes lenguajes de programación se pueden utilizar en COSIMIR® para la programación de robots:

- MELFA IV lenguaje de programación Basic robot Mitsubishi Para modelos RV-M1 y RV-M2.
- MRL lenguaje de programación de robots Mitsubishi.
- Idioma de robot industrial estandarizada (IRL DIN 66312).

Todos los robots de Mitsubishi han sido programados con MELFA Básico IV, como siempre que esta lengua se apoya en el controlador correspondiente. MRL ha sido utilizado para el resto de los robots de Mitsubishi. Otros tipos de robots han sido programados utilizando IRL. La sección "Programación del robot" del Asistente de Robótica incluye información completa sobre la programación de robots. Los detalles relativos a los lenguajes de programación están contenidos en el capítulo en el tema "Lenguajes de programación" en la ayuda del sistema.

Con el fin de crear un programa para el robot, ciertas posiciones deben ser definidas para que el robot se desplace bajo ciertas condiciones. En términos generales, un robot puede hacerse avanzar con la ayuda de un control de accionamiento manual. COSIMIR proporciona a los usuarios dos métodos diferentes para hacer avanzar el robot manualmente:

- Con el mouse.
- Con la ventana TEACH-IN.

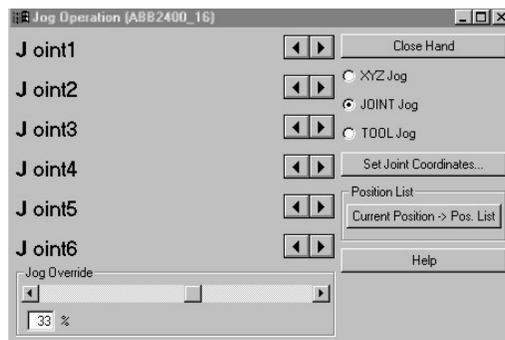


Ilustración 2. 3 Movimiento por articulación.

El robot puede moverse a lo largo de los ejes de coordenadas, y la pinza puede girar en torno a estos ejes pulsando los botones correspondientes. TEACH-IN PINZA CONTROL ofrece la capacidad para configurar las tareas de agarre de la pinza de agarre; también la capacidad de cerrar y abrir la pinza de agarre manualmente y poder configurar advertencias para las opciones de agarre de la pinza (zur, 2003).



Ilustración 2. 4 Control de pinza de COSIMIR.

Para el desarrollo de programas y/o rutinas de trabajo más elaboradas o complejas cuenta con el panel de trabajo que facilita la creación y administración de los proyectos de programación.

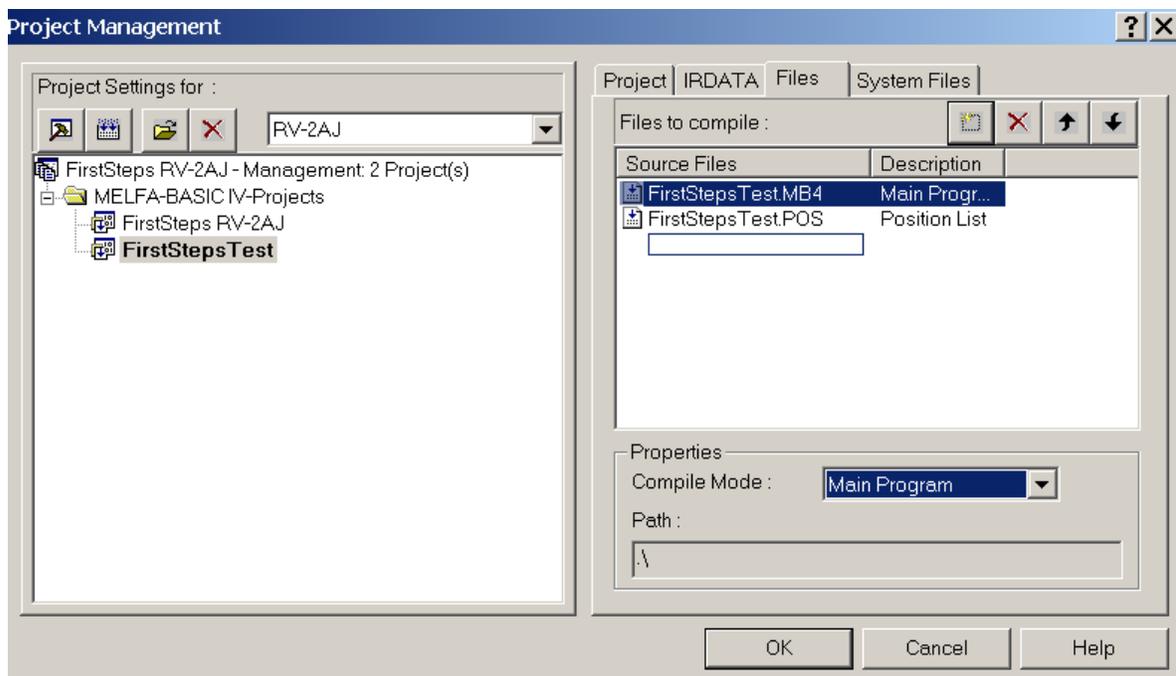


Ilustración 2. 5 Área de trabajo para rutinas en COSIMIR.

Adicionalmente, todos los programas creados pueden ser descargados a los robots vía COSIMIR seleccionando el modelo y la vía de comunicación, con la opción de ser ejecutados una vez finalizada la descarga en el robot.

También se puede programar mediante BASIC y enviar los archivos mediante el puerto de impresora (FESTO COSIMIR).

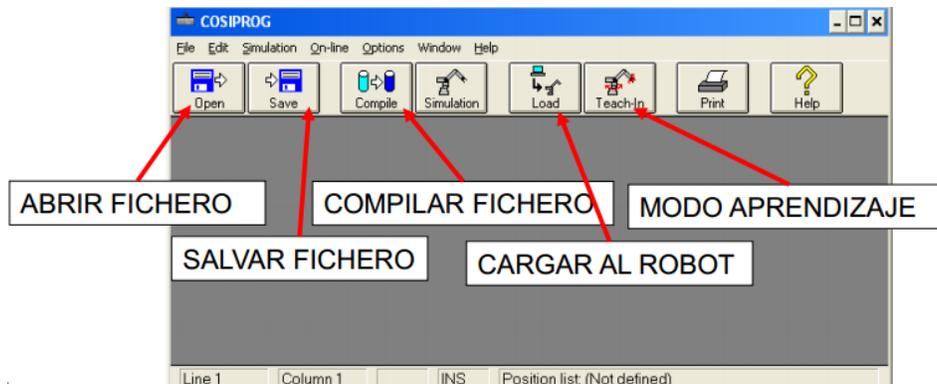


Ilustración 2. 6 Menú opciones de programación en COSIMIR.

Para realizar el programa del robot se debe tener en cuenta que se deben crear dos ficheros:

- Fichero de posiciones (extensión .POS).
- Fichero con el programa (extensión .MR.)

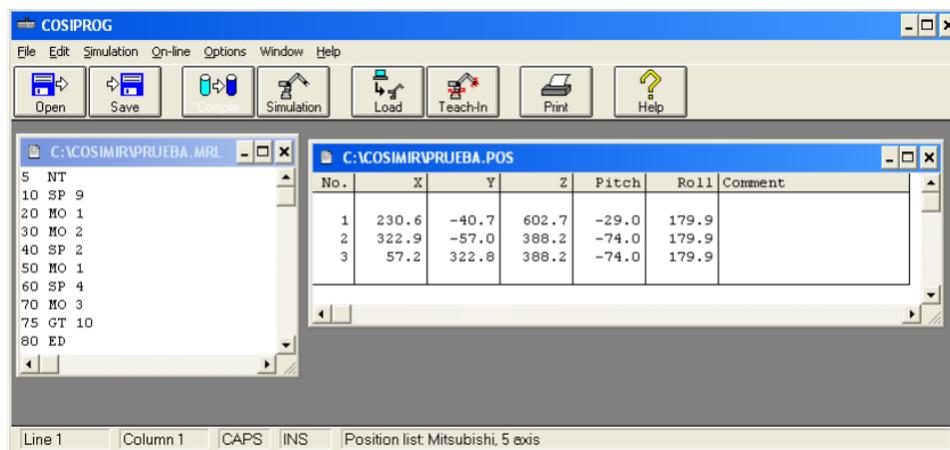


Ilustración 2. 7 Lectura de ficheros en COSIMIR.

Se pueden crear los ficheros de posiciones de la siguiente manera:

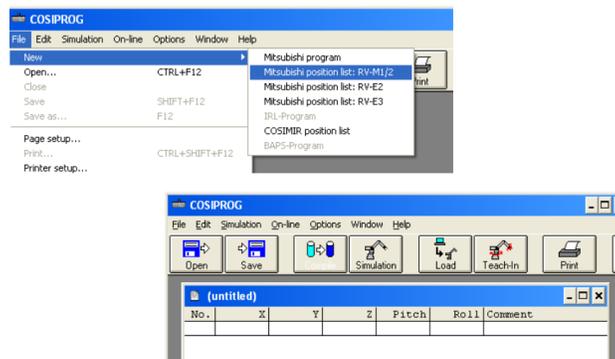


Ilustración 2. 8 Menú de lectura de ficheros en COSIMIR.

Contiene un panel para poder introducir las distintas posiciones introduciendo el número correspondiente y sus coordenadas. El proceso es más sencillo si se utiliza la opción Teach-In, teniendo conectado el robot. De este modo se puede guiar a las posiciones deseadas y después introducirlas en un archivo con el número que se desee (FESTO COSIMIR).

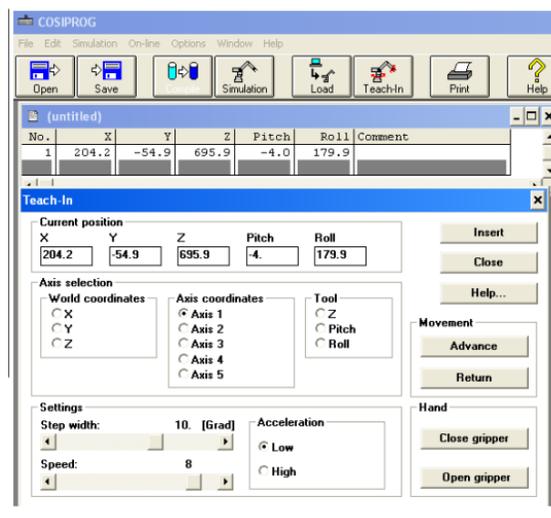


Ilustración 2. 9 Menú de manipulación en COSIMIR.

Una vez creado el fichero de posiciones se debe de introducir el listado del programa en el fichero correspondiente (MRL). Para crear dicho fichero se debe proceder de la siguiente manera:

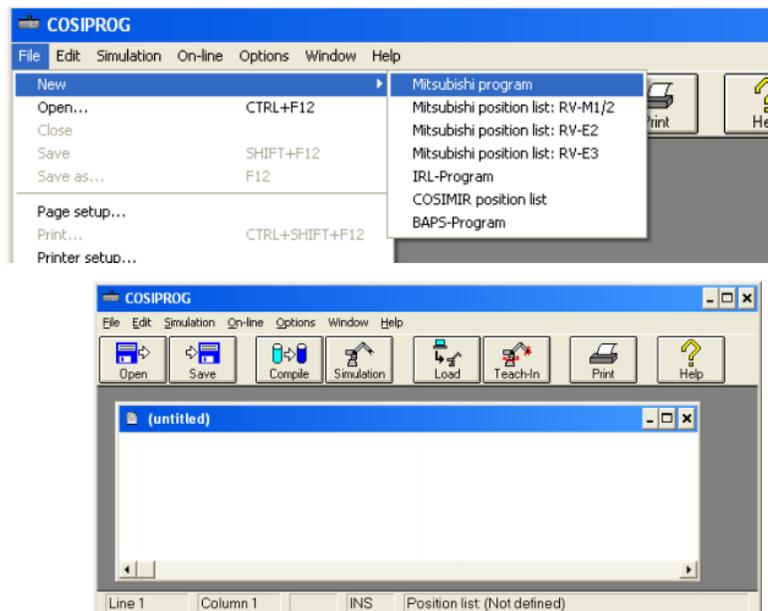
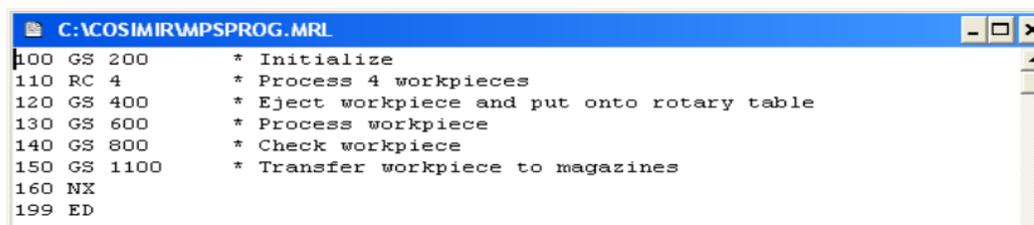


Ilustración 2. 10 Menú para crear archivo en COSIMIR.

Cada línea debe tener una numeración que le servirá de etiqueta. La numeración será progresiva, aunque no necesariamente consecutiva. Por ejemplo, los números de línea pueden ir de 10 en 10. Igualmente, en cada línea se puede introducir un comentario precedido del carácter (FESTO COSIMIR).

Cuando está terminado el programa puede guardar y después compilarlo con la opción correspondiente (COMPILE o File, Compile). Por último, se deben de transferir al robot, los dos ficheros creados. Para ello se selecciona la ventana y se pulsa la opción LOAD (On-Line, Load file). Solamente quedaría ejecutar el programa mediante On-Line, Execute program (FESTO COSIMIR).



```
C:\COSIMIR\MPS\PROG.MRL
100 GS 200 * Initialize
110 RC 4 * Process 4 workpieces
120 GS 400 * Eject workpiece and put onto rotary table
130 GS 600 * Process workpiece
140 GS 800 * Check workpiece
150 GS 1100 * Transfer workpiece to magazines
160 NX
199 ED
```

Ilustración 2. 11 Consola de salida de ejecución del fichero en COSIMIR.

2.7. METODOLOGÍA.

Para determinar las funcionalidades del sistema se considera como premisa la construcción de estructuras que determinen los pasos a seguir y los procedimientos que se deben tomar en cuenta para generar una correcta arquitectura del proyecto presentado a continuación.

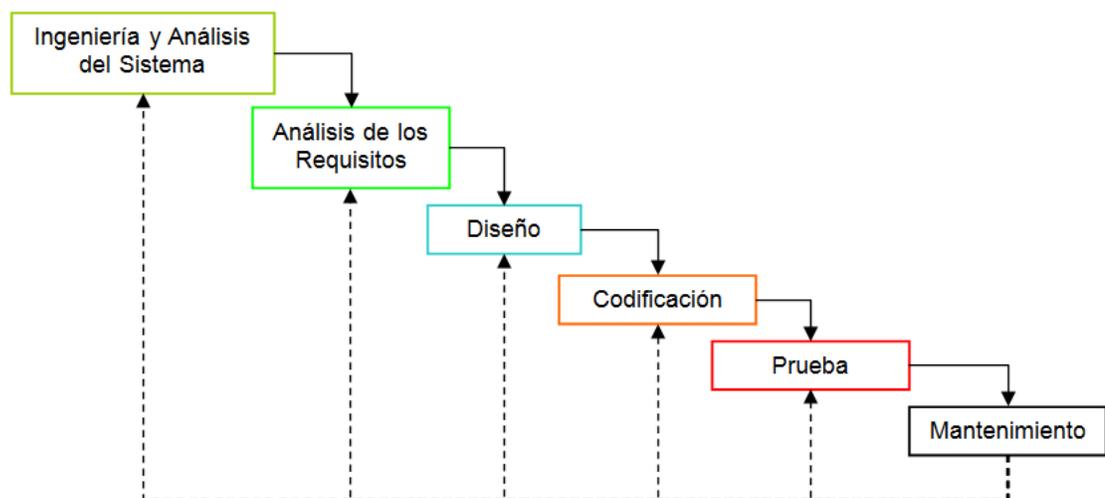


Ilustración 2. 12 Metodología de desarrollo de software.

2.7.1. MODELO EN CASCADA.

El más conocido, está basado en el ciclo convencional de una ingeniería, el paradigma del ciclo de vida abarca las siguientes actividades:

2.7.2. INGENIERÍA Y ANÁLISIS DEL SISTEMA.

Debido a que el software es siempre parte de un sistema mayor el trabajo comienza estableciendo los requisitos de todos los elementos del sistema y luego asignando algún subconjunto de estos requisitos al software.

2.7.3. ANÁLISIS DE LOS REQUISITOS DEL SOFTWARE.

El proceso de recopilación de los requisitos se centra e intensifica especialmente en el software. El ingeniero de software (Analistas) debe comprender el ámbito de la información del software, así como la función, el rendimiento y las interfaces requeridas.

2.7.4. DISEÑO.

El diseño del software se enfoca en cuatro atributos distintos del programa: la estructura de los datos, la arquitectura del software, el detalle procedimental y la caracterización de la interfaz. El proceso de diseño traduce los requisitos en una representación del software con la calidad requerida antes de que comience la codificación.

2.7.5. CODIFICACIÓN.

El diseño debe traducirse en una forma legible para la máquina. El paso de codificación realiza esta tarea. Si el diseño se realiza de una manera detallada la codificación puede realizarse mecánicamente.

2.7.6. PRUEBAS.

Una vez que se ha generado el código comienzan las pruebas del programa. Las pruebas se centran en la lógica interna del software, y en las funciones externas, realizando pruebas que aseguren que la entrada definida produce los resultados que realmente se requieren.

2.7.7. MANTENIMIENTO.

El software sufrirá cambios después de que se entrega al cliente. Los cambios ocurrirán debidos a que hayan encontrado errores, a que el software deba adaptarse a cambios del entorno externo (sistema operativo o dispositivos periféricos), o debido a que el cliente requiera ampliaciones funcionales o del rendimiento.

2.8. HERRAMIENTAS PARA EL DESARROLLO DE LA APLICACIÓN.

Para el desarrollo de este proyecto se utilizaron diversas herramientas de software, las cuales se enlistan a continuación.

- MATLAB.
- SIMULINK

2.8.1. MATLAB.

Es una herramienta de 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, Mac OS X y GNU/Linux. 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). Además, se pueden ampliar las capacidades de MATLAB con las cajas de herramientas (toolboxes); y las de Simulink con los paquetes de bloques (blocksets) (<https://es.mathworks.com>, 2018).

2.8.2. SIMULINK.

Simulink ofrece un editor gráfico, bibliotecas de bloques personalizables y solvers para modelar y simular sistemas dinámicos. Se integra con MATLAB[®], lo que permite incorporar algoritmos de MATLAB en los modelos y exportar los resultados de la simulación a MATLAB para llevar a cabo más análisis (<https://es.mathworks.com>, 2018).

CAPÍTULO 3: DESARROLLO.

3.1. INTRODUCCIÓN.

Para encontrar analíticamente las expresiones de los límites de los manipuladores es necesario lo siguiente:

- Desarrollar un conjunto de criterios analíticos para obtener el posicionamiento de la muñeca en términos de coordenadas generalizadas.
- Determinar la superficie límite debido a las singularidades (singularidad; posición peculiar de ejes que puede impedir la ejecución de movimientos del robot) asociadas a él.
- Determinar el subconjunto de estas superficies debido a los límites de las articulaciones. combinar todas las superficies para desarrollar el espacio de trabajo.

3.2. CINEMÁTICA ROBÓTICA.

3.2.1. ARTICULACIONES.

Un manipulador mecánico puede ser modelado como una cadena articulada con varios cuerpos rígidos (eslabones) conectados en serie por articulaciones revoluta o prismáticas accionadas por actuadores (Goyal, 2002).

El movimiento relativo de las uniones da como resultado el enlace de movimiento que posiciona la mano en una orientación deseada, es decir, se está interesado en la descripción espacial del efector final del manipulador con respecto a un sistema de coordenadas de referencia fijo. La cinemática del brazo del robot se ocupa del estudio analítico de la geometría del movimiento de un brazo del robot con respecto a un sistema de coordenadas de referencia fijo. Por lo tanto, se trata de la descripción analítica del desplazamiento espacial del robot, en particular las relaciones entre el espacio variable de las uniones, su posición y orientación final del efector del brazo robótico. Los dos problemas fundamentales son:

- Para un manipulador dado, dado un vector de ángulo de unión $q(t) = (q_1(t), q_2(t), \dots, q_n(t))^T$ y los parámetros de enlaces geométricos, donde n es el número de grados de libertad, es la posición y orientación del efector final del manipulador con respecto a un sistema de coordenadas de referencia.
- Dada la posición deseada y la orientación del efector final del manipulador y los parámetros del enlace geométrico con respecto a un sistema de coordenadas de

referencia, el manipulador puede alcanzar la posición y orientación de mano del manipulador prescrito deseado. y si puede, cuántas configuraciones diferentes del manipulador satisfarán la misma condición.

El primer problema usualmente se refiere como cinemática directa, mientras que el segundo a la cinemática inversa. Existe una relación entre la cinemática inversa y directa. La salida de una es la entrada a la segunda y viceversa; el diagrama presentado a continuación indica la relación entre las dos.

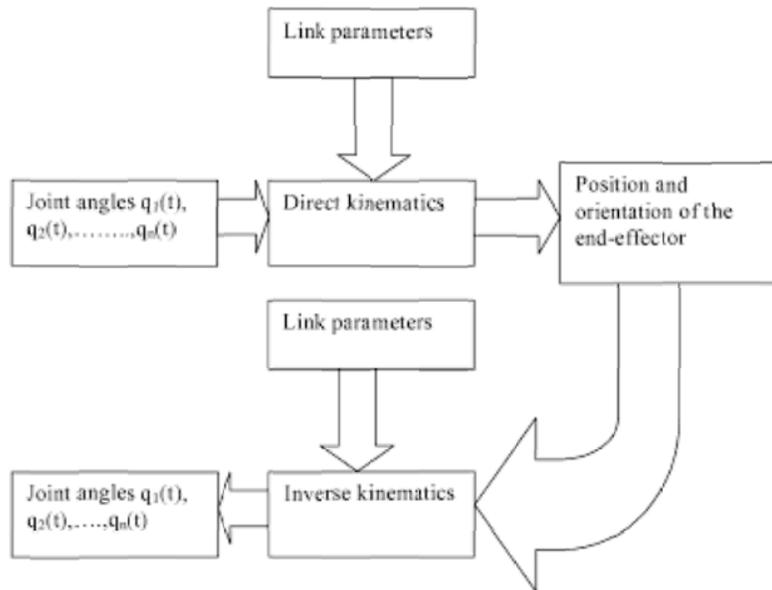


Ilustración 3. 1 Relación cinemáticas.

Vector en el espacio; si el vector comienza en el origen, se describe por la siguiente figura:

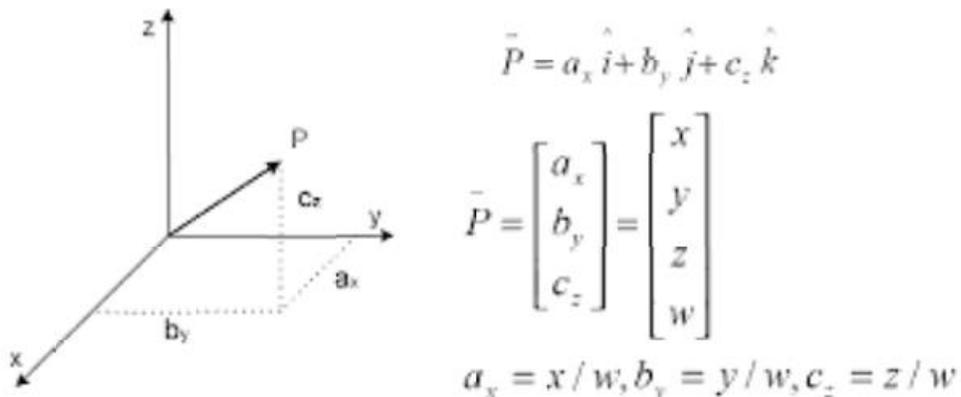


Ilustración 3. 2 Vector en el espacio.

Para marco en el origen de referencia fijo:

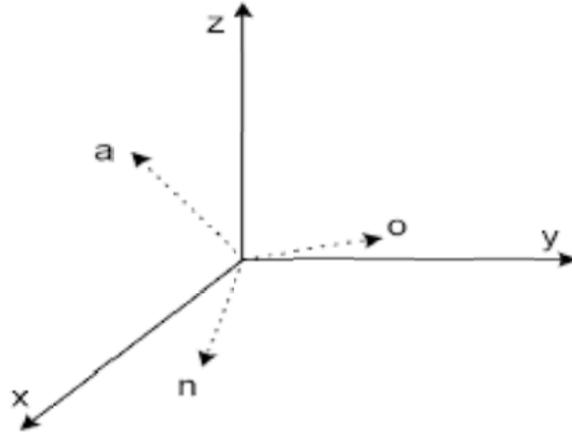


Ilustración 3.3 Marco de referencia fijo.

- Representado por 3 vectores cada uno perpendicular a otro:
- Vector normal “n”.
- Vector de orientación “o”.
- Vector de aproximación “a”.
- Y estos 3 vectores están representados a su vez por la siguiente matriz:

$$F = \begin{matrix} n_x & O_X & a_x \\ n_y & O_Y & a_y \\ n_z & O_Z & a_z \end{matrix} \quad \text{Ecuación 1}$$

3.2.2. REPRESENTACIÓN DE TRANSFORMACIONES.

Una transformación es definida como un movimiento en el espacio. Cuando un marco es trasladado a un espacio relativo a su marco de referencia, se puede representar con una matriz de translación. Cuando un marco es trasladado en el espacio sin ningún cambio en su orientación es llamado translación pura. La nueva ubicación del marco puede encontrarse añadiendo el vector que representa la translación al vector que representa la ubicación original. Las primeras 3 columnas no presentan ningún movimiento de rotación, mientras que la última columna representa la translación. La transformación para realizar una translación de un vector en el espacio por una distancia “dz” en dirección de x, “dy” en dirección de y, y “dx” en dirección de z está dada por:

$$T = \begin{vmatrix} 1 & 0 & 0 & d_x \\ 0 & 1 & 0 & d_y \\ 0 & 0 & 0 & d_z \\ 0 & 0 & 0 & 1 \end{vmatrix} \quad \text{Ecuación 2}$$

3.2.3. MATRICES DE ROTACIÓN.

Para abordar el problema de modelar la orientación se parte de una condición inicial donde los sistemas de referencia, móvil y fijo, tienen plenamente coincidencia entre sus ejes y luego se procede a girar el sistema móvil un ángulo θ , alrededor del eje “z” (Fernandez, 2016).

Se puede definir una matriz de rotación como una matriz de transformación que opera sobre un vector de posición en un espacio tridimensional y mapea sus coordenadas expresadas en un sistema de coordenadas girado a un sistema de coordenadas de referencia, las rotaciones de un vector alrededor de cada uno de los tres ejes con un ángulo, puede realizarse mediante transformaciones de rotación (Goyal, 2002).

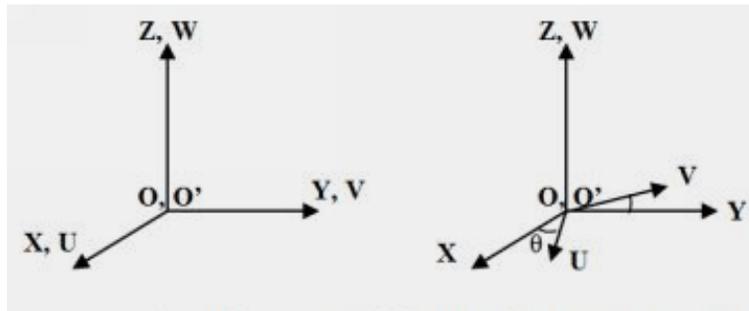


Ilustración 3. 4 Sistema de coordenadas.

En la figura anterior se puede mencionar que la imagen de la izquierda representa la condición inicial de coincidencia entre sistemas de referencia móvil y fijo. Para la imagen de la derecha el giro del sistema de referencia móvil alrededor del eje “Z”. La matriz de rotación indica la orientación y la transformación entre dos marcos de referencia, que a su vez es utilizada para obtener la cinemática directa. En un sistema de dos coordenadas, ambos sistemas con el mismo origen y uno de ellos rotando.

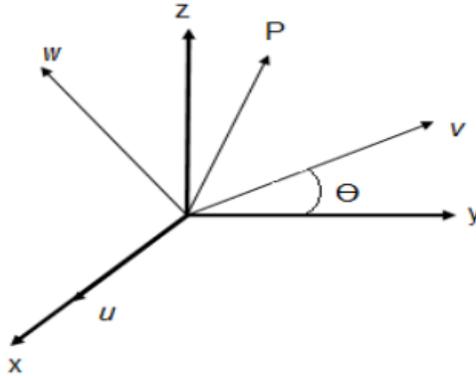


Ilustración 3. 5 Sistema de dos coordenadas.

En la figura anterior se puede observar que el sistema coordenado “UVW” gira un ángulo θ en torno al sistema “XYZ”, tomando como eje de rotación el eje “x”. El vector P, con componentes “u,v,w” en el sistema “UVW” está representado como:

$$P_{uvw} = p_u i_u + p_v i_v + p_w i_w$$

Donde “ p_x ”, “ p_y ” y “ p_z ” representan las componentes de “p” a lo largo del eje “XYZ”, y se efectúa el producto escalar de P con los vectores unitarios en el sistema “XYZ”: “ $i_x + j_y + k_z$ ”. Ahora bien:

$$\begin{aligned} p_x &= i_x \cdot P = i_x \cdot i_u p_u + i_x \cdot j_v p_v + i_x \cdot k_w p_w \\ p_y &= j_y \cdot P = j_y \cdot i_u p_u + j_y \cdot j_v p_v + j_y \cdot k_w p_w \\ p_z &= k_z \cdot P = k_z \cdot i_u p_u + k_z \cdot j_v p_v + k_z \cdot k_w p_w \end{aligned}$$

$$\begin{aligned} p_x &= p_u \\ p_y &= p_v \cos \theta - p_w \sin \theta \\ p_z &= p_v \sin \theta + p_w \cos \theta \end{aligned}$$

Ilustración 3. 6 Producto escalar.

Y su representación en forma matricial se presenta en la siguiente figura:

$$\begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix} = \begin{bmatrix} i_x \cdot i_u & i_x \cdot j_v & i_x \cdot k_w \\ j_y \cdot i_u & j_y \cdot j_v & j_y \cdot k_w \\ k_z \cdot i_u & k_z \cdot j_v & k_z \cdot k_w \end{bmatrix} \begin{bmatrix} p_u \\ p_v \\ p_w \end{bmatrix}$$

Ilustración 3. 7 Representación matricial.

La matriz de rotación estará dada por:

$$R = \begin{bmatrix} i_x \cdot i_u & i_x \cdot j_v & i_x \cdot k_w \\ j_y \cdot i_u & j_y \cdot j_v & j_y \cdot k_w \\ k_z \cdot i_u & k_z \cdot j_v & k_z \cdot k_w \end{bmatrix}$$

Ilustración 3. 8 Matriz de rotación.

Como los productos escalares son conmutativos, esto significa que la inversa de una matriz de rotación es igual a su transpuesta (Cortés, 2012):

$$Q = R^{-1} = R^T \quad \text{Ecuación 3}$$

$$QR = R R = R^{-1}R = I_3 \quad \text{Ecuación 4}$$

Donde I_3 es la matriz de identidad. Ahora se obtiene las matrices de rotación sobre los ejes “xyz”:

Rotación respecto al eje x con un ángulo θ :

$$Rot(x, \theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & C\theta & -S\theta \\ 0 & S\theta & C\theta \end{bmatrix} \quad \text{Ecuación 5}$$

Rotación respecto al eje y con el ángulo θ :

$$Rot(y, \theta) = \begin{bmatrix} C\theta & 0 & S\theta \\ 0 & 1 & 0 \\ -S\theta & 0 & C\theta \end{bmatrix} \quad \text{Ecuación 6}$$

Rotación respecto al eje z con un ángulo θ :

$$Rot(z, \theta) = \begin{bmatrix} C\theta & -S\theta & 0 \\ S\theta & C\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \text{Ecuación 7}$$

La matriz de rotación que se representa una rotación en el ángulo α con respecto al eje x seguida de una rotación del ángulo θ respecto al eje z , y por ultimo una rotación del ángulo Φ respecto “a” y la matriz de rotación resultan:

$$R = Rot(y, \Phi)I_3Rot(w, \theta)Rot(u, \alpha) \quad \text{Ecuación 8}$$

$$= \begin{bmatrix} C\Phi & 0 & S\Phi \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} C\theta & -S\theta & 0 \\ S\theta & C\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & C\alpha & -S\alpha \\ 0 & S\alpha & C\alpha \end{bmatrix} \quad \text{Ecuación 9}$$

$$= \begin{bmatrix} C\phi C\theta & S\phi S\alpha - C\phi S\theta C\alpha & C\phi S\theta S\alpha + S\phi C\alpha \\ S\theta & C\theta C\alpha & -C\theta S\alpha \\ S\phi C\theta & S\phi S\theta C\alpha + C\phi S\alpha & C\phi C\alpha - S\phi S\theta S\alpha \end{bmatrix}$$

Ecuación 10

Los sistemas de coordenadas son incidentes, de aquí la matriz de rotación es una matriz de identidad I3 de tres por tres. (Bonilla, 2013)

3.2.4. MATRIZ DE TRANSFORMACIÓN HOMOGÉNEA.

Las matrices de rotación homogénea y una matriz traslación homogénea se pueden multiplicar juntas para obtener una matriz de transformación homogénea compuesta, ya que la multiplicación de la matriz no es conmutativa. Se debe prestar mucha atención al orden en que se multiplican estas matrices. La matriz de transformación homogénea es utilizada para explicar la relación geométrica que existe entre un sistema “uvw” y un sistema de coordenadas de referencia “xyz”. Esta nueva matriz realmente incluye una sub matriz para describir la orientación (matriz de rotación), la posición (vector de traslación), la perspectiva (vector nulo, por simplicidad) y el factor de escalado (igual a 1).

$$T = \begin{bmatrix} \text{Rotación}_{3 \times 3} & \text{Traslación}_{3 \times 1} \\ \text{Perspectiva}_{1 \times 3} & \text{Escala}_{1 \times 1} \end{bmatrix} = \begin{bmatrix} R_{3 \times 3} & P_{3 \times 1} \\ 0_{1 \times 3} & 1 \end{bmatrix}$$

Ilustración 3. 9 Composición matriz homogénea.

Un vector “p” en un espacio tridimensional se expresa en coordenadas homogéneas de la siguiente manera:

$$\hat{p} = \left[(p_x, p_y, p_z, 1)^T \right]$$

Ilustración 3. 10 Representación de un vector.

Puede ser ampliada una matriz de rotación homogénea para operaciones de rotación como se muestra en la siguiente figura:

$$T_{x,\alpha} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & C\theta & -S\theta & 0 \\ 0 & S\theta & C\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad T_{y,\phi} = \begin{bmatrix} C\phi & 0 & S\phi & 0 \\ 0 & 1 & 0 & 0 \\ -S\phi & 0 & C\phi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad T_{z,\theta} = \begin{bmatrix} C\theta & -C\theta & 0 & 0 \\ S\theta & C\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Ilustración 3. 11 Matriz de rotación ampliada.

La posición del origen de uvw, con respecto al de xyz en forma de matriz homogénea queda de la siguiente manera:

$$T_{\text{trans}} = \begin{bmatrix} 1 & 0 & 0 & dx \\ 0 & 1 & 0 & dy \\ 0 & 0 & 1 & dz \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Ilustración 3. 12 Matriz de traslación homogénea.

La matriz total debe pre multiplicar a la matriz que representa la rotación, por la matriz que representa la traslación, como se muestra en la siguiente figura:

$$F = \begin{bmatrix} n_x & s_x & a_x & p_x \\ n_y & s_y & a_y & p_y \\ n_z & s_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} R_{3 \times 3} & P_{3 \times 1} \\ 0 & 1 \end{bmatrix}$$

Ilustración 3. 13 Matriz homogénea.

Donde s es la orientación, a el vector de aproximación, y p el vector de posición de la herramienta. Finalmente el vector n está definido como:

$$n = s \times a \quad \text{Ecuación 11}$$

3.2.5. PARÁMETROS DENAVIT-HARTENBERG.

La representación Denavit-Hartenberg establece que al seleccionarse adecuadamente los sistemas de coordenadas asociados, a cada eslabón, será posible pasar de uno al siguiente mediante cuatro transformaciones básicas que dependen exclusivamente de las características geométricas del eslabón. El patrón de transformaciones para relacionar el sistema de referencia del elemento “i” con respecto al sistema del elemento “i-1”:

- Rotación alrededor del eje Zi-1 un ángulo θ_{i-1} .
- Traslación a lo largo del eje Zi-1 una distancia d_{i-1} .
- Traslación a lo largo del eje Xi a una distancia a_{i-1} .
- Rotación alrededor del eje Xi un ángulo α_{i-1} .

Dónde:

- “a” representa la distancia normal a los ejes de las articulaciones.
- “ α ” representa un ángulo de torsión.

- “d” es la distancia entre las dos rectas normales a_{i-1} y a_i en la dirección del eje de la articulación “i”.
- “ θ ” que representa la distancia angular desde la recta a_{i-1} a la recta a_i .

También cabe mencionar que:

- Los parámetros a y α son características únicas de cada eslabón siempre son constante.
- Los parámetros θ y “d” describen el movimiento relativo entre los eslabones consecutivos.
- En una articulación rotoide, θ es variable y “d” es constante.
- En una articulación prismática, “d” es variable y θ es constante. (Fernandez, 2016)

3.2.6. LINKS, JOINTS Y SUS PARÁMETROS.

Un manipulador mecánico consiste en una secuencia de cuerpos rígidos, llamados enlaces, conectados por cualquiera de las articulaciones giratorias y prismáticas. Cada par de enlace de unión constituye un grado de libertad. Por lo tanto, para un n grado de libertad manipulador, hay n pares de enlaces articulados con el enlace 0 adjunto a una base de soporte donde generalmente se establece un marco de coordenadas inerciales para este sistema dinámico, y el último enlace se adjunta con una herramienta, las articulaciones y los enlaces están numerados hacia el exterior desde la base; esto, la articulación 1 es el punto de conexión entre el enlace 1 y la base de soporte. Cada enlace está conectado, como máximo, a otro para que no se formen bucles cerrados. En general, dos enlaces están conectados por una articulación de par inferior que tiene superficies que se deslizan una sobre la otra mientras permanecen en contacto (Goyal, 2002).

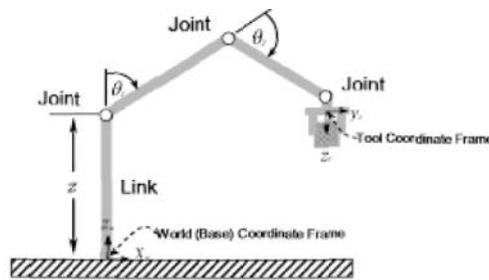


Ilustración 3. 14 Representación de una articulación y un enlace.

3.2.7. ALGORITMO DENAVIT-HARTENBERG.

Además de definir los parámetros, también se establecieron un conjunto de convenciones sobre los sistemas de ejes asociados que permiten sistematizar el análisis de los robots manipuladores:

- Los eslabones se enumeran crecientemente. Desde la base hacia el efector final. Asignando el número 1 al eslabón que se encuentre unido a la base. La base recibirá el número 0.
- Las articulaciones se enumeran también en forma creciente comenzando por 1, desde la que una a la base con el primer elemento hasta la que sirve de junta al elemento terminal.
- Localizar los ejes de las articulaciones, en las rotoide el eje será el eje de giro; y en las prismáticas, el eje de desplazamiento.
- Definir los ejes Z_i empezando desde la base, cada Z_i estará ubicado en el eje de la articulación correspondiente al extremo final del elemento.
- El origen del sistema de coordenadas de la base (S_0) se escogerá arbitrariamente, como cualquier punto del eje Z_0 . También será arbitrario el sentido de X_0 , pero Y_0 quedará determinado por las reglas del producto vectorial, tal que $Z = (X.Y)$.
- El origen de cada sistema ' S_i ' quedará determinado por la intersección de la normal (a) común a cada eje Z_i y Z_{i-1} . El eje X_i estará en dirección de esa normal y correspondiente Y_i cumplirá con la regla de todo sistema dextrógiro.
- El sistema ' S_n ' se ubicará en el elemento terminal del robot y el eje Z_n conservará la dirección de Z_{n-1} .
- Θ_i será el ángulo en que debe girarse al eje X_{i-1} , alrededor de Z_{i-1} , para quedar alineado con X_i .
- d_i será la distancia que deberá desplazarse a X_{i-1} , en la dirección de Z_{i-1} , para quedar alineado a X_i .
- a_i será la distancia que deberá desplazarse al sistema S_{i-1} , en la dirección de X_i para que su origen coincida con el de S_i .
- α_i será el ángulo en que debe girarse a Z_{i-1} , alrededor de X_i para quedar alineado con Z_i .

- Deben obtenerse tantas matrices de paso homogéneas A_i^{i-1} como elementos que existan en la cadena cinemática, incluyendo la matriz A_1^0 correspondiente a la base.
- La matriz de transformación del robot se obtendrá como el producto de todas las matrices de paso homogéneas. $T = A_1^0 * A_2^1 * A_n^{n-1}$. Dependiendo del tipo de articulación utilizada, cada matriz de paso homogénea tendrá como variable al ángulo θ (si es *rotoide*) o la distancia d (si es *prismática*). A estos dos se les llama parámetro de articulación (q).
- Para cada movimiento que se realiza a partir de la condición alcanzada con el movimiento previo, el modelo explicativo de todo procedimiento se obtiene mediante transformaciones que se post-multiplican, tal como se muestra a continuación: (Bonilla, 2013)

$$A_i^{i-1} = T(z, \theta_i)T(0,0, a_i)T(d_i, 0,0)RT(x, \alpha_i) \quad \text{Ecuación 12}$$

$$A_i^{i-1} = \begin{bmatrix} C\theta_i & -S\theta_i & 0 & 0 \\ S\theta_i & C\theta_i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & C\alpha_i & -S\alpha_i & 0 \\ 0 & S\alpha_i & C\alpha_i & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{Ecuación 13}$$

3.2.8. CINEMÁTICA DIRECTA.

Proporcionará la posición y la orientación del efector final del robot expresada en coordenadas operacionales pero en función de sus coordenadas articulares. El empleo de la cinemática directa resulta de utilidad en la planificación de trayectorias y en el control cartesiano (Cortés, 2012).

Las articulaciones generalmente representan un grado de libertad, y pueden ser prismáticas o articuladas y/o revolutas. Para ejemplificar se presenta la siguiente ilustración donde se puede observar un ángulo “ θ ” como una variable para una articulación revoluta y una variable “ d ” que representa distancia para una articulación prismática. Estas representan y determinan la orientación de los marcos cinemáticos de coordenadas.

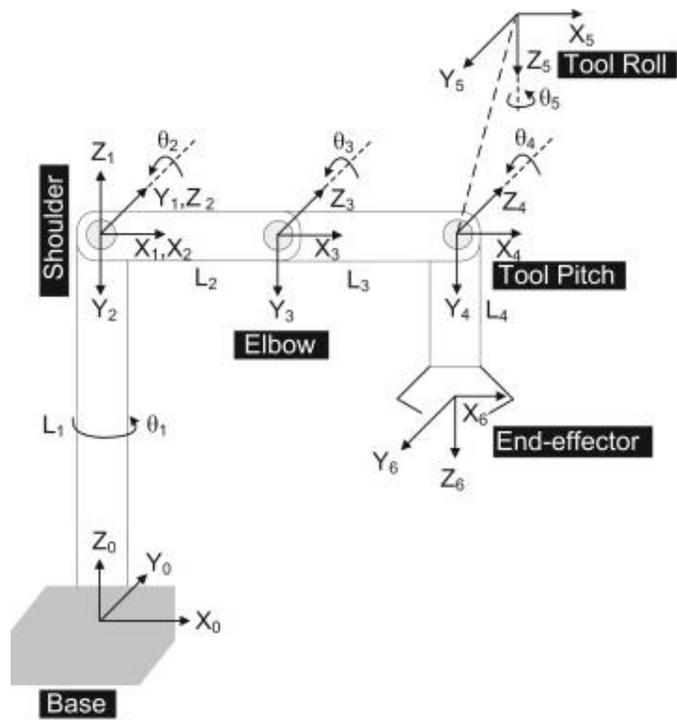
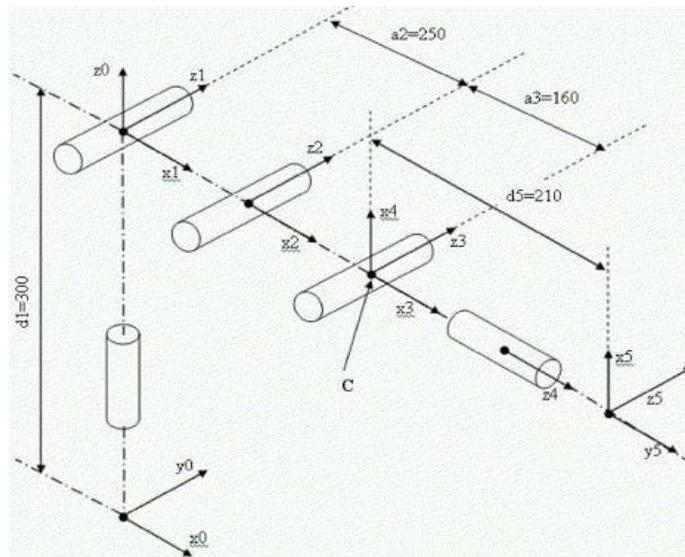


Ilustración 3. 15 Representación de sistemas.



RV-M1 robot arm link coordinate parameters					
Joint i	a_i	α_i	d_i	θ_i	Joint range
1	0	-90°	300	θ_1	-150° to 150°
2	250	0	0	θ_2	-100° to 30°
3	160	0	0	θ_3	0° to 110°
4	0	-90°	0	θ_4	-180° to 0°
5	0	90°	210	θ_5	-180° to 180°

Ilustración 3. 16 Representación y tabla denavit-hartenberg.

3.2.9. CINEMÁTICA INVERSA.

Consiste en encontrar los valores que deben adoptar las coordenadas articulares del robot, para que su extremo se posicione y oriente según una determinada localización espacial, es decir es un problema no lineal que relaciona las coordenadas cartesianas y la orientación de la herramienta del extremo final del robot manipulador (Cortés, 2012).

3.3. CARACTERÍSTICAS PRINCIPALES DEL ROBOT.

3.3.1. DESCRIPCIÓN Y CARACTERÍSTICAS DEL ROBOT “RVM1”

El manipulador Mitsubishi RV-M1 es un robot de ensamble para piezas mecánicas, cuenta con cinco grados de libertad. (INDUSTRIAL MICRO-ROBOT SYSTEM MODEL RV-M1). Su sistema permite el manejo del robot y está constituido por:

- Brazo articulado
- Efecto final.
- Teaching box.
- Controlador
- Cables de conexión.

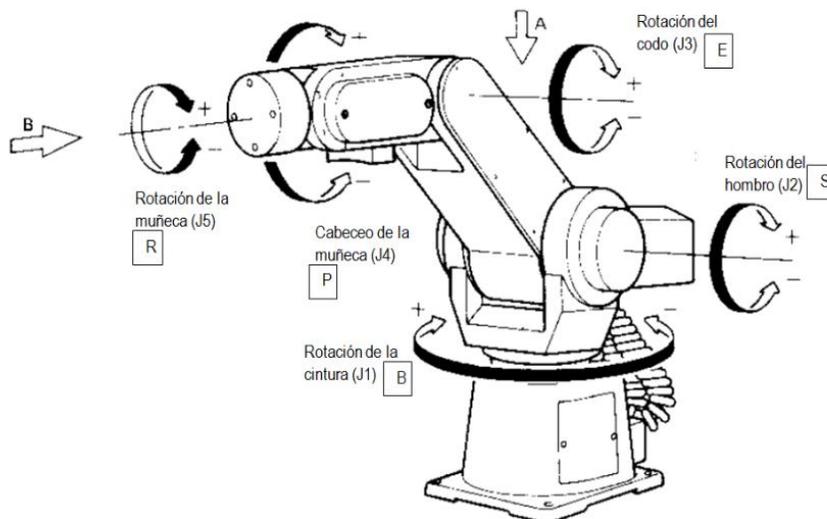


Ilustración 3. 17 Robot Mitsubishi RV-M1.

El robot Mitsubishi MOVEMASTER RV-M1 posee ciertas características las cuales permiten saber cómo opera cada uno de los eslabones que lo componen.

Eje	Control
J1: cintura	X
J2: hombro	Y
J3: codo	Z
J4: pitch	P
J5: roll	R

Tabla 3.1. Movimiento de los ejes.

Tipos de movimientos:

- PTP: movimiento de articulación por articulación del robot.
- XYZ: este tipo de movimiento, el robot se mueve de tal manera que la posición de su herramienta o pinza se mantienen deslizándose sobre uno de los ejes cartesianos.
- TOOL: arrastra los ejes XYZ a la muñeca de la herramienta del brazo del robot y los ejes se transportan según sea la posición.
- Características de las articulaciones: (INDUSTRIAL MICRO-ROBOT SYSTEM MODEL RV-M1)

Elemento	Especificación	Observaciones
Rotación de cintura	3000	Eje J1 de revolución
Rotación de hombro	1300	Eje J2 de revolución
Rotación de codo	1100	Eje J3 de revolución
Muñeca (pitch)	+/- 900	Eje J4 de revolución
Muñeca (roll)	+/- 1800	Eje J5 de revolución
Parte superior del brazo	250mm	
Brazo delantero	160mmk	

Tabla 3.2. Características generales.

3.3.2. ESPACIO DE TRABAJO.

El robot requiere de espacio para su operación, suficiente para realizar sus movimientos a continuación se presenta el mapa del espacio de operación del robot (INDUSTRIAL MICRO-ROBOT SYSTEM MODEL RV-M1).

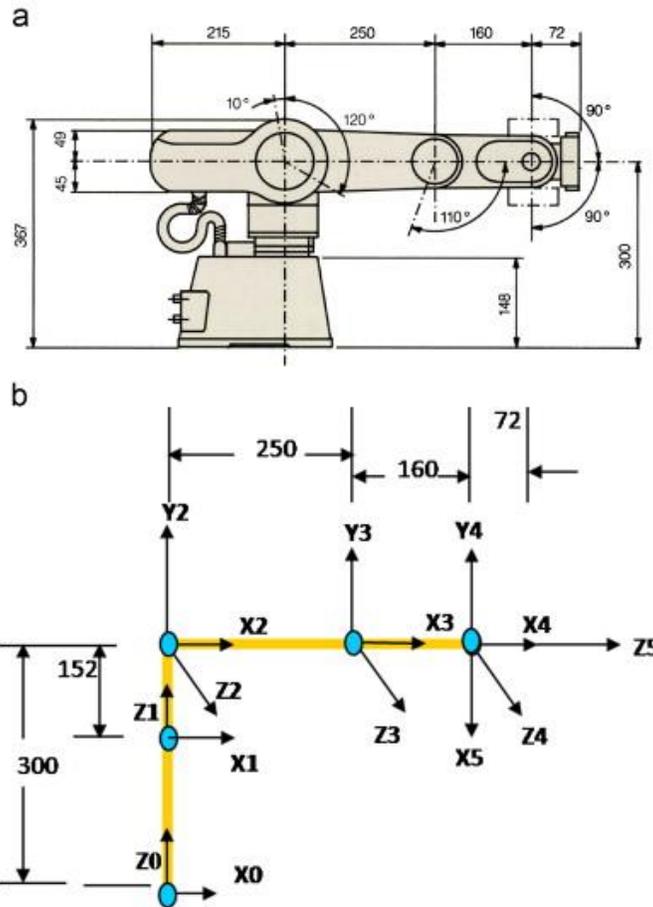


Ilustración 3. 20 Medidas del robot.

Se establecen los ejes que determinan los parámetros de Denavit-Hartenber y los parámetros D-H para el manipulador se presentan a continuación en la siguiente tabla:

Grado de Libertad	Θ_i	d_i	a_i	α_i
1	Θ_1	d_1	0	90
2	Θ_2	0	a_2	0
3	Θ_3	0	a_3	0
4	$\Theta_4 + 90^\circ$	0	0	90
5	Θ_5	d_4	0	0

Tabla 3.3. Parámetros denavit-hartenberg.

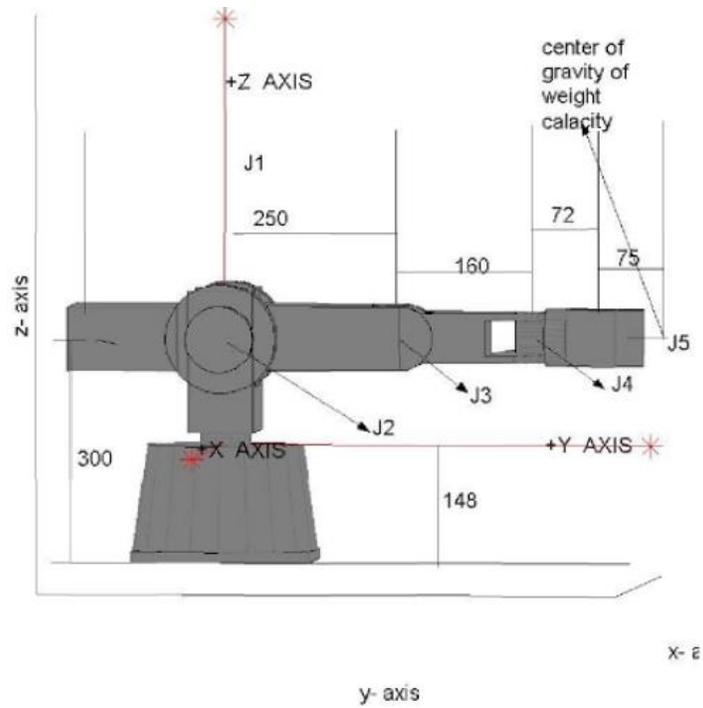


Ilustración 3. 21Robot RV-M1.

Para determinar el valor de los parámetros se recurre a las notas técnicas del robot, que establece la configuración de referencia:

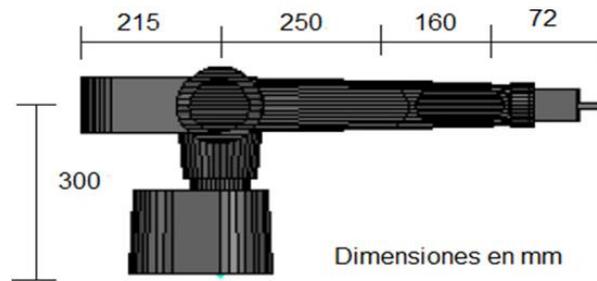


Ilustración 3. 22 Medidas de referencia del robot.

$$d_1 = 300^\circ$$

$$d_5 = 72^\circ$$

$$a_2 = 250^\circ$$

$$a_3 = 160^\circ$$

$$a_1 = 90^\circ$$

$$a_4 = 90^\circ$$

La matriz de transformación homogénea está dada por:

$$A_i^{i-1} = \begin{bmatrix} \cos\theta & \cos\alpha\sin\theta & \sin\alpha\sin\theta & a\cos\theta \\ \sin\theta & \cos\alpha\sin\theta & -\sin\alpha\cos\theta & a\sin\theta \\ 0 & \sin\alpha & \cos\theta & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{Ecuación 14}$$

A partir de estos valores se puede obtener la matriz de transformación homogénea A_{i-1} sustituyendo los parámetros de eslabón.

$$A_1^0 = \begin{bmatrix} \cos\theta_1 & 0 & \sin\theta_1 & 0 \\ \sin\theta_1 & 0 & -\cos\theta_1 & 0 \\ 0 & 1 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{Ecuación 15}$$

$$A_2^1 = \begin{bmatrix} \cos\theta_2 & -\sin\theta_2 & 0 & a_2\cos\theta_2 \\ \sin\theta_2 & \cos\theta_2 & 0 & a_2\sin\theta_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{Ecuación 16}$$

$$A_3^2 = \begin{bmatrix} \cos\theta_3 & -\sin\theta_3 & 0 & a_2\cos\theta_3 \\ \sin\theta_3 & \cos\theta_3 & 0 & a_2\sin\theta_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{Ecuación 17}$$

$$A_4^3 = \begin{bmatrix} \cos(\theta_4 + 1.5708) & 0 & \sin(\theta_4 + 1.5708) & 0 \\ \sin(\theta_4 + 1.5708) & 0 & -\cos(\theta_4 + 1.5708) & 0 \\ 0 & 1.0 & 0 & 0 \\ 0 & 0 & 0 & 1.0 \end{bmatrix} \quad \text{Ecuación 18}$$

$$A_5^4 = \begin{bmatrix} \cos\theta_5 & -\sin\theta_5 & 0 & 0 \\ \sin\theta_5 & \cos\theta_5 & 0 & 0 \\ 0 & 0 & 1 & d_5 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{Ecuación 19}$$

Las matrices homogéneas de transformación se multiplican para obtener la matriz de cada marco: $T_0^5 = T_0^1 T_0^2 T_0^3 T_0^4 T_0^5$

$$A_{10} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 300 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{Ecuación 20}$$

$$A_{20} = A_{10} \cdot A_{21} = \begin{bmatrix} 1 & 0 & 0 & 250 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 300 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{Ecuación 21}$$

$$A_{30} = A_{20} \cdot A_{32} = \begin{bmatrix} 1 & 0 & 0 & 410 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 300 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{Ecuación 22}$$

$$A_{40} = A_{30} \cdot A_{43} = \begin{bmatrix} -3.67 \times 10^{-6} & 0 & 1 & 410 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 3.67 \times 10^{-6} & 300 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{Ecuación 23}$$

$$A_{50} = A_{40} \cdot A_{54} = \begin{bmatrix} -3.67 \times 10^{-6} & 0 & 1 & 482 \\ 0 & -1 & 0 & 0 \\ 1 & 0 & 3.67 \times 10^{-6} & 300 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{Ecuación 24}$$

$$= \begin{bmatrix} n_x & s_x & a_x & p_x \\ n_y & s_y & a_y & p_y \\ n_z & s_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{Ecuación 25}$$

Donde s es la orientación, a del vector de aproximación y p el vector de posición de la herramienta. Finalmente el vector n está definido como: $n = s \times a$, de esta forma se obtiene la cinemática directa del robot; a su vez esta servirá en el futuro para poder calcular por medio de operaciones algebraicas la cinemática inversa del robot.

3.3.4. SOLUCION DE CINEMÁTICA INVERSA.

El problema de la cinemática inversa en robot manipuladores consiste en determinar los valores que deben tomar sus articulaciones para que el efector final alcance una determinada posición (x, y, z) y orientación (yaw, pitch, roll). Este problema es difícil de resolver, ya que a diferencia de la cinemática directa pueden existir distintas configuraciones de las articulaciones que sitúen al efector final en una misma posición y orientación lo que implica que existen múltiples soluciones para una misma configuración del efector final.

Existen varios métodos para resolver dicho problema que se pueden clasificar principalmente en dos tipos, primeramente se tienen los métodos analíticos que son aquellos métodos que permiten encontrar una solución cerrada o en otras palabras encontrar

expresiones matemáticas que definan directamente cada variable articular a partir de su modelo cinemático y de la posición y orientación deseada para el efector final. Dentro de esta categoría se tienen el método geométrico, el método de resolución por matrices de transformación homogénea y el método de desacoplamiento cinemático.

La principal ventaja de estos métodos es que permiten encontrar expresiones matemáticas que permiten un rápido cálculo de la cinemática inversa para cualquier posición y orientación alcanzable por el robot.

Sin embargo, las principales desventajas que poseen estos métodos es que no suele ser sencillo encontrar relaciones matemáticas que permitan encontrar las ecuaciones necesarias para cada una de las articulaciones; además las solución encontrada únicamente es válida para el modelo del robot al cual se le hizo el análisis y otra desventaja es que existen robots que debido a su estructura no pueden ser resuelto por medio de estos métodos.

En segundo se tienen los métodos iterativos; estos métodos suelen partir de la configuración actual del robot o de una configuración predeterminada y van ajustando los valores de las variables articulares hasta converger a una configuración que satisfaga la posición y orientación deseada. Dentro de esta clasificación se tienen el algoritmo CCD y el método de Newton-Raphson por mencionar algunos. La principal ventaja de estos métodos es que pueden resolver el problema sin la necesidad de encontrar relaciones matemáticas entre las articulaciones, lo que permite implementar un mismo algoritmo para varios tipos de robots o con robots con iguales grados de libertad, ya que únicamente basta con ajustar el algoritmo según la naturaleza de cada uno de las articulaciones (tipo de articulación, límites de movimiento, parámetros DH). La principal desventaja de estos métodos es el tiempo que les toma converger a una solución lo que los vuelve ineficaces para aplicaciones en tiempo real, además de que la convergencia no siempre está garantizada.

Hay que mencionar que también trabajos recientes resuelven este problema utilizando algoritmos de aprendizaje automático como los algoritmos genéticos y las redes neuronales. Entre las ventajas que ofrecen estos métodos es que al igual que los métodos iterativos estos métodos no dependen de la estructura del manipulador y la mayoría de ellos ni siquiera requieren que se les especifique la naturaleza de cada articulación. Además, algunos de estos métodos una vez que han pasado por la etapa de entrenamiento pueden resolver la cinemática inversa rápidamente por lo que pueden ser aplicados en problemas de

tiempo real. Sin embargo, las principales desventajas de estos métodos es que el entrenamiento del robot suele ser un proceso altamente tardado además de que la eficiencia final del robot dependerá del entrenamiento que se le haya realizado.

Debido a todas las ventajas y desventajas mencionadas anteriormente en la práctica son preferibles los métodos analíticos ya que la solución encontrada puede ser utilizada en aplicaciones en tiempo real, y además son los únicos que garantizan siempre encontrar una solución siempre y cuando la posición y orientación deseada es alcanzable por el robot. Aplicaciones en tiempo real y además son los únicos que garantizan siempre encontrar una solución siempre y cuando la posición y orientación deseada es alcanzable por el robot.

En este trabajo se utilizó el método de resolución por matrices de transformación homogénea; este método consiste en encontrar una expresión para cada uno de las variables articulares a partir de las matrices de transformación homogénea que definen el modelo cinemático directo. Antes de exponer el método con mayores detalles se definirán las matrices homogéneas usadas en el modelo cinemático directo del robot RV-M1; la matriz de transformación que relaciona el sistema de referencia fijo con el sistema de referencia de la herramienta está definida de la siguiente manera:

$$T_5^0 = A_1^0 A_2^1 A_3^2 A_4^3 A_5^4 \quad \text{Ecuación 26}$$

Donde las matrices A_i^{i-1} son las matrices que relacionan el marco de referencia de la articulación $i-1$ con el marco de referencia de la articulación i . Para el robot RV-M1 y siguiendo la notación Denavith-Hartenbergh, estas matrices se definen así:

$$A_2^1 = \begin{bmatrix} c_2 & -s_2 & 0 & a_2 c_2 \\ s_2 & c_2 & 0 & a_2 s_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_3^2 = \begin{bmatrix} c_3 & -s_3 & 0 & a_3 c_3 \\ s_3 & c_3 & 0 & a_3 s_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{Ecuación 27}$$

$$A_4^3 = \begin{bmatrix} c_4 & 0 & s_4 & 0 \\ s_4 & 0 & -c_4 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_5^4 = \begin{bmatrix} c_5 & -s_5 & 0 & 0 \\ s_5 & c_5 & 0 & 0 \\ 0 & 0 & 1 & d_5 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Donde s_i y c_i es respectivamente el seno y el coseno de la variable articular θ_i y en donde d_1 , a_2 , a_3 y d_5 son los parámetros DH. Sustituyendo las matrices de (27) en (26) tenemos que la matriz de transformación del marco fijo al marco de referencia queda definido como:

$$T_5^0 = \begin{bmatrix} c_1 c_{234} c_5 + s_1 s_5 & -c_1 c_{234} s_5 + s_1 c_5 & c_1 s_{234} & c_1 (d_5 s_{234} + a_3 c_{23} + a_2 c_2) \\ s_1 c_{234} c_5 - c_1 s_5 & -s_1 c_{234} s_5 - c_1 c_5 & s_1 s_{234} & s_1 (d_5 s_{234} + a_3 c_{23} + a_2 c_2) \\ s_{234} c_5 & -s_{234} s_5 & -c_{234} & d_1 - d_5 c_{234} + a_3 s_{23} + a_2 s_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{Ecuación 28}$$

La orientación y posición de la herramienta puede ser definida por medio de una matriz de orientación R_H y una matriz de posición P_H definidas de la siguiente manera:

$$R_H = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \quad \text{y} \quad P_H = \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix} \quad \text{Ecuación 29}$$

Con esto la matriz que define la posición y orientación deseada queda de la siguiente:

$$H = \begin{bmatrix} R_H & P_H \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & p_x \\ r_{21} & r_{22} & r_{23} & p_y \\ r_{31} & r_{32} & r_{33} & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{Ecuación 30}$$

También se puede representar por medio de los vectores de posición p , de enfoque a , de orientación o y del vector normal n de la siguiente manera:

$$H = \begin{bmatrix} n & o & a & p \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{Ecuación 31}$$

Ahora con esto se puede definir el problema de la cinemática inversa como el problema de encontrar los valores de θ_1 , θ_2 , θ_3 , θ_4 , θ_5 que satisfaga la ecuación:

$$T_5^0 = H$$

Lo que es igual a

$$\begin{bmatrix} c_1 c_{234} c_5 + s_1 s_5 & -c_1 c_{234} s_5 + s_1 c_5 & c_1 s_{234} & c_1 (d_5 s_{234} + a_3 c_{23} + a_2 c_2) \\ s_1 c_{234} c_5 - c_1 s_5 & -s_1 c_{234} s_5 - c_1 c_5 & s_1 s_{234} & s_1 (d_5 s_{234} + a_3 c_{23} + a_2 c_2) \\ & s_{234} c_5 & -s_{234} s_5 & d_1 - d_5 c_{234} + a_3 s_{23} + a_2 s_2 \\ & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Ecuación 32

La ecuación anterior (32) junto con las matrices de (27) son la base del método de resolución por matrices de transformación homogénea. La igualdad en (32) indica que cada elemento de la matriz T_5^0 es igual a su respectivo elemento en la matriz H , de esta forma la ecuación nos proporciona una expresión en términos de las variables articulares para cada uno de los elementos de la matriz de orientación y de la matriz de posición de la herramienta. Para resolver la cinemática inversa iremos multiplicando las matrices de (32) con las inversas de las matrices de (27) consecutivamente siguiendo el mismo orden en el que se fueron multiplicando en (26), es decir, primero se multiplica con la matriz $(A_1^0)^{-1}$, luego con $(A_2^1)^{-1}$ y así sucesivamente; en cada ocasión se trata de encontrar expresiones matemáticas que permitan resolver cada una de las variables articulares en términos de elementos que ya se conozcan. Como se mencionó anteriormente, se inicia multiplicando (32) con la inversa de la matriz A_1^0 ; la matriz inversa está definida de la siguiente manera:

$$(A_1^0)^{-1} = \begin{bmatrix} c_1 & s_1 & 0 & 0 \\ 0 & 0 & 1 & -d_1 \\ s_1 & -c_1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Ecuación 33

Multiplicando esta matriz en (32) se tiene:

$$(A_1^0)^{-1} T_5^0 = (A_1^0)^{-1} H$$

$$\begin{bmatrix} c_1 & s_1 & 0 & 0 \\ 0 & 0 & 1 & -d_1 \\ s_1 & -c_1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c_1 c_{234} c_5 + s_1 s_5 & -c_1 c_{234} s_5 + s_1 c_5 & c_1 s_{234} & c_1 (d_5 s_{234} + a_3 c_{23} + a_2 c_2) \\ s_1 c_{234} c_5 - c_1 s_5 & -s_1 c_{234} s_5 - c_1 c_5 & s_1 s_{234} & s_1 (d_5 s_{234} + a_3 c_{23} + a_2 c_2) \\ & s_{234} c_5 & -s_{234} s_5 & d_1 - d_5 c_{234} + a_3 s_{23} + a_2 s_2 \\ & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} c_1 & s_1 & 0 & 0 \\ 0 & 0 & 1 & -d_1 \\ s_1 & -c_1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} f_{11}(n) & f_{11}(o) & f_{11}(a) & f_{11}(p) \\ f_{21}(n) & f_{21}(o) & f_{21}(a) & f_{21}(p) \\ f_{31}(n) & f_{31}(o) & f_{31}(a) & f_{31}(p) \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} n_x c_1 + n_y s_1 & o_x c_1 + o_y s_1 & a_x c_1 + a_y s_1 & p_x c_1 + p_y s_1 \\ n_z & o_z & a_z & p_z - d_1 \\ n_x s_1 - n_y c_1 & o_x s_1 - o_y c_1 & a_x s_1 - a_y c_1 & p_x s_1 - p_y c_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Ecuación 34

Donde para un vector cualquiera $k = [k_x \quad k_y \quad k_z]$ se define:

$$f_{11}(k) = c_1 k_x + s_1 k_y$$

$$f_{21}(k) = k_z$$

$$f_{31}(k) = s_1 k_x - c_1 k_y$$

Ecuación 35

Si se toman los componentes (3, 4) de ambas matrices en (34) se tiene la siguiente ecuación:

$$p_x s_1 - p_y c_1 = 0$$

Re arreglando se tiene:

$$p_x s_1 = p_y c_1$$

$$\frac{s_1}{c_1} = \frac{p_y}{p_x}$$

Por definición $\tan(\alpha) = \frac{\sin(\alpha)}{\cos(\alpha)}$ sustituyendo en la ecuación anterior se tiene:

$$t_1 = \frac{p_y}{p_x}$$

Aplicando el arco tangente en ambos lados de la ecuación para despejar θ_1 se tiene:

$$\theta_1 = \text{atan}\left(\frac{p_y}{p_x}\right) = \text{atan2}(p_y, p_x)$$

Ecuación 36

De (34) también se pueden obtener expresiones para s_5 y c_5 tomando ahora los elementos (3, 1) y (3, 2) respectivamente. Tomando los elementos (3, 1) se obtiene la siguiente ecuación:

$$c_1(c_1 s_5 - c_{234} c_5 s_1) + s_1(s_1 s_5 + c_{234} c_1 c_5) = n_x s_1 - n_y c_1$$

Simplificando

$$c_1^2 s_5 - c_{234} c_5 s_1 c_1 + s_1^2 s_5 + c_{234} c_5 s_1 c_1 = n_x s_1 - n_y c_1$$

$$c_1^2 s_5 + s_1^2 s_5 = n_x s_1 - n_y c_1$$

$$s_5(c_1^2 + s_1^2) = n_x s_1 - n_y c_1$$

Usando la identidad trigonométrica $\sin(\alpha)^2 + \cos(\alpha)^2 = 1$ se tiene:

$$s_5 = n_x s_1 - n_y c_1$$

Ecuación 37

Ahora tomando los elementos (3, 2) se tiene:

$$c_1(c_1 c_5 + c_{234} s_1 s_5) + s_1(c_5 s_1 - c_{234} c_1 s_5) = o_x s_1 - o_y c_1$$

Simplificando

$$c_1^2 c_5 + c_{234} s_5 c_1 s_1 + c_5 s_1^2 - c_{234} s_5 c_1 s_1 = o_x s_1 - o_y c_1$$

$$c_1^2 c_5 + s_1^2 c_5 = o_x s_1 - o_y c_1$$

Ecuación 38

$$c_5 = o_x s_1 - o_y c_1$$

Ahora usando la definición de la tangente dividiendo la ecuación 37 entre la ecuación 38 se tiene:

$$\frac{s_5}{c_5} = \frac{n_x s_1 - n_y c_1}{o_x s_1 - o_y c_1}$$

Lo que es igual a:

$$t_5 = \frac{n_x s_1 - n_y c_1}{o_x s_1 - o_y c_1}$$

Aplicando el arco tangente en ambos lados se tiene:

$$\theta_5 = \text{atan}\left(\frac{n_x s_1 - n_y c_1}{o_x s_1 - o_y c_1}\right) = \text{atan2}(n_x s_1 - n_y c_1, o_x s_1 - o_y c_1)$$

Ecuación 39

De igual manera de la ecuación (22) se pueden obtener expresiones para s_{234} y c_{234} a partir de los elementos (1, 3) y (2,3). Tomando los elementos (1, 3) se tiene:

$$s_{234} c_1^2 + s_{234} s_1^2 = a_x c_1 + a_y s_1$$

Simplificando:

$$s_{234} = a_x c_1 + a_y s_1$$

Ecuación 40

Tomando ahora los elementos (2, 3) se tiene:

$$-c_{234} = a_z$$

Multiplicando por -1 la ecuación anterior se tiene:

$$c_{234} = -a_z$$

Ecuación 41

Usando nuevamente la fórmula de la tangente se tiene:

$$t_{234} = \frac{s_{234}}{c_{234}} = \frac{a_x c_1 + a_y s_1}{-a_z}$$

Aplicando el arco tangente se tiene:

$$\theta_{234} = \text{atan}\left(\frac{a_x c_1 + a_y s_1}{-a_z}\right) = \text{atan2}(a_x c_1 + a_y s_1, -a_z) \quad \text{Ecuación 42}$$

Ahora se pueden usar los elementos (1, 4) y (2, 4) para buscar una expresión para θ_3 . De esos elementos se obtienen las siguientes ecuaciones:

$$\begin{aligned} a_3 c_{23} + a_2 c_2 + d_5 s_{234} &= p_x c_1 + p_y s_1 \\ a_3 s_{23} + a_2 s_2 - d_5 c_{234} &= p_z - d_1 \end{aligned} \quad \text{Ecuación 43}$$

Si se define:

$$\begin{aligned} A &= p_x c_1 + p_y s_1 - d_5 s_{234} \\ B &= p_z - d_1 + d_5 c_{234} \end{aligned} \quad \text{Ecuación 44}$$

Sustituyendo (43) en (44) se tiene:

$$\begin{aligned} a_3 c_{23} + a_2 c_2 &= A \\ a_3 s_{23} + a_2 s_2 &= B \end{aligned} \quad \text{Ecuación 45}$$

Elevando al cuadrado ambas ecuaciones y sumándolas se tiene:

$$\begin{aligned} a_3^2 c_{23}^2 + 2a_3 a_2 c_2 c_{23} + a_2^2 c_2^2 + a_3^2 s_{23}^2 + 2a_3 a_2 s_2 s_{23} + a_2^2 s_2^2 &= A^2 + B^2 \\ a_3^2 (c_{23}^2 + s_{23}^2) + a_2^2 (c_2^2 + s_2^2) + 2a_2 a_3 (c_2 c_{23} + s_2 s_{23}) &= A^2 + B^2 \end{aligned}$$

Utilizando la identidad trigonométrica $\cos(\alpha - \beta) = \cos(\alpha)\cos(\beta) + \sin(\alpha)\sin(\beta)$ para simplificar se obtiene:

$$a_3^2 + a_2^2 + 2a_2 a_3 c_3 = A^2 + B^2$$

Despejando c_3 :

$$c_3 = \frac{A^2 + B^2 - a_3^2 - a_2^2}{2a_2 a_3} \quad \text{Ecuación 46}$$

Se puede definir el seno de un ángulo en términos del coseno del mismo ángulo de usando la identidad $\sin(\alpha) = \sqrt{1 - \cos(\alpha)}$ dado que ya se conoce c_3 se puede obtener la siguiente ecuación para s_3 :

$$s_3 = \pm \sqrt{1 - c_3} \quad \text{Ecuación 47}$$

Ahora que ya se conoce c_3 y s_3 se puede obtener una ecuación para θ_3 de manera similar a los ángulos anteriores usando la definición de la tangente lo que queda:

$$\theta_3 = \text{atan}\left(\frac{\pm s_3}{c_3}\right) = \text{atan2}(\pm s_3, c_3) \quad \text{Ecuación 48}$$

El seno y el coseno de θ_2 se obtiene resolviendo simultáneamente las ecuaciones dadas en (45) de donde se obtiene:

$$c_2 = \frac{(a_3 c_3 + a_2)A + (a_3 s_3)B}{a_2^2 + a_3^2 + 2a_2 a_3 c_3} \quad \text{Ecuación 49}$$

Y de manera similar a como se hizo con s_3 se puede obtener una ecuación para s_2 :

$$s_2 = \pm \sqrt{1 - c_2} \quad \text{Ecuación 50}$$

Y por lo tanto al igual que con θ_3 se obtiene la siguiente expresión para θ_2 :

$$\theta_2 = \text{atan}\left(\frac{\pm s_2}{c_2}\right) = \text{atan2}(\pm s_2, c_2) \quad \text{Ecuación 51}$$

Finalmente, se puede obtener θ_4 a partir de los ángulos θ_{234} , θ_2 , θ_3 de la siguiente manera:

$$\theta_4 = \theta_{234} - \theta_2 - \theta_3 \quad \text{Ecuación 52}$$

Con estas ecuaciones el algoritmo para calcular la cinemática inversa es el siguiente:

Se calcula θ_1 usando (36).

Teniendo θ_1 se puede calcular ahora θ_5 usando (39).

Se calcula s_{234} usando (40).

Se calcula c_{234} usando (41).

Ahora se calcula θ_{234} usando (42) con los valores obtenidos en los pasos 3 y 4.

Se calcula A y B usando (44).

Se calcula c_3 usando (46) con los valores obtenidos en el paso 6.

Se calcula s_3 usando (47) con el valor obtenido en el paso 7.

Se calcula θ_3 usando (48) con los valores obtenidos en los pasos 7 y 8.

Se calcula c_2 usando (49) con los valores obtenidos en los pasos 6, 7 y 8.

Se calcula s_2 usando (50) con el valor obtenido en el paso 10.

Se calcula θ_2 usando (51) con los valores obtenidos en los pasos 10 y 11.

Finalmente, se calcula θ_4 usando (52) a partir de los valores obtenidos en los pasos 9 y 12.

Durante la implementación existen varios aspectos que se deben considerar para que los cálculos sean los correctos. Primeramente, se debe tener en cuenta que para que la vista del manipulador en matlab sea la adecuada; se han rotado los ejes x y y en 90° alrededor del eje z; debido a esto se debe rotar la orientación y la posición deseada 90° alrededor del eje z para que los valores de las variables articulares que se calculen produzcan realmente

la posición y orientación debida. Para lograr esto se multiplican las matrices R_H y P_H definidas en (30) con la siguiente matriz de rotación:

$$R = \begin{bmatrix} \cos(90) & -\sin(90) & 0 \\ \sin(90) & \cos(90) & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \text{Ecuación 53}$$

Para el cálculo de θ_1 se debe tener en cuenta que atan produce valores en el rango de $[-90, 90]$ y que atan2 produce valores en el rango $[-180, 180]$, mientras que por otro lado θ_1 solamente puede tomar valores que estén en el rango $[-150, 150]$. Para resolver este problema se utiliza primeramente atan2 para el cálculo luego se verifica si el valor obtenido se encuentra dentro del rango permitido para θ_1 ; si ese es el caso se vuelve a calcular θ_1 usando ahora la función atan.

Una consideración muy importante que se debe tomar en cuenta es que c_3 representa el coseno de un ángulo por lo que debe ser siempre un valor entre -1 y 1, sin embargo debido a la naturaleza de (46) se puede dar el caso de que al momento de calcular produzca un valor mayor a 1; ese problema se da cuando la posición y orientación deseada produce una configuración que hace que los elementos que influyen en la fórmula de (46) formen lo que sería un triángulo degenerado lo que en geometría se define como un triángulo que posee un lado con longitud 0; en ese caso el ángulo formado es de 180 por lo que el valor de c_3 debe ser 1.

Ahora bien se debe notar que el rango permitido para el ángulo θ_3 es $[-110, 0]$, es decir, siempre se tienen valores negativos; debido a eso los dos posibles valores que se obtienen en (48) se deben multiplicar con -1 para asegurar que sean valores permitidos.

Para el cálculo de c_2 se tiene un problema similar al que existe en c_3 , sin embargo aquí no solo se pueden producir valores mayores a 1, sino que también se pueden producir valores menores a -1, por lo que al igual que en c_3 si se da este caso el valor correcto debe ser de 1 y -1 respectivamente. Además, hay que mencionar que como este cálculo depende de c_3 y de s_3 existe dos posibles valores para c_2 cada una correspondiente a los dos posibles valores de s_3 .

Por lo mencionado anteriormente y dado que en (50) existen dos posibles valores para s_2 se deriva que se tienen 4 posibles valores para θ_2 al momento de calcular usando (26) ya que para cada uno de los dos posibles valores de c_2 se tienen dos posibles valores para s_2 .

Para finalizar se debe tener en cuenta que el algoritmo produce 4 posibles soluciones, sin embargo, cada una de ellas podría ser válida o no por lo que es necesario verificar cada solución. Para esto se toma cada uno de los ángulos que conforman la solución y se verifica si se encuentra en su respectivo rango si todos los ángulos se encuentran dentro de los rangos permitidos entonces se considerara a la solución válida; de lo contrario, si al menos un ángulo está fuera de rango, entonces la solución se considerará inválida. Una vez verificado, lo más común es que exista más de una solución válida por lo que se necesita escoger una; para hacerlo se ha usado la cinemática directa del robot de tal manera que se calcula la posición que alcanzaría el robot al usar cada uno de las soluciones válidas obtenidas, y se calcula posteriormente la distancia que existe entre esta posición y la posición deseada de la herramienta usando la distancia euclidiana; una vez hecho esto, simplemente se toma aquella cuya distancia sea la menor (si existe más de una solución que produce la distancia mínima entonces tomamos la primera de manera arbitraria).

3.4. SIMULACIÓN Y RESULTADOS.

La simulación es el proceso de diseñar un modelo de un sistema real, obtener experiencia con la simulación, con la finalidad de comprender el comportamiento del sistema a evaluar, el objetivo de la simulación es mejorar la comprensión del movimiento del brazo del robot, sin necesidad de usar un prototipo físico.

Para poder lograr el objetivo se hace uso de un Toolbox de Robótica implementado en MATLAB; en este caso se analizará la respuesta del mecanismo mencionado ante la simulación de la cinemática directa e inversa.

Inicialización general del programa.

Mediante la simulación de la cinemática se busca conocer la relación que existen entre las coordenadas articulares y la ubicación del robot, realizando los cálculos numéricos para obtener sus correspondientes resultados numéricos y también gracias, por lo cual se emplea el uso de MATLAB y de Toolbox.

Para el calcular y simular la cinemática directa se realiza lo siguiente:

3.4.1 FUNCIÓN LOADDATA.

Función “loaddata” importa datos del modelo del robot una vez leído los datos CAD se almacenan en variables locales, para tenerlos disponibles.

```

% Esta función carga los datos de 3D CAD.
%
function loaddata
% Carga todos los datos de enlace del archivo.
% Esta información proviene de un modelo CAD Pro / E 3D y se hizo con cad2matdemo.m

[modelado]=load('modelado.mat','Base0','Esla1','Esla2','Esla3','Esla4','Herr5','Pinz6','Areal');

%Asigna los datos del archivo a variables locales.
setappdata(0,'Base',modelado.Base0);
setappdata(0,'Eslabon1',modelado.Esla1);
setappdata(0,'Eslabon2',modelado.Esla2);
setappdata(0,'Eslabon3',modelado.Esla3);
setappdata(0,'Eslabon4',modelado.Esla4);
setappdata(0,'Herramienta',modelado.Herr5);
setappdata(0,'Pinza',modelado.Pinz6);
setappdata(0,'Area',modelado.Areal);

end

```

Ilustración 3. 23 Función loaddata.

3.4.2. FUNCIÓN INITHOME

La primera representación del robot y espacio de trabajo se realiza con la función InitHome, la cual tiene dos tareas fundamentales dibujar y posicionar el robot en una configuración inicial y calcular la cinemática de directa de esa configuración inicial, esos valores serán almacenados para posteriormente ser utilizados y realizar los subsecuentes cálculos. A continuación se describe los procesos que realiza esta función:

- Creación de la ventana y de la cuadrícula.
- Recupera los valores de la función “loaddata”.
- define y asigna valores a las constantes de acuerdo a los parámetros definidos con anterioridad en este capítulo en la tabla de parámetros de Denavit-hartenberg.
- Define y asigna valores a las constantes de los ángulos de las articulaciones de acuerdo a una configuración inicial de acuerdo al fabricante.
- Llamado de la función “tmat” la cual realiza la matriz de transformación homogénea y la asigna a las variables definidas. En este caso las “n” número de veces necesarias de acuerdo a los grados de libertad.
- Realiza la multiplicación de todas las matrices en forma ordenada.
- Define los datos de los vértices reales de los enlaces del robot.
- Dibuja los elementos del robot, de acuerdo a los datos obtenidos de la cinemática directa y los datos de las piezas del robot obtenidas en la función “loaddata”

```

Función Inithome
% Función para dibujar y posicionar el robot en configuración
% inicial , utilizando la cinemática directa
set(0,'Units','pixels')
dim = get(0,'ScreenSize');
Fig = figure('Name','inithome','m','Position',[0,35,dim(3)-200,dim(4)-110],...
'Number','none','Name','MITSUBISHI RVM1',...
'NumberTitle','off','CloseRequestFcn','close_app'; %genera la ventana y el título de la ventana
hold on
light('Position',[-1 0 0]); %iluminación
daspect([1 1 1])
view(225,20); %vamos
xlabel('X'),ylabel('Y'),zlabel('Z'); %etiqueta los ejes.
title('Mitsubishi R22 Simulador RVM '); % título
axis([-600 600 -600 600 0 1000]); %axiación de la escala a los ejes
grid on %genera cuadrícula

%ingra valores cargados en la función loaddata
Base0 = getappdata(0,'Base');
Eje1a = getappdata(0,'Eje1a');
Eje1b = getappdata(0,'Eje1a');
Eje1c = getappdata(0,'Eje1a');
Eje1d = getappdata(0,'Eje1a');
Eje1e = getappdata(0,'Eje1a');
Eje1f = getappdata(0,'Eje1a');
Eje1g = getappdata(0,'Eje1a');
Eje1h = getappdata(0,'Eje1a');
Eje1i = getappdata(0,'Eje1a');
Eje1j = getappdata(0,'Eje1a');

% Parametros D-H definidos en documento
a2 = 250;
a3 = 140;
d1 = 300;
d4 = 72;
d5 = 107;
Pz = 5000;
Py = 5000;
Px = 5000;

%Configuración Home según el fabricante.
t1 = 0;
t2 = 100;
t3 = 0;
t4 = -90;
t5 = 0;

```

Ilustración 3. 24 Función inithome.

El resultado de las funciones anteriores se presenta a continuación:

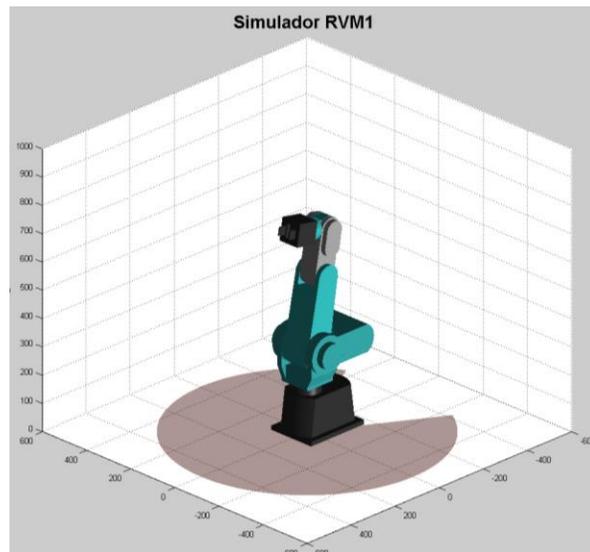


Ilustración 3. 25 Resultados de las funciones loaddata e inithome.

Posteriormente el programa procede a crear los diferentes controles o herramientas para la manipulación del robot, botón de estado del robot, botón de mensaje de error, panel de resultado de la cinemática directa, panel de resultado de la cinemática inversa, teaching box, los cuales se describen a continuación:

Creación de barras de deslizamiento para la manipulación del movimiento del robot. Estas moverán al robot de acuerdo a los parámetros permitidos. La barra se desplaza en incrementos definidos para poder observar claramente los movimientos. Cada articulación cuenta con su propia barra para aumentar o disminuir los valores de acuerdo a sus límites permitidos.

Creación de status del robot. Este botón indica si el robot se encuentra en operación.
Botón de error, este botón indica si existe un punto inalcanzable.

Panel de cinemática inversa mediante el cual se pueden introducir los parámetros e imprimir los resultados de los cálculos.

Panel de cinemática directa el cual despliega los resultados de los cálculos.

Teaching box en la vida real es un dispositivo, que permite manejar manualmente el robot. Cuenta con un teclado cuyos botones presentan varias funciones; a continuación se presentan y se describen en la siguiente figura los botones en funcionamiento que fueron implementados como muestra de lo que se puede llegar a realizar:

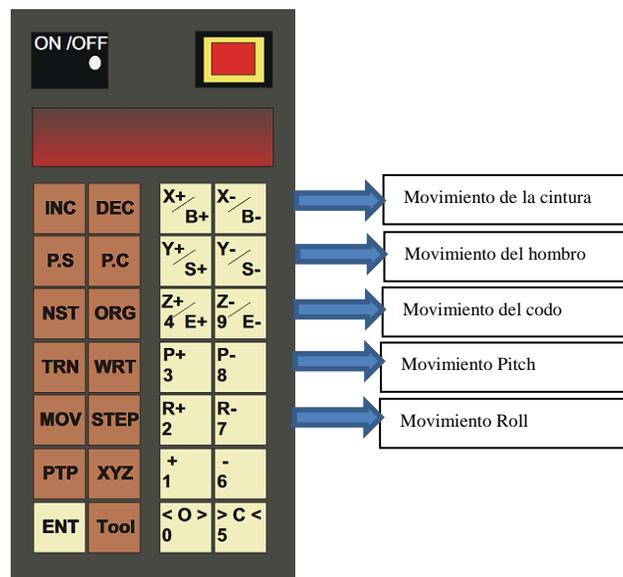


Ilustración 3. 26 Teaching box.

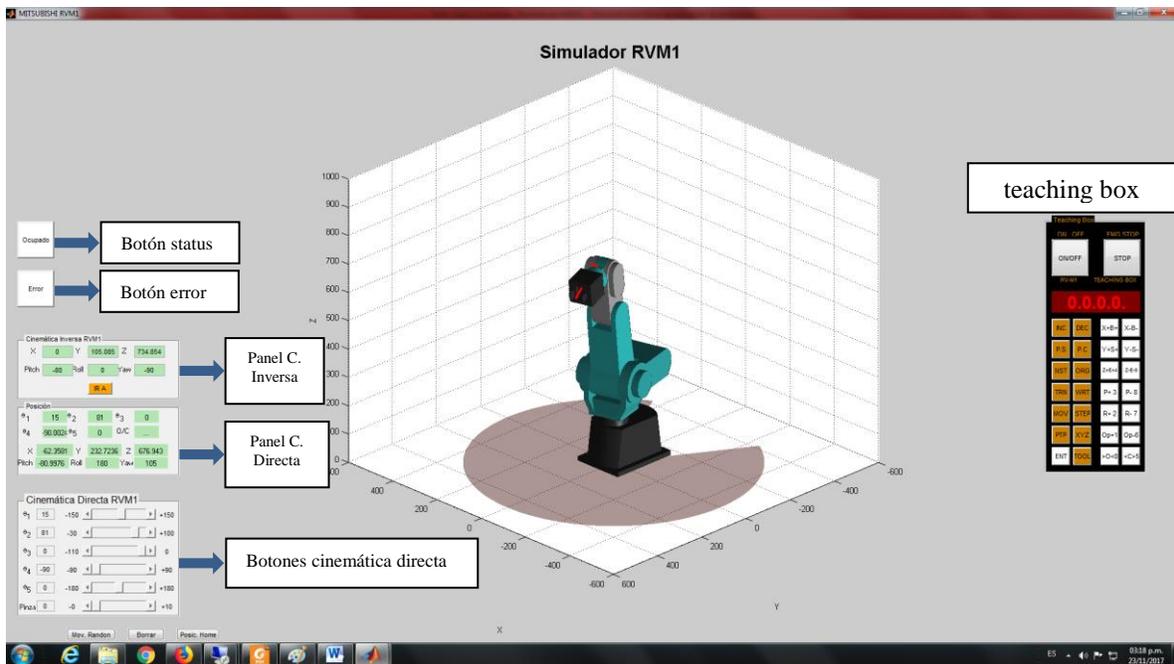


Ilustración 3. 27 Carga final del programa.

3.4.3. FUNCIÓN PARA EL CÁLCULO CINEMÁTICA DIRECTA.

Con anterioridad se ha creado la interfaz de la simulación así como también se calculó la cinemática directa con parámetros específicos que servirán de referencia para los cálculos futuros de acuerdo a las acciones solicitadas; ahora bien se describirá la función de la programación que se encarga del cálculo de la cinemática directa al interactuar con los botones creados para la manipulación de las articulaciones, es decir, cada vez que el usuario realice un incremento o decremento en los botones correspondientes a las variables articulares del robot, el programa procede a ejecutar la siguiente rutina “RVM1_Ani”, la cual se describe a continuación:

- Define y asigna valores de acuerdo a los parámetros denavit-hartenberg.
- Ilumina el botón de ocupado durante la ejecución de la función.
- Define y asigna valores previamente calculados.
- Define y asigna valores recuperados de los botones de cinemática directa.
- Invoca la función “tmat” la cual se encarga del cálculo de la cinemática directa.
- Realiza el cálculo para los valores de yaw,pitch, roll.
- Actualiza los valores en el panel de cinemática directa.
- Realiza la animación.
- Libera el botón de ocupado al término de la función.

```

%% Cinesematica directa con animacion
function RVM1_Ani(theta1,theta2,theta3,theta4,theta5,theta6,s,trail)
% Funcion para animar al robot
% s es el numero de pasos para la animacion del robot
% Parametros de D-R
a2 = 250;
a3 = 140;
d1 = 300;
d4 = 72; %ARRASTRISTA
d5 = 107;
set(horcon_ocupado,'BackgroundColor',[0 1 0]);
set(horcon_estado,'BackgroundColor',[1 1 1]);
ThetaOld = getappdata(0,'ThetaOld');
theta1old = ThetaOld(1);
theta2old = ThetaOld(2);
theta3old = ThetaOld(3);
theta4old = ThetaOld(4);
theta5old = ThetaOld(5);
theta6old = ThetaOld(6);
%
t1 = linspace(theta1old,theta1,n);
t2 = linspace(theta2old,theta2,n);
t3 = linspace(theta3old,theta3,n)/% -180;
t4 = linspace(theta4old,theta4,n);
t5 = linspace(theta5old,theta5,n);
t6 = linspace(theta6old,theta6,n);
a1 = num2str(theta1);
a2 = num2str(theta2);
a3 = num2str(theta3);
set(vlabz_text_t1,'string',a1);
set(vlabz_text_t2,'string',a2);
set(vlabz_text_t3,'string',a3);
a1 = num2str(theta4);
a2 = num2str(theta5);
set(vlabz_text_t4,'string',a1);
set(vlabz_text_t5,'string',a2);
for i = 1:n
% Forward Kinematics
%
T_01 = tmat(90,0,d1,t5(i)+90); %+90(cambio de eje x a y)
T_12 = tmat(0,a2,0,t2(i));
T_23 = tmat(0,a3,0,t3(i));
T_34 = tmat(90,0,0,t4(i)+90);
T_45 = tmat(0,0,d4,t5(i));

```

Ilustración 3. 28 Función rvm1_ani.

3.4.4. DESCRIPCIÓN DE LOS BOTONES DE CINEMÁTICA DIRECTA.

Se crearon 5 botones para asignar valores para cada uno de los ángulos de las articulaciones del robot; los valores fueron definidos de acuerdo a las características del robot, es decir, solamente se pueden asignar los valores permitidos. A continuación, se describe el funcionamiento general para cada uno de las barras de desplazamiento.

- El botón realiza la validación si existe evento de incremento o decremento.
- Define y asigna los últimos valores para las demás articulaciones.
- Invoca la función RVM1_Ani. (función descrita con anterioridad)

```

% control para movimiento de Theta 1.
function t1_slider_button_press(h,-)
slider_value = round(get(h,'Value'));
set(t1_edit,'string',slider_value);
T_Old = getappdata(0,'ThetaOld');
t2old = T_Old(2); t3old = T_Old(3); t4old = T_Old(4);
t5old = T_Old(5); t6old = T_Old(6);
RVM1_Ani(slider_value+t1_home,t2old,t3old,t4old,t5old,t6old,10,'y');
end
%
% control para movimiento de Theta 2.
function t2_slider_button_press(h,-)
slider_value = round(get(h,'Value'));
set(t2_edit,'string',slider_value);
T_Old = getappdata(0,'ThetaOld');
t1old = T_Old(1); t3old = T_Old(3); t4old = T_Old(4);
t5old = T_Old(5); t6old = T_Old(6);
RVM1_Ani(t1old,slider_value+t2_home,t3old,t4old,t5old,t6old,10,'y');
end
%
% control para movimiento de Theta 3.
function t3_slider_button_press(h,-)
slider_value = round(get(h,'Value'));
set(t3_edit,'string',slider_value);
T_Old = getappdata(0,'ThetaOld');
t1old = T_Old(1); t2old = T_Old(2); t4old = T_Old(4);
t5old = T_Old(5); t6old = T_Old(6);
RVM1_Ani(t1old,t2old,slider_value+t3_home,t4old,t5old,t6old,10,'y');
end
%
% control para movimiento de Theta 4.
function t4_slider_button_press(h,-)
slider_value = round(get(h,'Value'));
set(t4_edit,'string',slider_value);
T_Old = getappdata(0,'ThetaOld');
t1old = T_Old(1); t2old = T_Old(2); t3old = T_Old(3);
t5old = T_Old(5); t6old = T_Old(6);
RVM1_Ani(t1old,t2old,t3old,slider_value+t4_home,t5old,t6old,10,'y');
end

```

Ilustración 3. 29 Función para creación de botones.

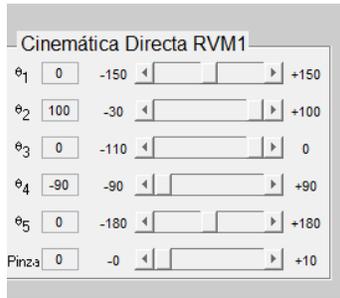


Ilustración 3. 30 Botones barra deslizamiento.

3.4.5. PANEL DE POSICIÓN CINEMÁTICA DIRECTA.

El visor de posición imprime los valores en pantalla recuperados de las funciones de cálculo de cinemática directa; a continuación se describe su creación y su funcionamiento:

- Se crea en la sección de código “VISOR POSICION” en la línea 365 del código, consta en la creación de un contenedor, 12 cajas de texto donde se imprimen los valores recuperados por las funciones, y cada caja de texto está etiquetada de acuerdo al valor que representa.

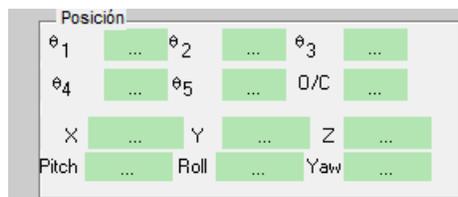


Ilustración 3. 31 Panel de posición cinemática directa.

```

367  % VISOR POSICION
368  visor = uipanel([0 0 270 240]);
369  'Title','Posición','FontSize',8);
370  uicontrol(visor,'style','text',...
371  'String','X',...
372  'Position',[30 200 20]); % L, B, W, H
373  visor_text_x = uicontrol(visor,'style','text',...
374  'String',...
375  'Position',[20 30 40 20]); % L, B, W, H
376  visor_text_y = uicontrol(visor,'style','text',...
377  'String',...
378  'Position',[14 30 55 20]); % L, B, W, H
379  visor_text_z = uicontrol(visor,'style','text',...
380  'String',...
381  'Position',[190 30 55 20]); % L, B, W, H
382  uicontrol(visor,'style','text',...
383  'String','Pitch',...
384  'Position',[5 10 180 20]); % L, B, W, H
385  visor_text_pitch = uicontrol(visor,'style','text',...
386  'String',...
387  'Position',[20 10 55 18]); % L, B, W, H
388  visor_text_roll = uicontrol(visor,'style','text',...
389  'String',...
390  'Position',[110 10 55 18]); % L, B, W, H
391  visor_text_yaw = uicontrol(visor,'style','text',...
392  'String',...
393  'Position',[190 10 55 18]); % L, B, W, H
394  uicontrol(visor,'style','text',...
395  'String','\theta_4',...
396  'Position',[0 40 190 20]); % L, B, W, H
397  visor_text_4 = uicontrol(visor,'style','text',...
398  'String',...
399  'Position',[40 40 40 20]); % L, B, W, H
400  visor_text_5 = uicontrol(visor,'style','text',...
401  'String',...
402  'Position',[14 40 40 20]); % L, B, W, H
403  visor_text_OC = uicontrol(visor,'style','text',...
404  'String',...
405  'Position',[190 40 40 20]); % L, B, W, H
406  uicontrol(visor,'style','text',...
407  'String','\theta_2',...
408  'Position',[0 85 180 20]); % L, B, W, H
409  visor_text_2 = uicontrol(visor,'style','text',...
410  'String',...

```

Ilustración 3. 32 Código para el panel de posición cinemática directa.

3.4.6. PANEL CINEMÁTICA INVERSA.

El panel de la cinemática inversa se crea en la sección de código de la línea 35 con el comentario “interfaz gráfica para cinemática inversa”; a continuación se describen sus principales características:

- Se crea un contenedor para los cuadros de texto y las etiquetas.
- Se crean las etiquetas y los cuadros de texto correspondientes para ingresar los valores en los que se desea posicionar y orientar el efector final.
- Creación botón para enviar datos.
- Para poder realizar los cálculos de la cinemática inversa, al presionar el botón se llama a la función “btn_ik_RVM1” la cual define y asigna valores a las variables, los cuales recibe de los cajas de texto con las etiquetas x,y,z,pitch, roll, yaw. Esta función invoca propiamente a la función “RVM1_IK3” que se describirá en el siguiente punto.

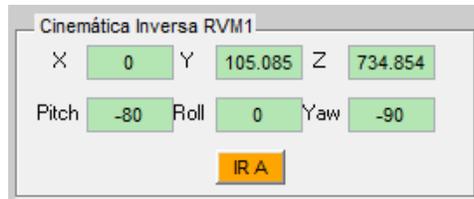


Ilustración 3. 33 Panel cinemática inversa.

```

20 % panel de la cinemática inversa.
21 %_p = uipanel(fig,...
22 %title,'panel',...
23 %position,[20 40 240 200],...
24 %title,'Cinemática Inversa RVM1','FontSize',12);
25
26 %
27 %_h_home = 0; % offset to define the "home" position as UP.
28 %_v_home = 0;
29 %_h_home = 0;
30 %_v_home = 0;
31 %_h = 100; % left, used to set the GUI.
32 %_v = 10; % height.
33 %_BT = 100; % button.
34
35 % interfaz grafica para cinemática inversa
36 ik_a = uipanel(fig,'title','panel',...
37 %position,[20 40 240 200],...
38 %title,'Cinemática Inversa RVM1','FontSize',8);
39 %
40 %_BT = uicontrol(ik_a,'style','text',...
41 %string,'IRA',...
42 %position,[100 180 200],...
43 %callback,@default,...
44 %background,[.7 .9 .7]); % L, B, W.
45 %_edit_x = uicontrol(ik_a,'style','edit',...
46 %string,'0',...
47 %callback,@default,...
48 %position,[40 60 20],...
49 %background,[.7 .9 .7]); % L, B, W.
50 %_edit_y = uicontrol(ik_a,'style','edit',...
51 %string,'105.085',...
52 %callback,@default,...
53 %position,[40 80 20],...
54 %background,[.7 .9 .7]); % L, B, W.
55 %_edit_z = uicontrol(ik_a,'style','edit',...
56 %string,'734.854',...
57 %callback,@default,...
58 %position,[40 100 20],...
59 %background,[.7 .9 .7]); % L, B, W.
60 %_edit_pitch = uicontrol(ik_a,'style','edit',...
61 %string,'-80',...
62 %callback,@default,...
63 %position,[80 180 20],...
64 %background,[.7 .9 .7]); % L, B, W.
65 %_edit_roll = uicontrol(ik_a,'style','edit',...
66 %string,'0',...
67 %callback,@default,...
68 %position,[80 160 20],...
69 %background,[.7 .9 .7]); % L, B, W.
70 %_edit_yaw = uicontrol(ik_a,'style','edit',...
71 %string,'-90',...
72 %callback,@default,...
73 %position,[80 140 20],...
74 %background,[.7 .9 .7]); % L, B, W.

```

Ilustración 3. 34 Código panel cinemática inversa.

3.4.7. FUNCIÓN DE CINEMÁTICA INVERSA RVM1_IK3.

La función rvm1_ik3 es el código que se encarga de realizar la cinemática inversa las características generales de la rutina se describen a continuación:

- La función recibe los parámetros de entrada del panel de cinemática inversa.

- Define y asigna valores a los parámetros denavit-hartenberg.
- Realiza el cálculo de la cinemática inversa; el código se basa en el modelo matemático descrito en este capítulo.
- Realiza la validación de los cálculos y determinar si es un punto alcanzable para el robot.
- Con los datos obtenidos procede a calcular nuevamente la cinemática directa la cual se encarga a su vez de realizar el llamado a la función correspondiente para realizar la animación.

```

1 function [theta1,theta2,theta3,theta4,theta5,error] = RVM1_TF3(px, py, pz, yaw, pitch, roll)
2 %funcion para el calculo de la cinemática inversa.
3
4 a2 = 250;
5 a3 = 140;
6 d1 = 300;
7 d4 = 72;
8 d5 = 107;
9 s = d1 + d3;
10 %
11
12 Rz = [cosd(yaw), -sind(yaw), 0,0;
13       sind(yaw), cosd(yaw), 0,0;
14       0,0, 0,0, 1,0];
15
16 Ry = [cosd(pitch), 0,0, sind(pitch);
17       0,0, 1,0, 0,0;
18       -sind(pitch), 0,0, cosd(pitch)];
19
20 Rx = [1,0, 0,0, 0,0;
21       0,0, cosd(roll), -sind(roll);
22       0,0, sind(roll), cosd(roll)];
23
24 O = Rz * Ry * Rx;
25 P = [px;
26      py;
27      pz];
28 v1,0];
29
30 TRz = [cosd(90), -sind(90), 0,0;
31        sind(90), cosd(90), 0,0;
32        0,0, 0,0, 1,0];
33
34 TRx = [0,0;
35        0,0;
36        1,0,0];
37 %
38 %
39 O = TRz * O;
40 P = TRz * P + Trx;
41
42 z = [0,0, 0,0, 0,0];
43
44 T06 = [0, P;
45        1,0];

```

Ilustración 3. 35 Código cinemática inversa.

3.4.8. BOTONES DE PROPÓSITO GENERAL.

Adicionalmente se crearon 3 diferentes botones los cuales se mencionan y describen a continuación:

- Botón de movimiento aleatorio: este botón se construye en la línea 12 código; al generar un evento sobre este botón (presionarlo) ejecuta la función “rnd_demo_button_press” la cual define valores aleatorios permitidos de acuerdo a las limitaciones de las articulaciones; posteriormente ejecuta la función “RVM1_Ani”, la cual se encarga de calcular la cinemática directa y realizar la animación.

```

623 %
624 % function dir_trail_button_press(h, dummy)
625 % disp('pushed clear trail bottom');
626 handles = getappdata(h,'pasc_h'); %
627 Tr = handles(9); %
628 %
629 setappdata(h,'trail',0); % used for trail tracking.
630 setappdata(h,'trail',0); % used for trail tracking.
631 setappdata(h,'trail',0); % used for trail tracking.
632 %
633 set(Tr,'xdata',0,'ydata',0,'zdata',0);
634 %
635 %
636 %
637 %
638 %
639 %
640 %
641 %
642 %
643 %
644 %
645 %
646 %
647 %
648 % function end_mov_aleatorio(h, dummy)
649 %
650 % t1_home = 0; %
651 % t2_home = 0;
652 % t3_home = 0;
653 % t4_home = 0;
654 % theta1 = -150 + 300*rand(); %
655 % theta2 = -30 + 120*rand(); %
656 % theta3 = -110 + 110*rand();
657 % theta4 = -90 + 180*rand();
658 % theta5 = -180 + 360*rand();
659 % theta6 = -0 + 0*rand();
660 %
661 % n = 50;
662 % RVN1_angi(theta1=t1_home,theta2=t2_home,theta3=t3_home,theta4,theta5,theta6,n,'y')
663 %
664 % set(i1_slider,'string',round(theta1)); %
665 % set(i1_slider,'value',round(theta1));
666 % set(i2_slider,'string',round(theta2));
667 % set(i2_slider,'value',round(theta2));
668 % set(i3_slider,'string',round(theta3));
669 % set(i3_slider,'value',round(theta3));
670 % set(i4_slider,'string',round(theta4));
671 % set(i4_slider,'value',round(theta4));
672 % set(i5_slider,'string',round(theta5));
673 % set(i5_slider,'value',round(theta5));
674 %
675 %
676 %
677 %
678 %
679 %
680 %
681 %
682 %
683 %
684 %
685 %
686 %
687 %
688 %
689 %
690 %
691 %
692 %
693 %
694 %
695 %
696 %
697 %
698 %
699 %
700 %
701 %
702 %
703 %
704 %
705 %
706 %
707 %
708 %
709 %
710 %
711 %
712 %
713 %
714 %
715 %
716 %
717 %
718 %
719 %
720 %
721 %
722 %
723 %
724 %
725 %
726 %
727 %
728 %
729 %
730 %
731 %
732 %
733 %
734 %
735 %
736 %
737 %
738 %
739 %
740 %
741 %
742 %
743 %
744 %
745 %
746 %
747 %
748 %
749 %
750 %
751 %
752 %
753 %
754 %
755 %
756 %
757 %
758 %
759 %
760 %
761 %
762 %
763 %
764 %
765 %
766 %
767 %
768 %
769 %
770 %
771 %
772 %
773 %
774 %
775 %
776 %
777 %
778 %
779 %
780 %
781 %
782 %
783 %
784 %
785 %
786 %
787 %
788 %
789 %
790 %
791 %
792 %
793 %
794 %
795 %
796 %
797 %
798 %
799 %
800 %
801 %
802 %
803 %
804 %
805 %
806 %
807 %
808 %
809 %
810 %
811 %
812 %
813 %
814 %
815 %
816 %
817 %
818 %
819 %
820 %
821 %
822 %
823 %
824 %
825 %
826 %
827 %
828 %
829 %
830 %
831 %
832 %
833 %
834 %
835 %
836 %
837 %
838 %
839 %
840 %
841 %
842 %
843 %
844 %
845 %
846 %
847 %
848 %
849 %
850 %
851 %
852 %
853 %
854 %
855 %
856 %
857 %
858 %
859 %
860 %
861 %
862 %
863 %
864 %
865 %
866 %
867 %
868 %
869 %
870 %
871 %
872 %
873 %
874 %
875 %
876 %
877 %
878 %
879 %
880 %
881 %
882 %
883 %
884 %
885 %
886 %
887 %
888 %
889 %
890 %
891 %
892 %
893 %
894 %
895 %
896 %
897 %
898 %
899 %
900 %
901 %
902 %
903 %
904 %
905 %
906 %
907 %
908 %
909 %
910 %
911 %
912 %
913 %
914 %
915 %
916 %
917 %
918 %
919 %
920 %
921 %
922 %
923 %
924 %
925 %
926 %
927 %
928 %
929 %
930 %
931 %
932 %
933 %
934 %
935 %
936 %
937 %
938 %
939 %
940 %
941 %
942 %
943 %
944 %
945 %
946 %
947 %
948 %
949 %
950 %
951 %
952 %
953 %
954 %
955 %
956 %
957 %
958 %
959 %
960 %
961 %
962 %
963 %
964 %
965 %
966 %
967 %
968 %
969 %
970 %
971 %
972 %
973 %
974 %
975 %
976 %
977 %
978 %
979 %
980 %
981 %
982 %
983 %
984 %
985 %
986 %
987 %
988 %
989 %
990 %
991 %
992 %
993 %
994 %
995 %
996 %
997 %
998 %
999 %
1000 %

```

Ilustración 3. 36 Código para el movimiento aleatorio.

- Botón de posicionamiento en inicio (home): Este botón permite posicionar al robot en la posición inicial llamada home, definida por la configuración de acuerdo a la documentación del robot. Esta función prácticamente llama a la función “home_button_press” que a su vez llama la función inithome que ya se ha descrito con anterioridad.

```

626 %
627 % function home_button_press(h, dummy)
628 % disp('pushed home bottom');
629 % gohome
630 %
631 %
632 %

```

Ilustración 3. 37 Código para el posicionamiento en home.

- Botones funcionales de la herramienta teaching box (botones de movimiento por joins). Como bien se mencionó anteriormente, ciertos botones del teaching box fueron habilitados para poder realizar una simulación de su función; en esta sección se describe en general el código para el movimiento de estos botones, en la sección de código documentada con el comentario “FUNCIONES MOVIMIENTOS POR JOIN” que se encuentra en la línea 1769; consta de 6 funciones que se describen a continuación:

- Función rotar “rotar_cintura” la cual está asociada a los botón del teaching box “B+” “B-“ que generan el movimiento en la base del robot o conocido también como cintura. Cada evento sobre los botones invoca la función “rotar_cintura” donde se genera el incremento o decremento definido de acuerdo a los parámetros permitidos para esas articulaciones, y así poder

observar la animación más fluida. A su vez esta función invoca la función “RVM1_Ani” que realiza la cinemática directa y la animación.

- Función rotar “rotar_hombro” la cual está asociada a los botón del teaching box “S+” “S-“ que generan el movimiento en la articulación 2 o conocido también como hombro. Cada evento sobre los botones invoca la función “rotar_hombro” donde se genera el incremento o decremento definido de acuerdo a los parámetros permitidos para esas articulaciones, y así poder observar la animación más fluida. A su vez, esta función invoca la función “RVM1_Ani” que realiza la cinemática directa y la animación.
- Función rotar “rotar_codo” la cual está asociada a los botón del teaching box “E+” “E-“ que generan el movimiento en la articulación 2 o conocido también como codo. Cada evento sobre los botones invoca la función “rotar_codo” donde se genera el incremento o decremento definido de acuerdo a los parámetros permitidos para esas articulaciones, y así poder observar la animación más fluida. A su vez esta función invoca la función “RVM1_Ani” que realiza la cinemática directa y la animación.
- Función rotar “rotar_pitch” la cual está asociada a los botón del teaching box “P+” “P-“ que generan el movimiento. Cada evento sobre los botones invoca la función “rotar_pitch” donde se genera el incremento o decremento definido de acuerdo a los parámetros permitidos para esas articulaciones, y así poder observar la animación más fluida. A su vez esta función invoca la función “RVM1_Ani” que realiza la cinemática directa y la animación.
- Función rotar “rotar_roll” la cual está asociada a los botón del teaching box “R+” “R-“ que generan el movimiento. Cada evento sobre los botones invoca la función “rotar_roll” donde se genera el incremento o decremento definido de acuerdo a los parámetros permitidos para esas articulaciones. y así poder observar la animación más fluida. A su vez esta función invoca la función “RVM1_Ani” que realiza la cinemática directa y la animación.

```

1768
1769
1770 %% FUNCIONES MOVIMIENTOS POR JOIN
1771
1772 function rotar_cintura(value,n,trail,sp)
1773     T_Old = getappdata(0,'ThetaOld');
1774     t2old = T_Old(2); t3old = T_Old(3); t4old = T_Old(4); t5old = T_Old(5); t6old = T_Old(6);
1775
1776     RVMI_Ani(value,t2old,t3old,t4old,t5old,t6old,n,trail);
1777     set(t1_edit,'string',value);
1778     set(t1_slider,'Value',value);
1779     set(t2_edit,'string',t2old);
1780     set(t2_slider,'Value',t2old);
1781     set(t3_edit,'string',t3old);
1782     set(t3_slider,'Value',t3old);
1783     set(t4_edit,'string',t4old);
1784     set(t4_slider,'Value',t4old);
1785     set(t5_edit,'string',t5old);
1786     set(t5_slider,'Value',t5old);
1787     set(t6_edit,'string',t6old);
1788     set(t6_slider,'Value',t6old);
1789     setappdata(0,'ThetaOld',[value,t2old,t3old,t4old,t5old,t6old]);
1790
1791 end
1792 function rotar_hombro(value,n,trail,sp)
1793     T_Old = getappdata(0,'ThetaOld');
1794     t1old = T_Old(1); t3old = T_Old(3); t4old = T_Old(4); t5old = T_Old(5); t6old = T_Old(6);
1795     RVMI_Ani(t1old,value,t3old,t4old,t5old,t6old,n,trail);
1796     set(t1_edit,'string',t1old);
1797     set(t1_slider,'Value',t1old); %At the home position, so all
1798     set(t2_edit,'string',value); %sliders and input boxes = 0.
1799     set(t2_slider,'Value',value);
1800     set(t3_edit,'string',t3old);
1801     set(t3_slider,'Value',t3old);
1802     set(t4_edit,'string',t4old);
1803     set(t4_slider,'Value',t4old);
1804     set(t5_edit,'string',t5old);
1805     set(t5_slider,'Value',t5old);
1806     set(t6_edit,'string',t6old);
1807     set(t6_slider,'Value',t6old);
1808     setappdata(0,'ThetaOld',[t1old,value,t3old,t4old,t5old,t6old]);
1809
1810 end
1811 function rotar_codo(value,n,trail,sp)
1812     T_Old = getappdata(0,'ThetaOld');
1813     t1old = T_Old(1); t2old = T_Old(2); t4old = T_Old(4); t5old = T_Old(5); t6old = T_Old(6);
1814     RVMI_Ani(t1old,t2old,value,t4old,t5old,t6old,n,trail);
1815     set(t1_edit,'string',t1old);

```

Ilustración 3. 38 Código para botones de teaching box.

A continuación en la siguiente imagen se representan de color verde los botones que están habilitados, los cuales se describieron con anterioridad; cabe señalar que no todos los botones se habilitaron, ya que es solamente una demostración de lo que se puede llegar a implementar en su totalidad.

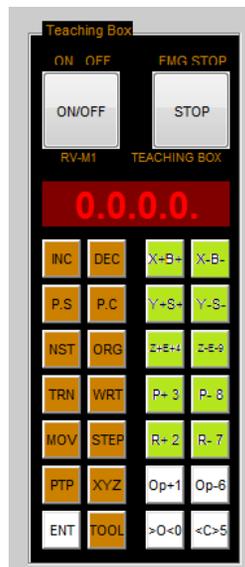


Ilustración 3. 39 Botones funcionales.

CAPÍTULO 4: CONCLUSIONES.

4.1 CONCLUSIONES.

En este trabajo se presentaron una serie de formulaciones analíticas para determinar el espacio de trabajo del robot. Esta formulación se realizó a través de la caja de herramientas de Matlab. Las diferentes funciones demostraron las restricciones en el espacio de trabajo del robot en términos de coordenadas generalizadas, y se demostró la implementación del cálculo de la cinemática directa del robot a través de la matriz de transformación homogénea en conjunto con su simulación. Se demostró la implementación del cálculo de la cinemática inversa del robot a través de Modelado con matrices inversas. Las ecuaciones en senos y cosenos son combinadas usando ecuaciones arctan de dos argumentos, para evitar ambigüedades de cuadrantes. Este método se aplica para el robot RV-M1 Mitsubishi y se desarrolla una caja de herramientas la cual cuando se ejecuta se muestra lo siguiente:

- Modelo tridimensional del robot y el espacio de trabajo.
- Simulación de movimiento del robot en su área de trabajo; esta área de trabajo obviamente se limita a la propia área de trabajo del robot.

4.2 TRABAJO A FUTURO.

Para desarrollar software del espacio de trabajo para el robot RV-M1 Mitsubishi en Matlab existe una amplia gama de posibilidades para extensiones y aperturas para investigación y desarrollo a futuros. Entre los cuales podrían incluir:

- La formulación analítica de este trabajo puede ser extendida, es decir, la resolución de los cálculos de la cinemática directa e inversa podrían ser resueltos por otros métodos matemáticos diferentes o precisos.
- Cálculo y simulación de la dinámica, es decir, obtener los valores de fuerzas y momentos requeridos en los actuadores con la finalidad de conseguir un comportamiento deseado en el movimiento del robot.
- Realizar la simulación en conjunto para la resolución de un problema de la vida real como por ejemplo, que el robot en su área de trabajo detecte objetos, interactúe con ellos y los mueva de lugar. Para esto se necesitarían incorporar algoritmos para la detección de objetos, mismos que fueron desarrollados en conjunto con esta tesis por estudiantes de la carrera de Ingeniería en Sistemas Computacionales del Instituto Tecnológico de Mérida.

REFERENCIAS

- Bonilla, F. R. (2013). *Clasificación de piezas de un robot mitsubishi rvm1*. México.
- Castro, J. A. (s.f.). *Unidad 10 Robótica Industrial. Práctica de laboratorio*.
- Cortés, F. R. (2012). *MATLAB aplicado a Robótica y Mecatrónica*. Pitágoras 1139, Col. Del Valle, 03100, México D.F.: Alfaomega.
- Craig, J. J. (2006). *Robótica*. México: PEARSON EDUCACIÓN, México, 2006.
- Fernández, W. E. (2016). *Cinemática de Robots Industriales*.
- FESTO COSIMIR. (s.f.). *COSIMIR EDUCATIONAL.GUÍA DE USUARIO*. GERMANY 2003.
- Goyal, K. (2002). *Matlab tool box for determining the workspace of mitsubishi robot RV-M1*.
- <http://manufacturabustos.blogspot.mx>. (03 de 09 de 2017). Recuperado el 03 de septiembre de 2017, de <http://manufacturabustos.blogspot.mx/2008/09/software-de-programacin-en-las-cim.html>
- <http://www.festo-didactic.com>. (Enero de 2016). Obtenido de <http://www.festo-didactic.com>: <http://www.festo-didactic.com>
- <http://www.ptolomeo.unam.mx:8080/xmlui/bitstream/handle/132.248.52.100/175/A5%20Cap%20C3%ADtulo%202.pdf?sequence=5>. (03 de septiembre de 2017). Recuperado el 03 de septiembre de 2017, de <http://www.ptolomeo.unam.mx:8080/xmlui/bitstream/handle/132.248.52.100/175/A5%20Cap%20C3%ADtulo%202.pdf?sequence=5>
- <https://es.mathworks.com>. (ENERO de 2018). Obtenido de <https://es.mathworks.com>: <https://es.mathworks.com>
- (s.f.). *INDUSTRIAL MICRO-ROBOT SYSTEM MODEL RV-M1*.
- Mitsubishi . (s.f.). *Manual de servicio Mitsubishi RV-M1*. Nagoya, Japón.
- Mitsubishi. (s.f.). *Manual de Instrucciones Mitsubishi RV-M1*. Nagoya, Japón.
- Mitsubishi RV-M1. (s.f.). *Safety Manual Mitsubishi RV-M1*. Nagoya, Japón.
- PEREZ, M. V. (MAYO de 2010). *SISTEMAS MECATRÓNICOS. ANALOGÍA Y MANUAL DE PRÁCTICAS*. MÉXICO D.F, MÉXICO.
- Zur, H. (2003). *Festo Didactic*. Denkendorf, Germany: D:HB-COSIMIR-EDUNET-EN.